

Marcelo Faquer Lorenz

**Técnicas de Reconhecimento de Arestas
em Imagens Baseado na Transformada de
Hough com Robustez Aumentada por
Aritmética Intervalar**

Monografia de Conclusão de Curso
apresentada à Escola Politécnica da
Universidade de São Paulo para obten-
ção do Título de Engenheiro

Marcelo Faquer Lorenz

Técnicas de Reconhecimento de Arestas em Imagens Baseado na Transformada de Hough com Robustez Aumentada por Aritmética Intervalar

Monografia de Conclusão de Curso
apresentada à Escola Politécnica da
Universidade de São Paulo para obten-
ção do Título de Engenheiro

Área de concentração:
Engenharia Mecatrônica

Orientador:
Prof. Dr. Marcos de Sales Guerra
Tsuzuki

Ficha Catalográfica

Lorenz, Marcelo Faquer

Técnicas de Reconhecimento de Arestas em Imagens Baseado na Transformada de Hough com Robustez Aumentada por Aritmética Intervalar. São Paulo, 2008. 33 p.

Monografia de Conclusão de Curso (Graduação) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Mecatrônica e de Sistemas Mecânicos.

1. Processamento de Imagens. 2. Transformada de Hough. 3. Detecção de Arestas. 4. Reconhecimento de Objetos I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Mecatrônica e de Sistemas Mecânicos.

à minha Família,
e à Mayra.

Agradecimentos

Agradeço a todos os professores, funcionários e amigos que tiveram participação ativa no meu desenvolvimento ao longo do curso.

Agradeço sobretudo à minha família, sempre presente mesmo estando longe, e à Mayra, por estar sempre do meu lado.

Resumo

A automação industrial utiliza-se da visão robótica para realizar as mais diversas atividades. Dentre elas há aquelas em que é necessário que objetos possam ser identificados nas imagens obtidas. Pode-se citar, por exemplo, o reconhecimento de caixas para o seu posterior transporte através de garras robóticas. Um método eficiente para identificar estes objetos é através da sua modelagem em formas geométricas primitivas, como retas e círculos, para posteriormente se recuperar a informação do objeto como um todo. Neste contexto, uma técnica de Processamento de Imagens, chamada Transformada de Hough, torna possível o reconhecimento de diversas formas geométricas em imagens. Neste trabalho é abordada sua versão para reconhecimento de retas, ou seja, das arestas dos objetos. A utilização da Aritmética Intervalar associada às variáveis da Transformada permitiu-se alcançar maior robustez neste método. Por fim são apresentados os resultados obtidos com sua implementação algorítmica em linguagem de programação C++, onde foram utilizadas como base bibliotecas “open source” de processamento de imagens, chamada CxImage, e de Aritmética Intervalar, chamada Boost.

Abstract

Industrial automation uses robotic vision to do many activities. Between them there are activities where it is necessary to recognize objects in images. For example, the recognition of boxes that will be transported by a robotic arm. An efficient way to identify these objects is through their modelling in primitive geometric shapes, like lines and circles, to obtain the information about the entire object later. In this context, there is an Image Processing technique, called Hough Transform, which makes possible the recognition of many different geometric shapes in images. In this work it is studied its version for recognition of lines, in order to obtain object edges. The utilization of Interval Arithmetic related to the variables of the Transform has increased the robustness of the method. In the end, some results are presented with its algorithm implemented in C++ programming language, using as base open sources libraries of image processing, called CxImage, and Interval Arithmetic, called Boost.

Conteúdo

Lista de Figuras

1	Introdução	11
2	Definições Iniciais	12
2.1	<i>Threshold</i>	13
3	Filtragem Espacial Linear	14
3.1	Filtros de Suavização	15
3.2	Filtros de Acentuação	16
3.3	Detecção de Bordas	17
4	Transformada de Hough	20
4.1	Detecção de Retas	20
4.2	Detecção de Retas - Forma Normal	23
5	Aritmética Intervalar	25
5.1	Definições Básicas	25
5.2	Aplicação no Processamento de Imagens	26
6	Resultados Práticos Obtidos	27
7	Conclusões	32
	Referências	33

Lista de Figuras

2.1	Vizinhanças dos pixels p1, p2 e p3 destacadas em tom de cinza. O pixel p1 é o único a possuir vizinhança completa.	12
2.2	(a) Imagem original, (b) <i>Threshold</i> aplicado com limite 126, (c) <i>Threshold</i> aplicado com limite 200.	13
3.1	Imagem com origem do sistema de coordenadas e pixels de interesse em evidência.	14
3.2	Máscara com pesos de cada pixel.	14
3.3	(a) Máscara de suavização 3x3, (b) Máscara de suavização 5x5, (c) Máscara de suavização 7x7	16
3.4	(a) Imagem original, (b) Imagem com ruídos, (c) Resultado da aplicação da máscara de suavização 3x3 sobre b, (d) Resultado da aplicação da máscara de suavização 5x5 sobre b.	16
3.5	(a) Máscara de acentuação 3x3, (b) Máscara de acentuação 5x5, (c) Máscara de acentuação 7x7.	17
3.6	(a) Imagem original, (b) Resultado da aplicação da Máscara de Acentuação.	17
3.7	(a) Máscara de Sobel para bordas horizontais, (b) Máscara de Sobel para bordas verticais.	18
3.8	(a) Imagem original, (b) Aplicada a máscara de Sobel para bordas horizontais, (c) Aplicada a máscara de Sobel para bordas verticais.	18
3.9	(a) Sobel total: composição dos resultados, (b) Sobel total, seguido pela aplicação do <i>threshold</i>	19
4.1	Exemplo de um conjunto de pixels de borda encontrados.	21
4.2	Mapeamento dos pixels no Espaço Paramétrico.	22
4.3	Traçamento das retas encontradas no Espaço Geométrico.	23
4.4	Representação de uma reta na Forma Normal no E. G.	24

