UNIVERSIDADE FEDERAL FLUMINENSE

BIANCA MENDES DACZKOVSKI

Um estudo sobre o espaço de parâmetros linear por partes na Transformada de Hough

NITERÓI

2024

BIANCA MENDES DACZKOVSKI

Um estudo sobre o espaço de parâmetros linear por partes na Transformada de Hough

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação.

Orientador:
LEANDRO AUGUSTO FRATA FERNANDES

NITERÓI

2024

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

D117e Daczkovski, Bianca Mendes
Um estudo sobre o espaço de parâmetros linear por partes na Transformada de Hough / Bianca Mendes Daczkovski. - 2024.
67 f.: il.

Orientador: Leandro Augusto Frata Fernandes.
Dissertação (mestrado)-Universidade Federal Fluminense, Instituto de Computação, Niterói, 2024.

1. Computação visual. 2. Processamento digital de imagem. 3. Reconhecimento de padrão. 4. Triangulação. 5. Produção intelectual. I. Fernandes, Leandro Augusto Frata, orientador. II. Universidade Federal Fluminense. Instituto de Computação. III. Título.

CDD - XXX

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

BIANCA MENDES DACZKOVSKI

Um estudo sobre o espaço de parâmetros linear por partes na Transformada de Hough

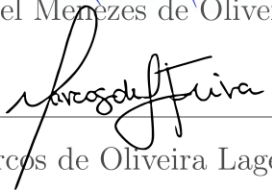
Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação.

Aprovada em Fevereiro de 2024.

BANCA EXAMINADORA

Prof. Leandro Augusto Frata Fernandes - Orientador, UFF

Prof. Manuel Menezes de Oliveira Neto, UFRGS



Prof. Marcos de Oliveira Lage Ferreira, UFF

Niterói

2024

Agradecimentos

Agradeço primeiramente à minha família e meus amigos, por todo o apoio e incentivo que me deram e pelo carinho e compreensão que tiveram comigo em todos os aspectos de minha vida durante todos esses anos e sei que futuramente também. Vocês são muito importantes para mim e contribuíram imensamente em minha formação. Sem vocês nada disso seria possível.

Agradeço à todos os professores que passaram pela minha vida e que me ajudaram a trilhar o meu caminho, em especial os professores Ralph Costa e Raissa Mendes, pelos quais tive o grande prazer de ser orientada durante minha graduação, e o professor Leandro Fernandes, que me acolheu e orientou desde o início do mestrado com muita dedicação e compreensão. Vocês me ensinaram com grande afinco e estavam sempre dispostos a ajudar. Possuo grande respeito, carinho e admiração por todos.

Agradeço também aos que atuam e atuaram no Programa de Pós-Graduação em Computação da UFF, por terem mantido a excelência do ensino e da infraestrutura durante estes anos tão difíceis que tivemos, e ao CNPQ pelo auxílio financeiro fornecido.

Agradeço, por fim, a Deus, por estar ao meu lado em toda minha trajetória e por ter colocado essas pessoas maravilhosas em minha vida.

Resumo

A Transformada de Hough (HT) é comumente utilizada para detectar formas facilmente parametrizadas em imagens relativamente pequenas. Esta técnica define um conjunto de células acumuladoras que representa o espaço de parâmetros discretizado da forma a ser encontrada, seguido por um processo de votação onde os dados de entrada depositam votos nestas células que correspondem às instâncias das formas que os contém e, por fim, realiza a busca por máximos locais no resultado da votação, sendo a localização desses máximos a representação das instâncias de formas geométricas mais aparentes no dado de entrada. No entanto, este método torna-se impraticável para formas com muitos parâmetros ou imagens consideravelmente grandes devido ao aumento da quantidade de memória necessária com base na resolução da imagem de entrada e à alta dimensionalidade do espaço de parâmetros, tornando-a inviável em cenários específicos.

Neste trabalho, é introduzida uma técnica que utiliza a HT para gerar um espaço de parâmetros formado por células acumuladoras que não caracteriza uma discretização regular do espaço de parâmetros, mas que estabelece uma variedade linear por partes por meio da triangulação de Delaunay e detecta máximos locais através da definição de Banchoff, permitindo assim a identificação de formas geométricas. Este método reduz tanto a quantidade de memória necessária quanto o tempo de processamento, tornando a HT viável para a detecção de linhas em imagens com mais de 1 bilhão de pixels ou a identificação de formas com espaço de parâmetros de maior dimensionalidade.

Palavras-chave: Transformada de Hough, Detecção de Linhas, Variedade Linear por Partes, Pontos Críticos.

Abstract

The Hough Transform (HT) is commonly employed for detecting easily parameterized shapes in relatively small images. This technique defines a set of accumulator cell that represent the discretized parameter space of the shape to be found, followed by a voting process where input data deposit votes in these cells corresponding to instances of the shapes they contain, and finally, searches for local maxima in the voting result, with the locations of these maxima representing the instances of geometric shapes most apparent in the input data. However, this method becomes impractical for shapes with many parameters or considerably large images due to the increasing memory requirements based on the input image's resolution and the high dimensionality of the parameter space, rendering it unfeasible in specific scenarios.

In this work, a technique is introduced that leverages the HT to create a parameter space composed of accumulator cells that does not characterize a regular discretization of the parameter space, but establishes a piecewise linear manifold through Delaunay triangulation and detecting local maxima using the Banchoff definition, thus enabling the identification of geometric shapes. This method reduces both memory requirements and processing time, thereby making the HT viable for line detection in images exceeding one billion pixels or detecting shapes with higher-dimensional parameter spaces.

Keywords: Hough Transformation, Line Detection, Piecewise Linear Manifold, Critical Points.