6.1	Imagem de teste original.	28
6.2	Pixels de borda encontrados aplicando-se a máscara de Sobel seguido por um threshold.	28
6.3	Espaço Paramétrico computado (ρ na vertical e θ na horizontal). .	29
6.4	(a) Retas encontradas traçadas em azul no E.G. e sobrepostas aos pixels de bordas, (b) Apenas as retas encontradas na imagem. . .	29
6.5	Imagem de teste original.	30
6.6	Pixels de borda encontrados aplicando-se a máscara de Sobel seguido por um threshold.	30
6.7	Espaço Paramétrico computado (ρ na vertical e θ na horizontal). .	30
6.8	(a) Retas encontradas traçadas em azul no E.G. e sobrepostas aos pixels de bordas, (b) Apenas as retas encontradas na imagem. . .	31

1 Introdução

As técnicas de Processamento de Imagens são meios poderosos para se extrair informações de uma imagem. Suas aplicações são diversas e vão desde a correção de contraste e luminosidade em uma imagem até técnicas mais complexas, como a Transformada de Hough aqui estudada ^[2, 8]. O objetivo deste documento é esclarecer os aspectos teóricos relacionados ao desenvolvimento da Transformada, para depois apresentar os resultados obtidos com sua implementação computacional. Os tópicos seguem uma ordem crescente de complexidade, se iniciando com a apresentação dos conceitos básicos que envolvem o processamento de imagens. A seguir são apresentados os métodos de Filtragem Espacial Linear, que são utilizados como uma etapa intermediária no desenvolvimento final proposto. Em seguida é apresentado o método principal, a Transformada de Hough ^[1, 4], em sua versão para detecção de retas em imagens, e a forma de utilização da Aritmética Intervalar. Por fim ocorre a apresentação e discussão dos resultados práticos obtidos, seguido pelas conclusões do trabalho desenvolvido.

2 Definições Iniciais

Primeiramente, deve-se definir o que é uma imagem. Simplificadamente, uma imagem pode ser representada como uma matriz de pixels, em que cada um deles possui uma coordenada (x,y) e um valor de intensidade, a qual define sua cor. Em uma imagem em escala de cinza (“preto-e-branco”) a intensidade do pixel assume valores entre 0 (preto) e 255 (branco). Já a imagem colorida é composta por três canais de intensidade de pixel: R (“Red” - Vermelho), G (“Green” - Verde) e B (“Blue” - Azul). Cada canal possui uma intensidade de pixel de 0 a 255, e a composição das intensidades nessas três cores é que formam a cor do pixel.

Cada pixel possui outros pixels ao redor que formam sua vizinhança. Normalmente são 2 vizinhos na vertical, 2 na horizontal e 4 nas diagonais. Entretanto, esses valores são menores para pixels nas bordas da imagem. Isto pode ser notado na Figura 2.1.

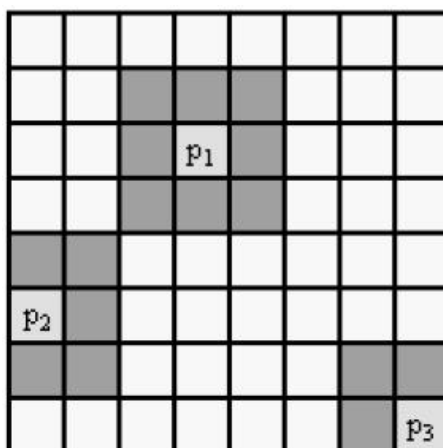


Figura 2.1: Vizinhanças dos pixels p1, p2 e p3 destacadas em tom de cinza. O pixel p1 é o único a possuir vizinhança completa.

2.1 Threshold

Muitas vezes o termo *threshold* é utilizado ao longo deste documento, por isso deve-se defini-lo corretamente. Por ser um termo em inglês de uso recorrente no Brasil, ele não será aqui traduzido. Porém, sua tradução seria algo como “separar por um limite”. Portanto, aplicar um *threshold* em uma imagem significa tornar sua intensidade de pixels binária: ou preta (0) ou branca (255). Assim, percorre-se a imagem, cada pixel desta assume um destes dois valores possíveis de intensidade. Entretanto, deve haver um critério para definir quais pixels tornar pretos e quais tornar brancos. Isso é feito através da definição do limite do *threshold*, que é um valor de intensidade de pixel entre 0 e 255.

Admitindo que x seja o limite definido, para cada pixel da imagem com intensidade $p(x,y)$:

- Se $p(x,y) \geq x$, então $p(x,y)_{NOVO} = 256$ (branco).
- Se $p(x,y) < x$, então $p(x,y)_{NOVO} = 0$ (preto).

A Figura 2.2 ilustra um exemplo da sua aplicação.



Figura 2.2: (a) Imagem original, (b) *Threshold* aplicado com limite 126, (c) *Threshold* aplicado com limite 200.

Em alguns casos o limite definido não é capaz de, por exemplo, destacar um objeto de interesse de ruídos presentes na imagem. Problemas desse tipo podem ser resolvidos através de técnicas mais avançadas de aplicação de *threshold*. Uma delas consiste em dividir a imagem em regiões de interesse e aplicar um *threshold* diferente em cada uma delas, com limites diferentes definidos de acordo com os objetivos da aplicação. Neste trabalho utiliza-se o método simples, com apenas um limite para toda a imagem.