Lista de Figuras

1 Espaço de parâmetros discretizado bidimensional. 12

2 Processo de votação de dois pontos para detecção de retas. 13

3 Reta equivalente ao ponto de máximo local (em preto). 13

4 Imagem I: dado em tons de cinza com ruído. 19

5 Imagens intermediárias G_x (esquerda) e G_y (direita). 23

6 Magnitude G (esquerda) e direção Θ (direita) do gradiente. 24

7 Imagens intermediárias G_{xx} (esquerda), G_{yy} (centro) e G_{xy} (direita). . 25

8 Magnitude da curvatura k. 25

9 Imagem original (esquerda) e imagem suavizada com o filtro Gaussiano ($\sigma = 3$ e kernel 5×5) (direita). 26

10 Imagem suavizada (esquerda) e magnitude do gradiente (direita). . . 27

11 Magnitude do gradiente (esquerda) e resultado da supressão dos não-máximos (direita). 27

12 Supressão dos não-máximos (esquerda) e resultado dos valores limites ($val_{max} = 20$ e $val_{min} = 4$): pixels descartados em cinza e remanescentes em branco (direita). 28

13 Resultado final do método de detecção de bordas. 28

14 Processo de votação de um ponto da imagem. 31

15 Pixels de borda correspondentes à reta vermelha. 31

16 Etapas da HT para detecção de retas. 32

17 Etapas da OHT para detecção de retas. 33

18 Etapas do método híbrido para detecção de retas. 34

19 Homeomorfismo entre cubo e esfera (verde); esfera furada, hemisfério, e disco aberto (cinza); e torus e caneca (vermelho). 36

20 Simplexos de dimensões 0 (ponto), 1 (segmento de linha), 2 (triângulo) e 3 (tetraedro), respectivamente. 36

21 Triângulo fechado e seus componentes. 37

22 Estrela do vértice. 38

23 Homeomorfismo entre um 3-simplexo e uma esfera S^2. 38

24 Classificação dos pontos em uma superfície em mínimo (a), máximo (b), regular (c), sela (d) e sela de macaco (e). 40

25 Espaço de parâmetros LPP (esquerda) e retas correspondentes na imagem (direita). 44

26 Espaço de Parâmetros LPP de máximos em verde e centros de agrupamentos com $votes \geq v_{min}$ em vermelho (esquerda), e linhas correspondentes na imagem (direita). 46

27 Processo de geração das imagens. 48

28 Retas identificadas pela abordagem proposta usando os valores padrão definidos para os limiares κ_{max} e v_{min}. 49

29 Pico de memória vs. tamanho da imagem para a DHT com e sem o limiar κ_{max}. 50

30 Tempo total de processamento vs. tamanho da imagem para a DHT com e sem o limiar κ_{max}. 50

31 Tempo percentual cada etapa do processamento vs. tamanho da imagem para a DHT. 52

32 Resultado da DHT com os limiares ajustados de acordo com a legenda de cada imagem. 53

33 Pico de memória vs. tamanho da imagem para as técnicas SHT, KHT e DHT. 55

34 Tempo total de processamento vs. tamanho da imagem para as técnicas SHT, KHT e DHT. 55

35 Comparação das técnicas para imagem de tamanho 500×500. 56

36 Comparação das técnicas para imagem de tamanho 1000×1000. . . . 56

37 Comparação das técnicas para imagem de tamanho 2000×2000. . . . 56

38 Comparação das técnicas para imagem de tamanho 4000×4000. . . . 57

39 Comparação das técnicas para imagem de tamanho 8000×8000. . . . 57

40 Comparação das técnicas para imagem de tamanho 16000×16000. . . 57

Lista de Tabelas

1 Memória de armazenamento utilizada para cada tamanho de imagem. 48

2 Valores médios de pico de memória e tempo de processamento obtidos nos testes da DHT com e sem o limite κ_{max} para a mesma imagem de tamanhos variados. 51

3 Valores médios de porcentagem do tempo de processamento de cada etapa em relação ao tempo total de processamento obtidos nos testes da DHT para a mesma imagem de tamanhos variados. 52

4 Valores médios de pico de memória e tempo de processamento obtidos nos tesdes da DHT, KHT e GHT para a mesma imagem de tamanhos variados. 54

Lista de Algoritmos

1 Processo de votação. 42

2 Detecção de máximos. 43

3 Agrupamento de máximos. 45

Lista de Abreviaturas e Siglas

DHT Transformada de Hough baseada em Dicionário

HT Transformada de Hough

KHT Transformada de Hough baseada em Kernel

LPP linear por partes

SHT Transformada de Hough padrão

Sumário

1 Introdução **12**

1.1 Abordagens relacionadas 14

1.2 Ideia central 15

1.3 Desafios 15

1.4 Resultados 16

1.5 Organização do trabalho 16

2 Fundamentos Teóricos **18**

2.1 Processamento digital de imagens 18

2.1.1 Conceitos básicos 18

2.1.1.1 Operadores de vizinhança lineares 20

2.1.2 Derivadas da imagem 21

2.1.2.1 Gradiente 22

2.1.2.2 Curvatura 24

2.1.3 Detecção de bordas 26

2.2 Transformada de Hough 28

2.2.1 Transformada de Hough para linhas retas 30

2.2.2 Transformada de Hough orientada 32

2.3 Topologia algébrica 34

2.3.1 Espaço topológico 34

2.3.2 Complexo simplicial 35

2.3.3 Variedade linear por partes 38
2.3.4 Pontos críticos geométricos 39
2.4 Discussões 40

3 Transformada de Hough Linear por Partes para Linhas Retas 41
3.1 Votação nas células acumuladoras 41
3.2 Detecção de máximos 42
3.3 Agrupamento de máximos 44
3.4 Outras formas geométricas 45

4 Experimentos 47
4.1 Metodologia 47
4.2 Resultados e discussões 48
4.2.1 Teste dos limiares 48
4.2.2 Tempo por etapa 51
4.2.3 Detecção de retas 52
4.2.4 Comparação entre técnicas 54

5 Conclusões e Trabalhos Futuros 58
5.1 Discussões 58
5.2 Limitações 59
5.3 Trabalhos futuros 59

REFERÊNCIAS 60

1 Introdução

A detecção de retas em imagens é um tema recorrente em visão computacional. Algumas de suas aplicações incluem modelagem baseada em imagem (DEBEVEC; TAYLOR; MALIK, 1996; HARTLEY; ZISSERMAN, 2003) e interfaces de usuário (FERREIRA et al., 2004), análise de imagem e compreensão de cena (LEE, J. et al., 2017), sensoriamento remoto (KARNIELI et al., 1996), reconhecimento de texto (KOO, 2016; EL BAHI; ZATNI, 2019), e inspeção industrial (BOLLAND et al., 1997; LEE, H.-J. et al., 2001).

A detecção de formas facilmente parametrizáveis, como retas, parábolas e elipses, em imagens formadas por uma série de pontos discretos é geralmente realizada usando a Transformada de Hough (HT, do inglês *Hough Transform*) (HOUGH, 1962; DUDA; HART, 1972; SKLANSKY, 1978; BENNETT; BURRIDGE; SAITO, 1999; FERNANDES; OLIVEIRA, 2008). Este método consiste em:

1. Definir um conjunto de células acumuladoras que representa o espaço de parâmetros discretizado da forma a ser encontrada (Figura 1);

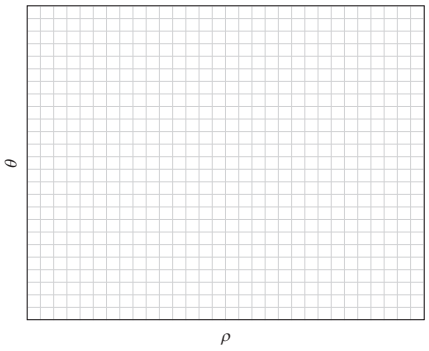


Figura 1: Espaço de parâmetros discretizado bidimensional.

2. Realizar o processo de votação, onde os dados de entrada depositam votos nestas células que correspondem às instâncias das formas que os contém (Fi-

gura 2);

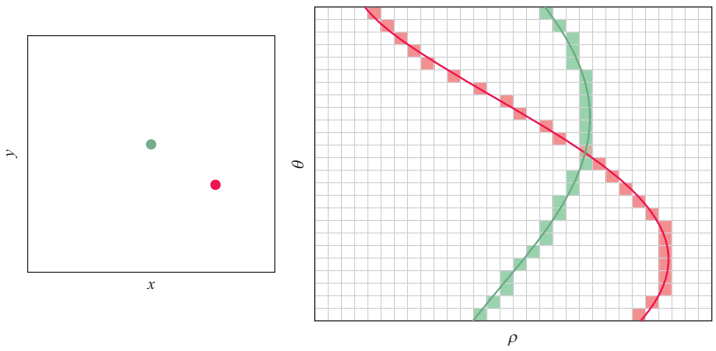


Figura 2: Processo de votação de dois pontos para detecção de retas.

3. Buscar por máximos locais no resultado da votação, sendo a localização desses máximos a representação das instâncias de formas geométricas mais aparentes no dado de entrada (Figura 3).

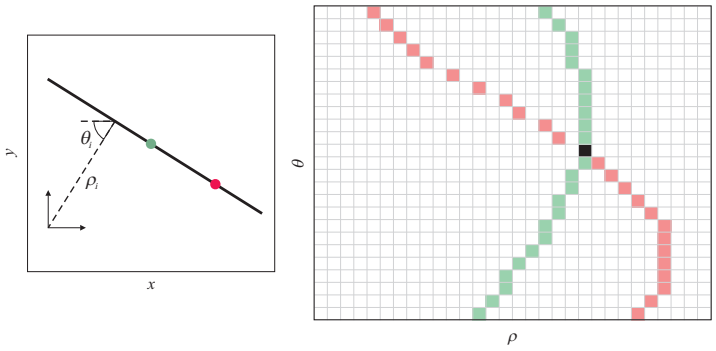


Figura 3: Reta equivalente ao ponto de máximo local (em preto).

Esta técnica tem suas vantagens no tratamento de ruído e descontinuidades em uma imagem. No entanto, possui limitações inerentes, como alto tempo de processamento, requisitos pesados de memória, poucos votos para retas curtas e a incapacidade de preservar a conectividade de pixels em bordas. Essas limitações estão relacionadas ao espaço de parâmetros: como ocorre a votação, sua discretização e seu tamanho, que está diretamente relacionado ao tamanho da imagem. Quando o espaço de parâmetros é discretizado de forma mais espaçada, pode ocorrer fusão imprópria e detecção imprecisa de retas, enquanto uma discretização mais densa aumenta significativamente o tempo de processamento e o uso de memória. Como resultado, o uso da HT para detecção de retas muitas vezes é restrito a imagens menores.

1.1 Abordagens relacionadas

Hough (1962) foi o primeiro a explorar a dualidade entre pontos e linhas através do mapeamento de pixels para retas em um espaço de parâmetros bidimensional usando uma parametrização de inclinação e interceptação (a.k.a. equação reduzida da reta). Nisso, o processo de detecção de retas foi convertido em um processo de detecção dos pontos de intersecção mais frequentes dessas retas, considerando um raio de tolerância na comparação das localizações, que representam as retas que melhor se ajustam ao conjunto de pixels de entrada. Em seu livro clássico sobre processamento de imagens, Rosenfeld (1969) propôs o uso de um mapa discreto e regular de acumuladores para representar o espaço de parâmetros da HT, facilitando o gerenciamento das intersecções de retas. Infelizmente, a utilização da parametrização de inclinação e interceptação não pode representar retas verticais.

Duda e Hart (1972) introduziram a parametrização de ângulo-raio com base na equação normal da reta, o que permitiu a representação de retas verticais e reduziu a memória necessária. Essa abordagem mapeia cada ponto de borda como uma senoide no espaço de parâmetros $\rho\theta$, abrangendo todos os possíveis valores de θ, onde ρ é a menor distância da reta a origem do sistema de coordenadas da imagem e $\theta \in [0, \pi]$ é o ângulo entre o eixo x e a normal à reta. Os picos formados nas interseções das senoides representam as retas que melhor se ajustam ao conjunto de pixels de entrada. Neste caso, a discretização da imagem e do espaço de parâmetros faz com que as linhas senoidais "interajam" em muitas células, criando picos secundários. O uso da parametrização normal para mapear senoides no espaço de parâmetros com informações de gradiente tornou-se a versão universalmente utilizada da HT para linhas retas.