3 Filtragem Espacial Linear

A Filtragem Espacial Linear é uma categoria dos métodos de Processamento de Imagens, sendo uma das mais utilizadas. Seus métodos consistem em se obter uma nova imagem onde cada um dos seus pixels é obtido pela transformação de um conjunto de pixels da imagem original. Este conjunto é formado pelo pixel equivalente (mesma posição x e y) da imagem original, mais seus pixels vizinhos. Portanto, a intensidade de cada novo pixel é mapeada em uma soma ponderada (linear) do valor das intensidades desse conjunto. Este processo ocorre através da aplicação de uma “máscara” sobre a imagem. Considere a Figura 3.1, em que a região de pixels de interesse encontra-se destacada.

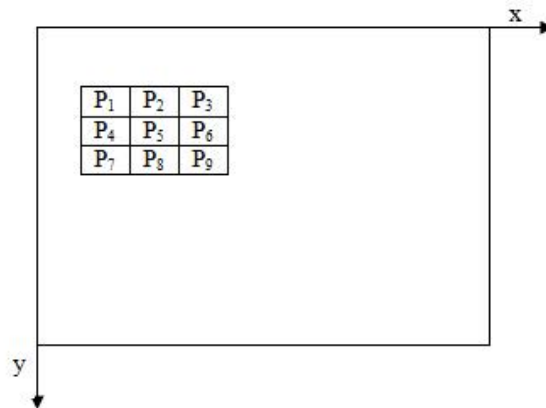


Figura 3.1: Imagem com origem do sistema de coordenadas e pixels de interesse em evidência.

Na Figura 3.2 está a representação da máscara a ser aplicada na imagem, definindo os “pesos” w_i de cada pixel.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figura 3.2: Máscara com pesos de cada pixel.

Esta máscara será aplicada na imagem, e na região de interesse teremos o novo valor para o pixel P_5 , salvo na nova imagem.

$$P_5^{Novo} = P_1w_1 + P_2w_2 + P_3w_3 + P_4w_4 + P_5w_5 + P_6w_6 + P_7w_7 + P_8w_8 + P_9w_9 \quad (3.1)$$

A aplicação da máscara será realizada em toda a imagem, percorrendo todas as suas linhas e colunas, salvando os novos valores dos pixels para formar a nova imagem. Atenção especial deve ser dada aos pontos das bordas para que não se tente lidar com pontos fora da imagem.

A máscara considerada como exemplo tem dimensão 3x3 (9 elementos), mas pode-se ter máscaras 5x5, 7x7, e assim por diante. Pode-se perceber que normalmente suas dimensões são ímpares. Isto ocorre para que haja um pixel central na região de interesse sob a máscara, para receber o novo valor calculado. No caso do exemplo anterior, este pixel era P_5 .

Os pesos w_i podem assumir qualquer valor inteiro, positivo ou negativo, de modo que a quantidade de máscaras possíveis é infinita. Porém, cada máscara é aplicada com uma finalidade, normalmente para acentuar ou eliminar alguma característica específica da imagem original. Desta forma, algumas máscaras são comumente utilizadas por terem sua eficiência comprovada em certa aplicação.

3.1 Filtros de Suavização

Os filtros de suavização são utilizados para “borrar” uma imagem ou eliminar detalhes desta. Esta primeira aplicação é um passo inicial útil quando o objetivo é a localização de grandes objetos na imagem, pois faz com que os pequenos detalhes sejam eliminados. Na segunda aplicação, dentre o detalhes que podem ser eliminados, pode-se citar eventuais ruídos na imagem.

A Figura 3.3 é um exemplo de um Filtro de Suavização. Estes filtros têm a característica de ter sempre todos os pesos positivos. No caso da Figura 3.3, como os pesos são unitários em todas as direções, pode-se afirmar que estas são igualmente importantes na composição do resultado final.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\frac{1}{49} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Figura 3.3: (a) Máscara de suavização 3x3, (b) Máscara de suavização 5x5, (c) Máscara de suavização 7x7

O resultado encontrado é dividido pela soma dos pesos da máscara aplicada, o que significa que estes filtros calculam a média ponderada dos pixels em questão. Assim, quanto maior a dimensão da máscara utilizada, maior será a suavização da imagem. Isto pode ser comprovado atentando-se paa os resultados obtidos na Figura 3.4.

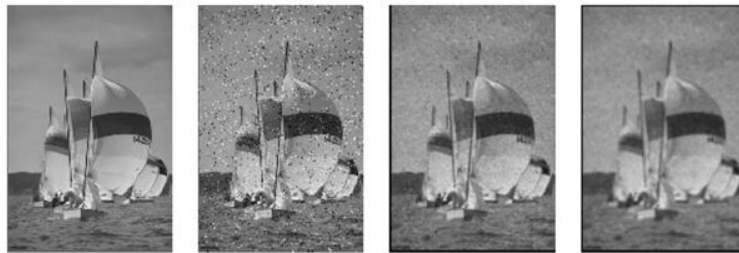


Figura 3.4: (a) Imagem original, (b) Imagem com ruídos, (c) Resultado da aplicação da máscara de suavização 3x3 sobre b, (d) Resultado da aplicação da máscara de suavização 5x5 sobre b.

3.2 Filtros de Acentuação

Os Filtros de Acentuação procuram acentuar pequenos detalhes na imagem. São, portanto, o inverso dos filtros de Suavização. A idéia deste filtro é enfatizar mudanças nos pixels. A Figura 3.5 mostra um exemplo deste tipo de filtro. Analisando os valores dos pesos, pode-se perceber que, quando a máscara esta sobre uma região com intensidade de pixels constante, o valor calculado vai à zero. Por outro lado, o valor calculado é máximo quando o pixel central sob a máscara difere bastante de seus pixels vizinhos. Esta é a situação que o Filtro de Acentuação procura evidenciar.