Fernandes e Oliveira (2008) apresentaram um esquema de votação aprimorado para a HT que opera em aglomerados de pixels aproximadamente colineares, permitindo desempenho em tempo real mesmo em imagens relativamente grandes, sendo empregado como parte da biblioteca de processamento de imagens do Observatório Vera C. Rubin, anteriormente referido como o Large Synoptic Survey Telescope (LSST), para detectar passagens de satélites artificiais. A abordagem melhora significativamente o desempenho do esquema de votação, produz um mapa de votação mais limpo e torna a HT mais robusta para detectar linhas espúrias.

1.2 Ideia central

Uma maneira de representar uma superfície contínua é através de estruturas topológicas definidas pela triangulação de pontos, conhecidas como 2-variedade linear por partes (LPP). Essas são comumente aplicadas em vários domínios, incluindo dados estéreo (BRUZZONE et al., 1992), modelagem de terreno (KIM; CHO; CHO, 1998), redes (LIEBEHERR; NAHAS; SI, 2002), reconstrução (RODRÍGUEZ et al., 2000), agrupamento espacial (YANG; CUI, 2010), entre outros. Seu uso frequentemente envolve definição de forma de objeto (ZAHN, 1971; WATSON, 1981; SIBSON, 1978), interpolação de pontos (FRANKE, 1982) ou mesmo redução de pontos (FAUGERAS et al., 1984).

Uma 2-variedade LPP preserva informações relevantes como proximidade, limites, conectividade e continuidade por meio de relações entre conjuntos (WILLARD, 2004), possibilitando o cálculo de propriedades geométricas mais simples, como área, volume e eixos de inércia (BOISSONNAT, 1984), e até mesmo aspectos mais complexos como gradientes e pontos críticos (BANCHOFF, 1970).

Construindo sobre as vantagens de empregar uma variedade LPP e visando abordar a questão do alto uso de memória e tempo de processamento na HT tradicional, especialmente para imagens com mais de 1 bilhão de pixels, propomos neste estudo uma Transformada de Hough baseada em Dicionário (DHT). Esta abordagem da HT gera um espaço de parâmetros formado apenas por células acumuladoras com votos suficientes, o que torna possível seu armazenamento em um dicionário, dispensando a necessidade da representação do espaço de parâmetros como um mapa de acumuladores mantido de forma completa na memória. Em seguida, uma variedade LPP é definida por meio da triangulação dessas células, o que permite a identificação de máximos locais de votos seguindo a definição de Banchoff. Por fim, esses máximos são agrupados de tal forma que os agrupamentos mais relevantes correspondem a retas na imagem.

1.3 Desafios

O principal desafio enfrentado foi formular um método eficiente capaz de extrair apenas as informações cruciais do espaço de parâmetros, visando a redução signifi-

cativa do uso de memória. Diversas técnicas foram exploradas e refinadas ao longo do processo até chegar no método apresentado neste trabalho.

Outra dificuldade significativa foi a implementação prática da abordagem para a condução dos experimentos. Isso envolveu a criação de muitos códigos, utilizando diversas bibliotecas diferentes e a realização de inúmeros testes para chegar à versão final adotada neste estudo. Ainda assim, definitivamente há espaço para aprimoramentos contínuos e refinamentos no desenvolvimento do método.

1.4 Resultados

Esta dissertação apresenta alguns resultados originais que incluem:

- Um método de seleção prévia dos dados de entrada visando os que possuem maior probabilidade de fazer parte da forma desejada;
- Um armazenamento apenas de células acumuladoras com votos suficientes na memória através do uso de um dicionário, descartando a necessidade da representação do espaço de parâmetros completo;
- Uma triangulação do espaço de parâmetros que define uma variedade LPP homeomorfa ao espaço de parâmetros original.

Além desses resultados originais, as seguintes afirmações sobre a abordagem proposta serão demonstradas:

- Reduz significativamente a memória e o tempo de processamento;
- Detecta retas em imagens com mais de 4 bilhões de pixels em menos de 3 segundos;
- Possui um algoritmo relativamente simples e fácil de implementar;
- Pode ser facilmente adaptável a qualquer forma parametrizada.

1.5 Organização do trabalho

Esta dissertação está organizada da seguinte maneira. No Capítulo 2, apresentamos toda a base teórica que usaremos ao longo deste trabalho. Abordaremos os con-

ceitos necessários sobre processamento de imagens, tais como derivadas da imagem e detecção de bordas; discorreremos sobre a HT e a HT Orientada; e introduziremos conceitos elementares da topologia algébrica, como espaço topológico, complexo simplicial e triangulação.

Depois, o Capítulo 3 possui todo o detalhamento da nossa abordagem para detecção de linhas retas. A abordagem está estruturada em três etapas bem definidas, com algorítimos e ilustrações que facilitam o entendimento do processo. No fim do capítulo, há uma seção demonstrando como o método pode ser adaptado para qualquer forma parametrizada.

Em seguida, o Capítulo 4, apresenta a metodologia utilizada e todos os testes realizados com a implementação desenvolvida da técnica. Define os limiares padrões para a técnica, corrobora alguns caminhos tomados para a construção da técnica e compara com outras abordagens já existentes na literatura,

O Capítulo 5 fecha a dissertação, centralizando os principais resultados apresentados no capítulo anterior, as limitações da técnicas e os trabalhos que virão a ser realizados futuramente.

2 Fundamentos Teóricos

Este capítulo apresenta os conceitos teóricos necessários para o entendimento da técnica proposta no Capítulo 3. Tais conceitos estão divididos em três partes: na Seção 2.1, apresentaremos conceitos sobre processamento digital de imagens, como derivadas da imagem e detecção de bordas; na Seção 2.2 discorreremos sobre a HT, com enfoque na HT para linhas retas e na HT Orientada; e na Seção 2.3 introduziremos conceitos elementares da topologia algébrica, como espaço topológico, complexo simplicial e triangulação.

2.1 Processamento digital de imagens

O processamento digital de imagens é uma área da computação que envolve processos cujas entradas e saídas são imagens, como realce e contraste, além de processos de extração de atributos de imagem, como segmentação e detecção, e o reconhecimento de objetos individuais (GONZALEZ; WOODS, 2010). Suas aplicações são amplamente utilizadas na medicina, ciência, segurança, entretenimento, automação industrial, e muitas outras.

Nesta seção, exploraremos conceitos básicos do processamento digital de imagens, discorreremos sobre as derivadas da imagem, com foco no gradiente e na curvatura, e abordaremos a detecção de bordas utilizando a Figura 4 como exemplo.

2.1.1 Conceitos básicos

As imagens digitais são representações numéricas de dados visuais, capturadas por meio de dispositivos de aquisição, como câmeras digitais, scanners ou sensores. Essas imagens são compostas por uma matriz bidimensional de pixels, que contêm informações de cor e intensidade daquele ponto na imagem.

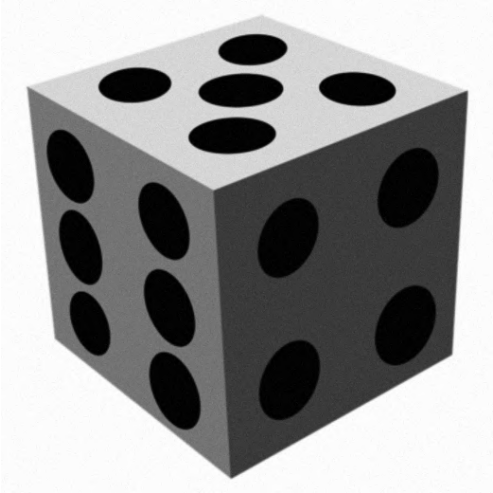


Figura 4: Imagem I: dado em tons de cinza com ruído.

Existem diferentes tipos de imagens, como imagens em tons de cinza, imagens coloridas e imagens binárias. Em uma imagem em tons de cinza, por exemplo, cada pixel possui um valor que varia de 0 (preto) a 255 (branco), representando a intensidade $I(x, y)$ do pixel na posição (x, y) da imagem. Desta forma, podemos representar a imagem como uma função $f(x, y)$, onde $f(x, y) = I(x, y)$. O tamanho da imagem é definido por M linhas e N colunas, formando uma matriz $M \times N$.

Os operadores mais básicos de processamento de imagens são aqueles que manipulam cada pixel independentemente de seus vizinhos, por isso são chamados de **operadores pontuais** ou **processos pontuais**. Esses operadores são capazes de realizar ajustes de brilho e contraste, além de corrigir e transformar de cores, por exemplo.

Também existem os **operadores de vizinhança**, onde o valor de cada novo pixel depende de um pequeno número de valores de entrada vizinhos. Esses operadores são subdivididos entre os **lineares**, que envolvem combinações ponderadas dos pixels da vizinhança, e os **não lineares**, como filtros morfológicos e transformações de distância. Além de realizar o ajuste de tom local, esses operadores podem ser usados para filtrar imagens, adicionar desfoque suave, aguçar detalhes, acentuar bordas ou remover ruídos.

Outra classe importante de operadores são as **transformações geométricas**, como rotações, cisalhamentos e deformações em perspectiva. Essas transformações, ao invés de alterar a intensidade de cada pixel como ocorre nos operadores pontuais,

alteram a sua posição (x, y), ou seja, são utilizados para manipular o domínio da imagem (SZELISKI, 2010).

2.1.1.1 Operadores de vizinhança lineares

Nos operadores de vizinhança lineares, o valor de um pixel de saída é determinado como uma somatória ponderada dos valores de pixel de entrada. Os mais utilizados entre eles são os operadores de **correlação** e **convolução**.

O operador de correlação é definido por

$$g(x, y) = \sum_{i,j} f(x + i, y + j)h(i, j) = \sum_{i,j} f(i, j)h(x + i, y + j), \tag{2.1}$$

onde $h(i, j)$ é conhecido como *máscara* ou *kernel*. Pode ser escrito de forma mais compacta como

$$g = f \otimes h. \tag{2.2}$$

O operador de convolução pode ser obtido através do operador de correlação invertendo o sinal dos deslocamentos,

$$g(x, y) = \sum_{i,j} f(x - i, y - j)h(i, j) = \sum_{i,j} f(i, j)h(x - i, y - j), \tag{2.3}$$

e sua versão compacta é dada por

$$g = f * h. \tag{2.4}$$

Ambos operadores são invariantes por deslocamento linear, obedecendo ao princípio da superposição,

$$(f_0 + f_1) \circ h = f_0 \circ h + f_1 \circ h, \tag{2.5}$$

e ao princípio da invariância por deslocamento,

$$g(x, y) = f(x + i, y + j) \quad \Leftrightarrow \quad (g \circ h)(x, y) = (f \circ h)(x + i, y + j), \tag{2.6}$$

ou seja, se comportam da mesma forma em qualquer lugar.

2.1.2 Derivadas da imagem

No campo da matemática, a derivada é uma medida da taxa de variação de uma função em relação a uma variável independente. Ela nos fornece informações sobre como a função se comporta à medida que a variável independente se altera. A derivada é frequentemente usada para descrever o comportamento das funções, identificar pontos críticos, determinar a taxa de crescimento ou decrescimento, entre outras aplicações.

No contexto do processamento de imagens, as derivadas são usadas para analisar a taxa de variação das intensidades dos pixels em relação às coordenadas espaciais da imagem. Elas são ferramentas poderosas para detectar bordas, contornos, texturas e outras características importantes presentes nas imagens digitais.