$\frac{1}{9} \times$

-1	-1	-1
-1	8	-1
-1	-1	-1

$\frac{1}{25} \times$

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	24	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

$\frac{1}{49} \times$

-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	48	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1

Figura 3.5: (a) Máscara de acentuação 3x3, (b) Máscara de acentuação 5x5, (c) Máscara de acentuação 7x7.

Comparando a resposta em termos da dimensão do filtro, temos que, quanto maior a dimensão da máscara, mais acentuada será a resposta obtida. Vide a Figura 3.6



Figura 3.6: (a) Imagem original, (b) Resultado da aplicação da Máscara de Acentuação.

3.3 Detecção de Bordas

O método de Detecção de Bordas é uma aplicação muito utilizada da Filtragem Espacial Linear. Seu objetivo é encontrar pixels de bordas de objetos na imagem, como por exemplo “quinas” de paredes e contornos de caixas. Ele se baseia no cálculo matemático aproximado do gradiente (primeiras derivadas) de uma função para criar as máscaras a serem aplicadas na imagem. A detecção é dividida em encontrar bordas horizontais e verticais, com uma máscara para cada finalidade. O resultado final será a soma dos dois componentes. Os “operadores de Sobel” são as máscaras mais utilizadas para essa finalidade, sendo apresentados na Figura 3.7.

Em ambas as direções, o princípio de funcionamento é o mesmo. Vamos analisar a detecção de bordas horizontais: quando a máscara está sobre uma região com intensidade de pixels relativamente constante, o valor calculado tende à zero.

$$S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Figura 3.7: (a) Máscara de Sobel para bordas horizontais, (b) Máscara de Sobel para bordas verticais.

Por outro lado, quando a máscara tem sua linha central $([0 \ 0 \ 0])$ exatamente em cima de uma borda, os pixels de um lado da borda (parte de cima da máscara) terão seus valores muito discrepantes em relação ao outro lado (parte de baixo da borda). Nesta situação, o valor calculado é máximo. Assim, os pixels pertencentes às bordas podem ser detectados como os que obtiveram valores resultantes maiores com a aplicação da máscara. Essa análise é análoga para a detecção vertical.

É comum, após a aplicação de uma máscara de detecção de bordas, se aplicar um *threshold* conveniente, para visualizar na imagem apenas os pixels de bordas detectados, com a mesma intensidade. As Figuras 3.8 e 3.9 exibem um exemplo da aplicação da Detecção de Bordas, bem como dos efeitos da utilização subsequente de um *threshold*.

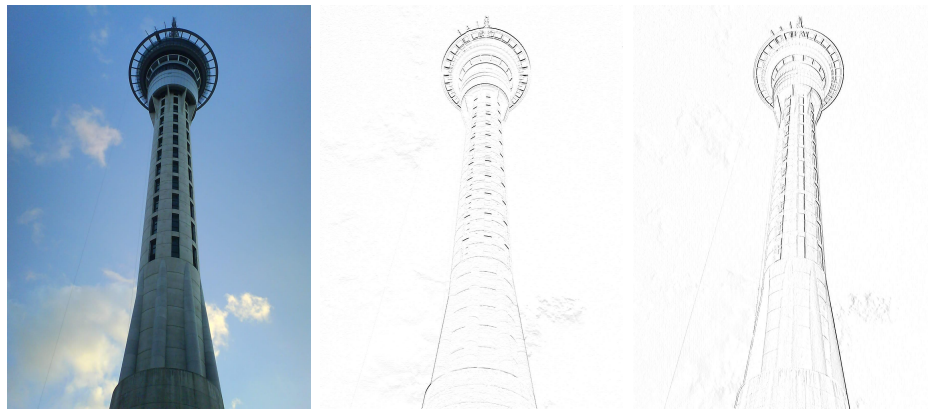


Figura 3.8: (a) Imagem original, (b) Aplicada a máscara de Sobel para bordas horizontais, (c) Aplicada a máscara de Sobel para bordas verticais.

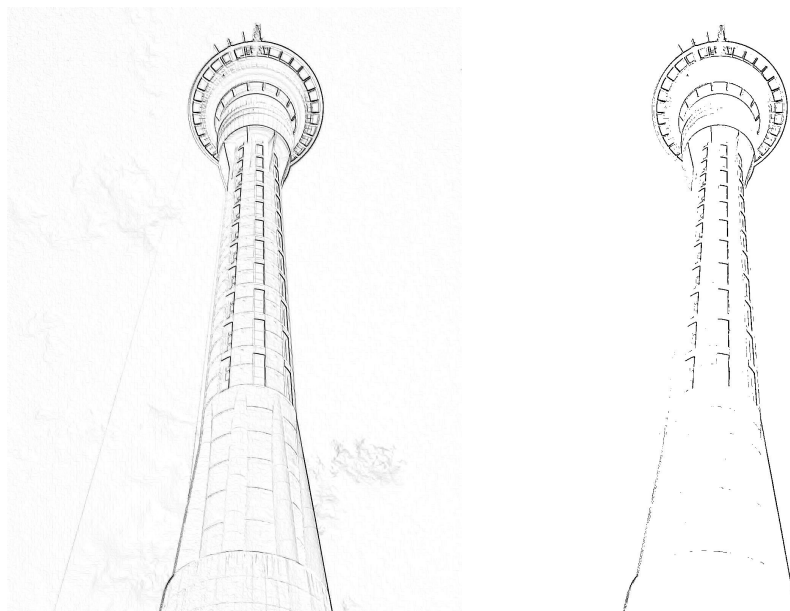


Figura 3.9: (a) Sobel total: composição dos resultados, (b) Sobel total, seguido pela aplicação do *threshold*.

4 Transformada de Hough

A Transformada de Hough é uma técnica para detecção de formas geométricas em imagens, dentre os quais se pode citar: linhas, círculos, triângulos, etc. A finalidade é, através da detecção dessas geometrias, obter informações de objetos reais presentes na imagem. Neste trabalho, será abordado seu formato para detecção de retas. A grande vantagem desta técnica frente a outras é sua robustez, ou seja, sua capacidade de obter bons resultados mesmo em imagens com presença de ruído ou com problemas de perda de informações em determinada região.

4.1 Detecção de Retas

A Transformada de Hough para detecção de retas utiliza os métodos de Filtragem Espacial Linear apresentados anteriormente. Os Filtros de Suavização ou Acen-tuação podem ser aplicados na imagem antes do início do processo, como uma forma de preparo da imagem, procurando melhorar a sensibilidade desta quanto às retas a serem detectadas. A aplicação de um desses filtros vai depender da sua conveniência na imagem em questão, sendo esta uma etapa que pode ser eliminada caso não seja necessária.

Iniciando efetivamente o processo de detecção de retas, parte-se da aplicação da máscara de detecção de bordas (Sobel) na imagem, seguida por um threshold conveniente. Com isso irão permanecer na imagem apenas os pixels de bordas detectados. Olhando a imagem neste estágio, pode parecer que obtivemos bordas formadas por linhas contínuas. Entretanto, na realidade, a detecção de bordas forneceu apenas conjuntos de pixels desconexos, sem nenhuma relação geométrica entre si. Cabe à Transformada de Hough encontrar essa relação.

Seguem algumas definições utilizadas no desenvolvimento da Transformada: denomina-se “Espaço Geométrico” (E.G.) o plano da imagem onde posteriormente as retas serão traçadas, enquanto que o “Espaço Paramétrico” (E.P.) é um plano auxiliar, explicado posteriormente. Nesta seção são demonstrados os passos tomados no desenvolvimento desta técnica através de um exemplo prático, de modo

a facilitar a compreensão.

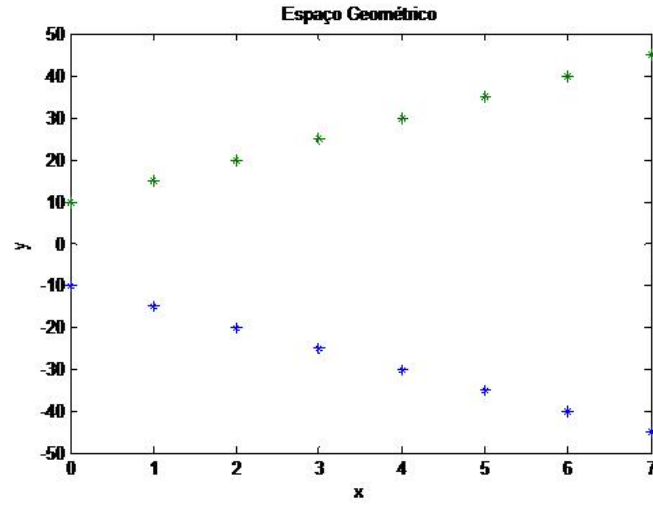


Figura 4.1: Exemplo de um conjunto de pixels de borda encontrados.

Tendo encontrado o conjunto de pixels de borda da imagem, pode-se dizer que uma reta, passando por um pixel i pertencente a este grupo, com coordenadas (x_i, y_i) , tem a seguinte representação matemática.

$$y_i = a.x_i - b \quad (4.1)$$

Pode-se rearranjar essa fórmula para ter o parâmetro b em função de a .

$$b = a.x_i - y_i \quad (4.2)$$

Assim, para um dado ponto (x_i, y_i) , variando o valor de “ a ” dentro de um intervalo definido, encontramos o valor de “ b ” correspondente, o que resulta em uma reta, representada no Espaço Paramétrico. Este espaço auxiliar tem esse nome por ser função dos parâmetros a e b da reta. Assim, cada pixel de borda será mapeado em uma reta neste espaço.

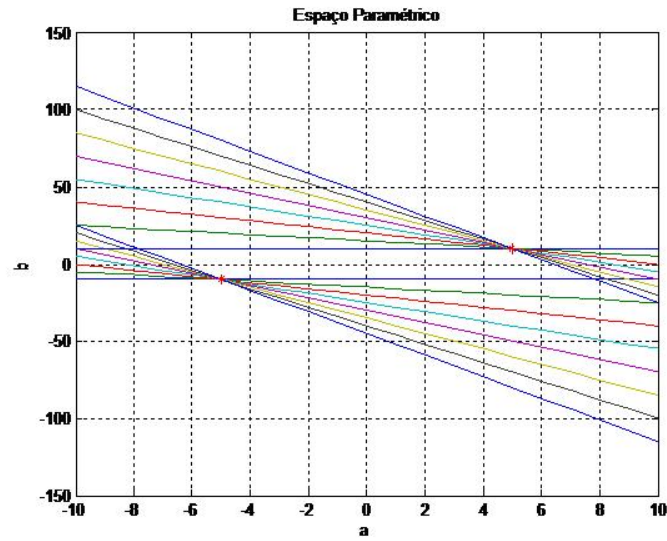


Figura 4.2: Mapeamento dos pixels no Espaço Paramétrico.

Tendo todos os pixels de borda mapeados no Espaço Paramétrico, a intersecção de retas em um ponto neste espaço fornece um conjunto de parâmetros (a,b) de uma reta que, no Espaço Geométrico, passa pelos pontos que geraram as retas interseccionadas no Espaço Paramétrico. Desta forma, quanto maior o número de intersecções em um dado ponto do Espaço Paramétrico, maior o número de pixels que estarão presentes na reta definida pelos parâmetros (a,b) ao ser traçada no Espaço Geométrico. Por isso, a estratégia adotada pela Transformada de Hough é procurar os pontos onde ocorre o maior número de intersecções entre as retas no Espaço Paramétrico, e traçar as retas no Espaço Geométrico com os parâmetros (a,b) encontrados.