As derivadas de primeira ordem no processamento de imagens medem a taxa de variação da intensidade dos pixels em relação às coordenadas espaciais. Elas permitem calcular a magnitude e a direção do gradiente de cada pixel, por exemplo, possibilitando a detecção de bordas e o realce de características nas imagens.

Existem duas derivadas de primeira ordem principais: a derivada parcial em relação a x, que mede a taxa de variação da intensidade na direção horizontal, e a derivada parcial em relação a y, que mede a taxa de variação na direção vertical.

Já as derivadas de segunda ordem medem a taxa de variação da taxa de variação da intensidade dos pixels em relação às coordenadas espaciais. Elas fornecem informações sobre a curvatura das intensidades e são úteis para detectar mudanças mais sutis nas imagens.

Uma forma muito comum de definir a derivada é através do **método das diferenças finitas**. Neste método, as derivadas da equação diferencial parcial são aproximadas por combinações lineares de valores de função nos pontos da grade de pixels. As aproximações de primeira ordem para as derivadas parciais podem ser obtidas através de uma expansão em série de Taylor da seguinte forma (GROSSMANN; ROOS; STYNES, 2007):

$$f_x(x,y) \approx \frac{f(x+i,y) - f(x-i,y)}{2i} \tag{2.7}$$

$$f_y(x,y) \approx \frac{f(x,y+j) - f(x,y-j)}{2j} \tag{2.8}$$

$$f_{xx}(x,y) \approx \frac{f(x+i,y) - 2f(x,y) + f(x-i,y)}{i^2} \tag{2.9}$$

$$f_{yy}(x,y) \approx \frac{f(x,y+j) - 2f(x,y) + f(x,y-j)}{j^2} \tag{2.10}$$

$$f_{xy}(x,y) \approx \frac{f(x+i,y+j) - f(x+i,y-j) - f(x-i,y+j) + f(x-i,y-j)}{4ij} \tag{2.11}$$

Estas aproximações são utilizadas para calcular operadores de convolução, como veremos a seguir.

2.1.2.1 Gradiente

O gradiente é uma medida vetorial da taxa de variação de uma função escalar em relação às suas coordenadas espaciais. No processamento de imagens, o gradiente é usado para identificar transições bruscas de intensidade que geralmente correspondem a bordas ou contornos presentes na imagem.

Para calcular o gradiente, aplica-se operadores de convolução de diferenciação discreta à imagem, como o operador de Sobel (SOBEL, 1968). Ele consiste em dois kernels separados, um para a direção horizontal (S_x), obtido através do produto externo entre um filtro Gaussiano e a derivada em relação a x (Equação 2.7):

$$S_x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \tag{2.12}$$

e outro para a direção vertical (S_y), obtido através do produto externo entre a derivada em relação a y (Equação 2.8) e um filtro Gaussiano:

$$S_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \tag{2.13}$$

Esses kernels são aplicados à imagem para estimar as derivadas parciais em relação às direções x e y.

A convolução desses kernels com a imagem I resulta em duas imagens intermediárias, G_x e G_y, que representam as derivadas em relação às direções x e y, respectivamente, i.e.:

$$G_x = S_x * I \qquad G_y = S_y * I \tag{2.14}$$

Tais imagens podem ser representadas pela Figura 5, onde os valores de intensidades foram remapeados para valores de 0 a 255, sendo 0 (preto) a representação do valor mínimo e 255 (branco) a representação o valor máximo.

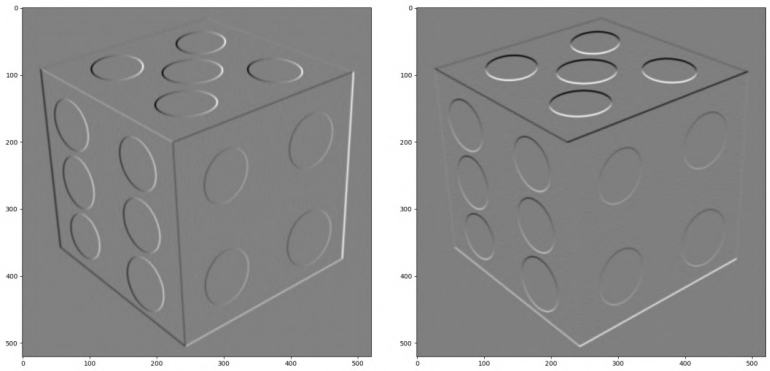


Figura 5: Imagens intermediárias G_x (esquerda) e G_y (direita).

O gradiente da imagem pode ser calculado a partir dessas derivadas parciais da seguinte maneira:

$$G = \sqrt{G_x^2 + G_y^2} \tag{2.15}$$

$$\Theta = \arctan(G_y/G_x) \tag{2.16}$$

onde G é a magnitude do gradiente, representando a variação de intensidade naquele ponto, e Θ é a direção do gradiente, indicando a orientação da maior variação.

No lado esquerdo da Figura 6, é possível ver que, quanto maior a intensidade (mais claro), maior o gradiente da imagem; no lado direito, as variações de tonalidade representam variações angulares de -90 (preto) a 90 graus (branco) em relação á direção x.

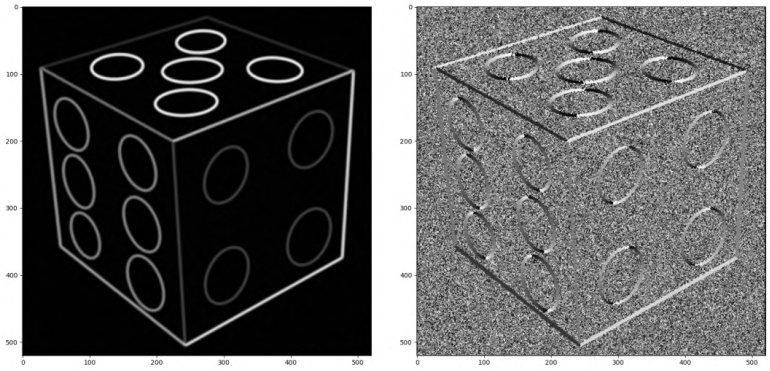


Figura 6: Magnitude G (esquerda) e direção Θ (direita) do gradiente.