Normalmente é definido um valor mínimo para o número de intersecções no Espaço Paramétrico para que os parâmetros (a,b) sejam selecionados. Na Figura 4.2, os pontos marcados em vermelho ficaram acima deste limite, sendo os escolhidos para definir as retas no Espaço Geométrico, e o resultado das retas traçadas pela Transformada de Hough é apresentado na Figura 4.3.

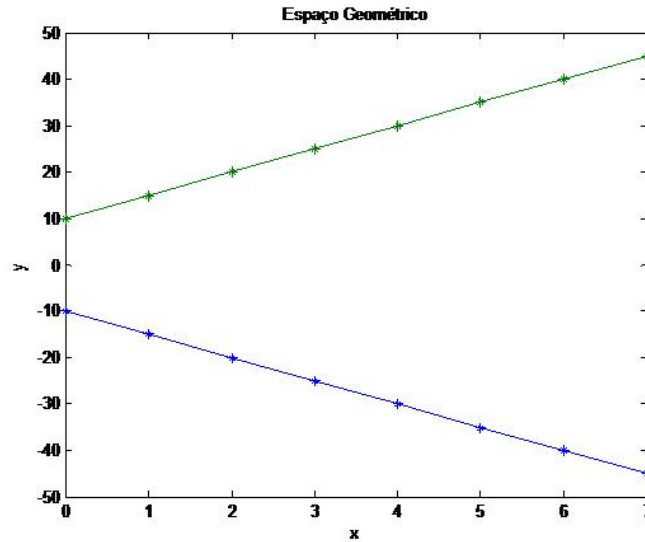


Figura 4.3: Traçamento das retas encontradas no Espaço Geométrico.

Resumindo, a Transformada de Hough para retas possui as seguintes características:

- Cada ponto no Espaço Geométrico produz uma reta no Espaço Paramétrico.
- Cada ponto encontrado no Espaço Paramétrico produz uma reta no Espaço Geométrico.
- O ponto onde k linhas se interseccionam no Espaço Paramétrico corresponde a uma reta através de k pontos no Espaço Geométrico.

4.2 Detecção de Retas - Forma Normal

A formulação da Transformada de Hough feita anteriormente possui uma desvantagem considerável. Utilizando a formulação clássica de uma reta (Equação 4.1) no Espaço Geométrico, os valores do parâmetro a tornam-se muito elevados para retas com elevada inclinação. Isso implica em calcular retas no Espaço Paramétrico para um intervalo muito grande de a , o que torna a implementação computacionalmente inviável. Para evitar este problema, adota-se uma forma alternativa de representar matematicamente uma reta no Espaço Geométrico, a Forma Normal:

$$\rho = x.\cos(\theta) + y.\sin(\theta) \quad (4.3)$$

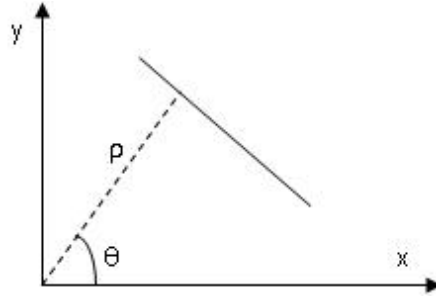


Figura 4.4: Representação de uma reta na Forma Normal no E. G.

Nesta formulação, ρ é a menor distância da origem do sistema de coordenadas até a reta e θ é o ângulo dele formado com o eixo x. Assim, o conjunto de parâmetros (ρ, θ) é capaz de identificar uma única reta, assim como faziam os parâmetros (a, b) . A vantagem está no fato de os dois parâmetros estarem definidos em intervalos regulares: $0 \leq \rho \leq R$ e $0 \leq \theta \leq 2\pi$, onde R é a maior distância que a reta pode se encontrar da origem. Em uma imagem, R será o valor da sua diagonal.

Utilizando esta formulação na Transformada de Hough, considera-se que cada pixel de borda (x_i, y_i) pertence a uma reta no Espaço Geométrico representada por:

$$\rho = x_i \cdot \cos(\theta) + y_i \cdot \sin(\theta) \quad (4.4)$$

No Espaço Paramétrico, para cada ponto (x_i, y_i) , iremos variar θ de 0 a 2π e calcular os valores de ρ . Agora o Espaço Paramétrico será em função de ρ e θ , não mais de a e b , e cada ponto no Espaço Geométrico resultará em uma curva, não mais uma reta, no Espaço Paramétrico. O restante do desenvolvimento da Transformada de Hough é semelhante ao apresentado anteriormente: mapeando todos os pontos de bordas em curvas no Espaço Paramétrico, determina-se as intersecções do maior número de curvas e são obtidos os parâmetros (ρ, θ) que irão formar as retas no Espaço Geométrico.

5 Aritmética Intervalar

A Aritmética Intervalar ^[3] é um método que estende a aritmética usual sobre números para trabalhar com intervalos destes. Seu domínio de aplicação é bastante variado, inclusive, como neste caso, para aplicações em Processamento de Imagens.

5.1 Definições Básicas

A aritmética Intervalar trabalha com operações sobre intervalos. Estes são completamente definidos por seu tipo (número inteiro, real, etc) e por seus limites inferior e superior. Para um intervalo real, por exemplo, sua forma de representação é a seguinte.

$$[a, b] = \{x/a \leq x \leq b\} \quad (5.1)$$

Seguindo esta notação, pode-se definir as quatro operações aritméticas básicas aplicadas sobre os intervalos.

$$[a, b] + [c, d] = [a + c, b + d] \quad (5.2)$$

$$[a, b] - [c, d] = [a - d, b - c] \quad (5.3)$$

$$[a, b] \cdot [c, d] = [\min(a \cdot c, a \cdot d, b \cdot c, b \cdot d), \max(a \cdot c, a \cdot d, b \cdot c, b \cdot d)] \quad (5.4)$$

$$[a, b] / [c, d] = [\min(a/c, a/d, b/c, b/d), \max(a/c, a/d, b/c, b/d)] \quad (5.5)$$

Como esta se lidando com intervalos, a comparação destes com números ou outros intervalos torna-se não-trivial, sendo uma tarefa que exige o estabelecimento de procedimentos. A comparação entre dois intervalos, por exemplo $[a, b]$ e $[c, d]$, pode resultar em três possibilidades: “certamente iguais”, “possivelmente iguais” e “certamente não-iguais”. Eles são “certamente iguais” se $a = c$ e $b = d$. São “certamente não-iguais” se a intersecção entre eles for nula. Já o caso intermediário, onde a intersecção entre eles é não-nula, resulta em “possivelmente iguais”. Cabe ao usuário da Aritmética Intervalar decidir o tratamento a ser dado a este

caso intermediário. Tendo uma abordagem conservadora, por exemplo, pode-se considerar os resultados de “possivelmente iguais” como “certamente não-iguais”.

5.2 Aplicação no Processamento de Imagens

As imagens digitais, objetos de estudo das técnicas de Processamento de Imagens, possuem um erro intrínseco à sua forma de obtenção. Isto porque, ao gerarmos uma imagem digital de um objeto real, há uma limitação na resolução com que podemos representá-lo digitalmente: o pixel. Cada pixel utiliza um único valor de intensidade para representar um intervalo de cores do objeto real. Desta forma, cada pixel da imagem incorre em um erro ao aproximar um intervalo contínuo em um único valor de intensidade.

Neste contexto, a aplicação da aritmética intervalar a técnicas de Processamento de Imagens busca considerar este erro dos pixels ao transformá-los em intervalos, aumentando a robustez das técnicas utilizadas. Além disso, o uso da Aritmética Intervalar possui a vantagem de evitar erros de arredondamento, que ocorreriam com cálculos feitos em ponto flutuante.

6 Resultados Práticos Obtidos

A Transformada de Hough para detecção de retas foi implementada em linguagem de programação C++ utilizando como base uma biblioteca de processamento de imagens chamada CxImage. Esta biblioteca possui uma interface para lidar com imagens e alguns métodos de processamento de imagens, como a aplicação de filtros espaciais ^[5-7]. Isto facilitou o ambiente de programação e testes realizados com imagens, porém os métodos implementados para a aplicação da Transformada de Hough não fazem parte da biblioteca, e tiveram que ser totalmente desenvolvidos. Optou-se pela representação das retas na forma normal, devido às vantagens explicadas anteriormente. As variáveis utilizadas na Transformada foram associadas à Aritmética Intervalar. Para isto foi associada à implementação outra biblioteca, chamada Boost, voltada ao uso da Aritmética Intervalar.

A seguir será apresentado um exemplo utilizando o programa. Pode-se perceber que ele segue a ordem teórica dos eventos já apresentados.

A Figura 6.1 exibe a imagem inicial que será submetida ao programa. O objetivo é encontrar as principais retas desta imagem. O programa é capaz de trabalhar com imagens em qualquer formato, coloridas ou não, sem perda de qualidade no resultado.



Figura 6.1: Imagem de teste original.

Em seguida aplica-se a máscara de Sobel, seguido por um threshold conveniente, para detecção dos pixels das bordas. O resultado obtido encontra-se na Figura 6.2.

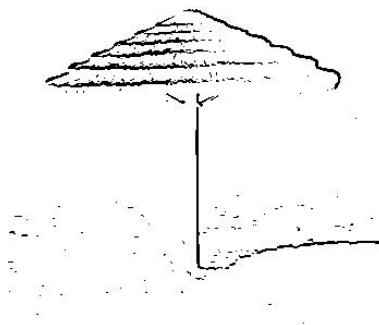


Figura 6.2: Pixels de borda encontrados aplicando-se a máscara de Sobel seguido por um threshold.

Os pixels das bordas dão origem a curvas no Espaço Paramétrico, o que é representado na imagem abaixo. Em seguida percorre-se este espaço e selecionam-se os parâmetros ρ e θ dos pontos com o maior número de intersecções. Na Figura 6.3, cada pixel de borda deu origem a uma curva em cinza e assim, quanto mais próximo do preto, maior é o número de intersecções naquele ponto. Os pontos cujos parâmetros foram escolhidos, por estarem acima de um limite definido, estão destacados por meio de um quadrado em azul.

Por fim, as retas com os parâmetros escolhidos são traçadas no Espaço Geo-

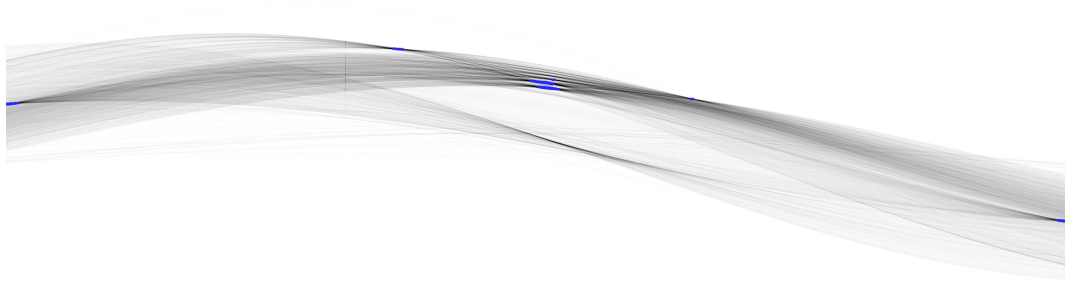


Figura 6.3: Espaço Paramétrico computado (ρ na vertical e θ na horizontal).

métrico. Primeiramente, foram traçadas as retas encontradas em azul por cima dos pixels de borda anteriormente encontrados, para se poder fazer uma comparação dos resultados obtidos. Também é fornecida uma imagem apenas com as retas obtidas traçadas sobre uma imagem em branco, sendo este o resultado final da Transformada de Hough.