2.1.2.2 Curvatura

Para curvas planas, a curvatura pode ser definida como uma medida da taxa de variação do vetor gradiente em relação às coordenadas x e y. Para calcular a curvatura, é necessário também usar as segundas derivadas parciais da imagem que podem ser obtidas aplicando os operadores construídos a partir das derivadas de segunda ordem (Equações 2.9, 2.10 e 2.11) à imagem:

$$G_{xx} = K_{xx} * I \qquad G_{yy} = K_{yy} * I \qquad G_{xy} = K_{xy} * I \tag{2.17}$$

onde K_{xx}, K_{yy} e K_{xy} podem ser definidos como

$$K_{xx} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.18}$$

$$K_{yy} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{bmatrix} \tag{2.19}$$

$$K_{xy} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \tag{2.20}$$

Outra forma de obter as segundas derivadas parciais da imagem é simplesmente aplicando os operadores de Sobel S_x e S_y nas as imagens intermediárias G_x e G_y

(Figura 7),

$$G_{xx} = S_x * G_x \qquad G_{yy} = S_y * G_y \qquad G_{xy} = S_x * G_y \tag{2.21}$$

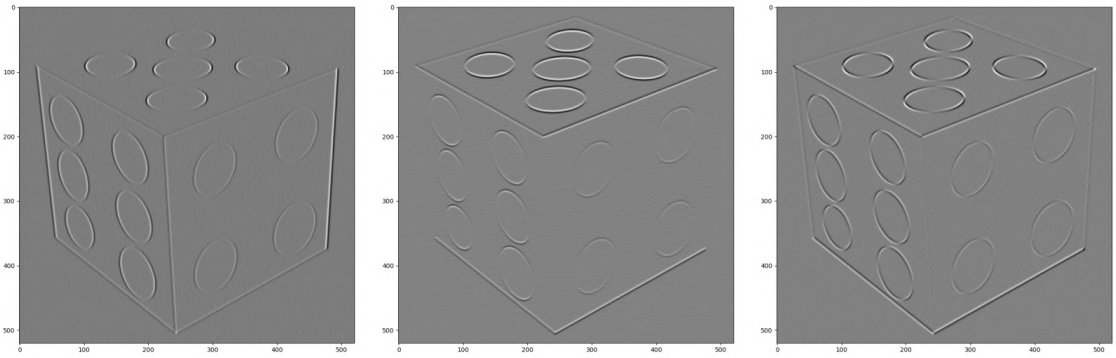


Figura 7: Imagens intermediárias G_{xx} (esquerda), G_{yy} (centro) e G_{xy} (direita).

Assim, Goldman (2005) escreve a fórmula da curvatura como sendo

$$\kappa = -\frac{\begin{pmatrix} -G_y & G_x \end{pmatrix} * \begin{pmatrix} G_{xx} & G_{xy} \\ G_{xy} & G_{yy} \end{pmatrix} * \begin{pmatrix} -G_y \\ G_x \end{pmatrix}}{(G_x^2 + G_y^2)^{3/2}}. \tag{2.22}$$

Uma curvatura de pequena magnitude em um ponto de bordo da imagem pode indicar que este ponto faz parte de um segmento de reta ou uma curva suave enquanto uma curvatura de grande magnitude pode indicar curvas acentuadas ou encontro de segmentos (quinas) (Figura 8).

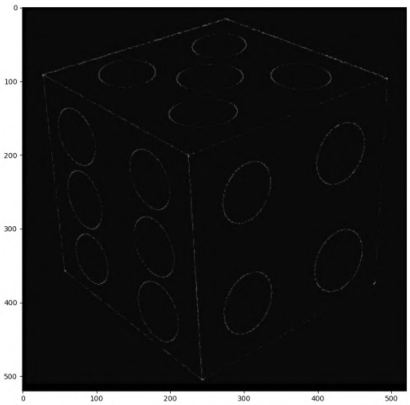


Figura 8: Magnitude da curvatura k.

2.1.3 Detecção de bordas

A detecção de bordas é uma etapa altamente difundida na análise de imagens e nas tarefas de visão computacional por conta do seu papel crucial na identificação de características importantes dentro da imagem, como limite de objetos, limites de sombras e bordas acentuadas. Tais características são utilizadas no reconhecimento de objetos, segmentação de imagens e análise de cenas, por exemplo.

Segmentar uma imagem em regiões coerentes é uma tarefa difícil, por isso é preferível detectar bordas usando apenas informações puramente locais. Assim, uma abordagem razoável é definir uma borda como um local de rápida variação de intensidade e, como vimos na Seção 2.1.2.1, o gradiente é uma forma matemática identificar tais variações.

Como o gradiente é muito suscetível a ruído na imagem, uma vez que a proporção de ruído em relação ao sinal é maior em altas frequências, o primeiro passo é suavizar a imagem com um filtro passa-baixa antes que o cálculo do gradiente seja feito. O filtro utilizado na maioria dos algoritmos é o Gaussiano por ser circularmente simétrico, o que garante que a detecção de bordas independa da orientação da imagem (Figura 9).