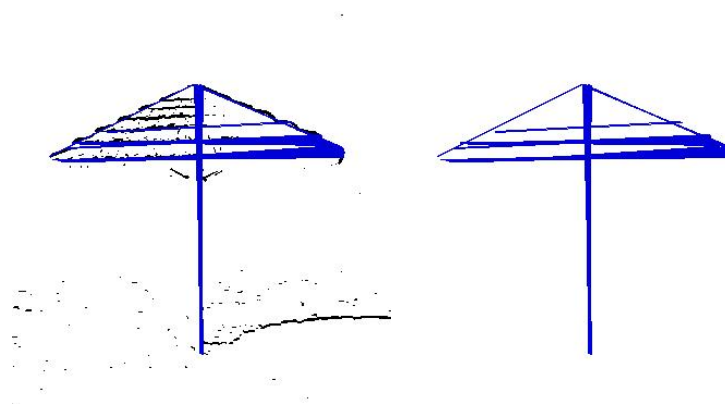


Figura 6.4: (a) Retas encontradas traçadas em azul no E.G. e sobrepostas aos pixels de bordas, (b) Apenas as retas encontradas na imagem.

Analisando o resultado final percebe-se que foram traçados apenas os segmentos de reta de interesse, e não uma reta inteira, como proposto originalmente pela Transformada. Esta é uma funcionalidade a mais implementada para garantir que a detecção das arestas de objetos seja precisa, de modo a se poder inferir corretamente suas dimensões na imagem.

Será apresentado um outro exemplo para efeitos de comparação.



Figura 6.5: Imagem de teste original.

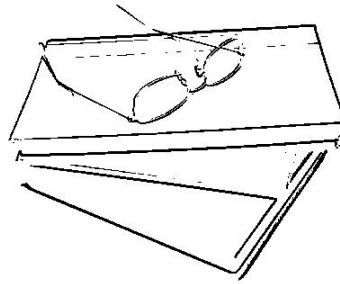


Figura 6.6: Pixels de borda encontrados aplicando-se a máscara de Sobel seguido por um threshold.



Figura 6.7: Espaço Paramétrico computado (ρ na vertical e θ na horizontal).

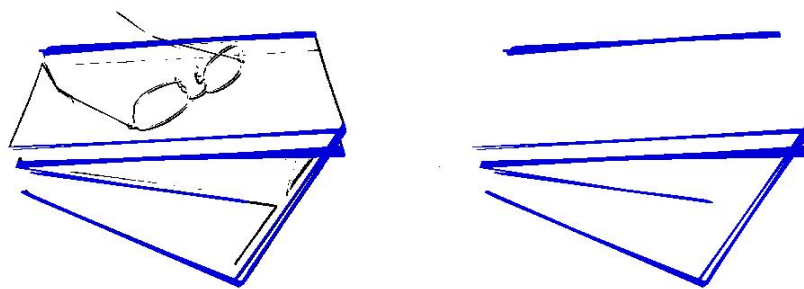


Figura 6.8: (a) Retas encontradas traçadas em azul no E.G. e sobrepostas aos pixels de bordas, (b) Apenas as retas encontradas na imagem.

7 Conclusões

A Transformada de Hough mostrou ser um método poderoso para reconhecimento de padrões em imagens e sua eficácia pôde ser comprovada por meio da sua efetiva implementação. O programa desenvolvido foi sendo melhorado ao longo do tempo até alcançar a sua versão final, apresentada neste trabalho. Ele é robusto e consegue determinar com precisão a posição e tamanho dos segmentos de reta a serem traçados, o que evidencia sua grande utilidade para determinação das retas principais da imagem.

Referências

- 1 AGUADO, A. S.; MONTIEL, E.; NIXON, M. S. Invariant characterisation of the Hough transform for pose estimation of arbitrary shapes. *Pattern Recognition Journal*, n. 35, p. 1083-1097, 2002.
- 2 BONCI, A.; LEO, T.; LONGHI, S. A Bayesian Approach to the Hough Transform for Line Detection. *IEEE Transactions on Systems*, v. 35, n. 6, p. 945-955, 2005.
- 3 BRONNIMANN, H.; MELQUIOND, G.; PION, S. The design of the Boost interval arithmetic library. Elsevier Science, 2004.
- 4 JI, Qiang; HARALICK, R. M. Error propagation or the Hough transform. *Pattern Recognition Letters*, n. 22, p. 813-823, 2001.
- 5 OLSON, C. F. Constrained Hough Transforms for Curve Detection. *Computer Vision and Image Understanding*, v. 73, n. 3, p. 329-345, 1999.
- 6 O'GORMAN, F.; CLOWES, M. B. Finding Picture Edges Through Collinearity of Feature Points. *IEEE Transactions on Computers*, v. C-25, n. 4, p. 449-456, 1976.
- 7 RHODY, Harvey. Digital Image Processing. New York: Rochester Institute of Technology, 2005.
- 8 TAVARES, R. S. Visualização e Análise de Seqüências de Imagens Temporais de Ressonância Magnética do Pulmão Utilizando Aritmética Intervalar. 2007. 117p. Monografia de Conclusão de Curso - Escola Politécnica, Universidade de São Paulo, São Paulo, 2007.