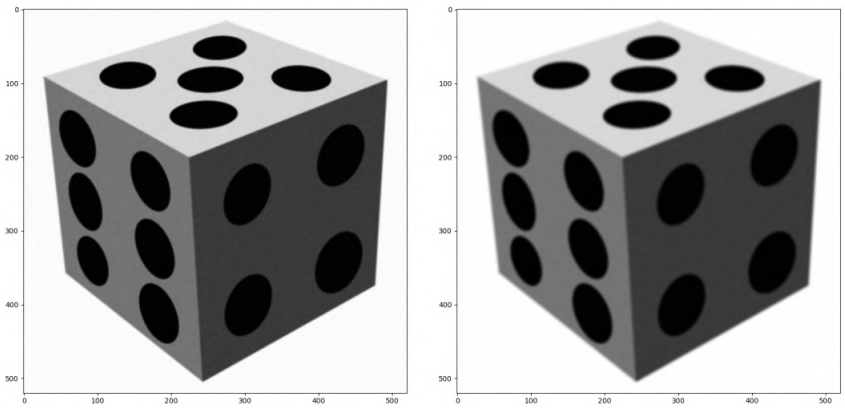


Figura 9: Imagem original (esquerda) e imagem suavizada com o filtro Gaussiano ($\sigma = 3$ e kernel 5×5) (direita).

Em seguida, aplica-se operadores de convolução de diferenciação discreta à imagem suavizada para obter a primeira derivada na direção horizontal (G_x) e na direção vertical (G_y) e encontrar o gradiente através das Equações 2.15 e 2.16 (Figura 10).

Em muitos casos é desejável diluir essa imagem de gradiente contínuo para re-

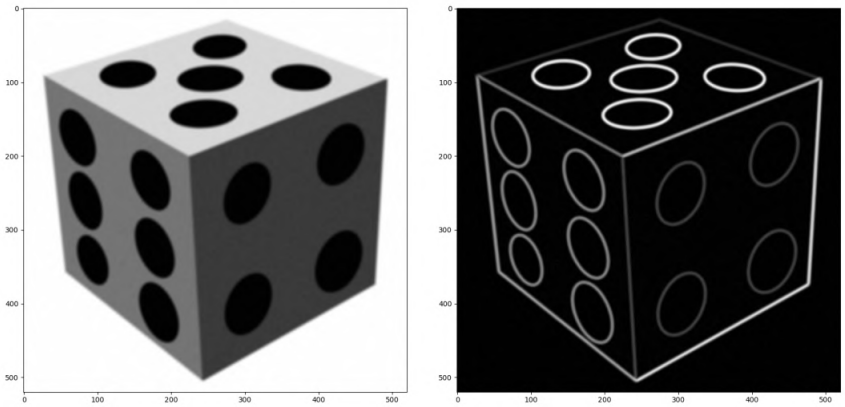


Figura 10: Imagem suavizada (esquerda) e magnitude do gradiente (direita).

tornar apenas arestas isoladas, isto é, remover quaisquer pixels indesejados que possam não constituir a borda. Isto pode ser feito procurando máximos na resistência da borda (magnitude do gradiente) em uma direção perpendicular à orientação da borda, ou seja, ao longo da direção do gradiente. O resultado obtido é uma imagem binária com "bordas finas" (Figura 11).

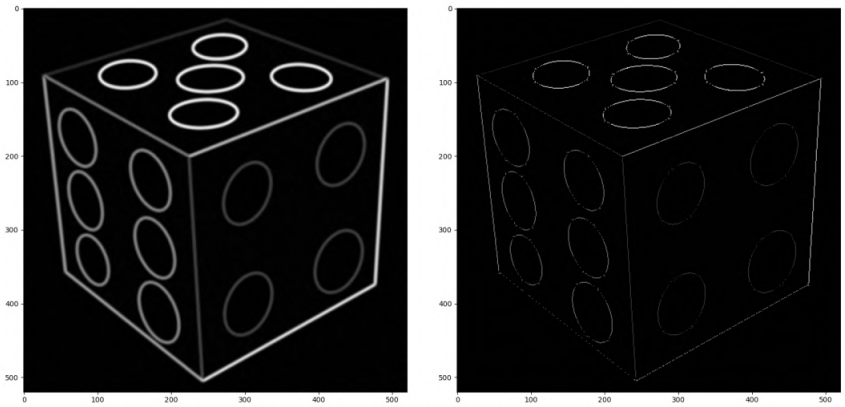


Figura 11: Magnitude do gradiente (esquerda) e resultado da supressão dos não-máximos (direita).

Por fim, alguns métodos utilizam valores limites para filtrar as bordas remanescentes. O método Canny (CANNY, 1986), por exemplo, utiliza dois valores limites, val_{min} e val_{max}. Quaisquer bordas com gradiente de intensidades maior que o val_{max} certamente serão bordas e as abaixo do val_{min} certamente não serão bordas, portanto descartadas. Aquelas que se encontram entre esses dois limites são classificadas como bordas ou não com base em sua conectividade. Se estiverem conectadas a pixels com gradiente de intensidades maior que o max_{val}, elas serão consideradas parte das bordas, caso contrário, elas também serão descartadas. Esta

etapa também remove pequenos ruídos que não foram eliminados durante a etapa de suavização (Figura 12).

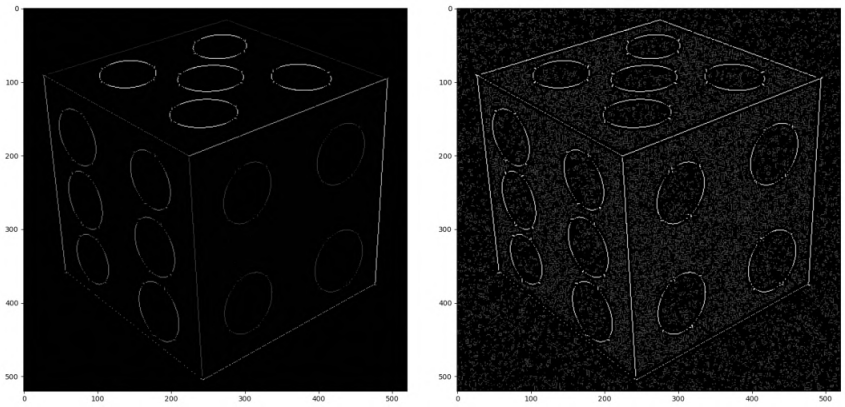


Figura 12: Supressão dos não-máximos (esquerda) e resultado dos valores limites ($val_{max} = 20$ e $val_{min} = 4$): pixels descartados em cinza e remanescentes em branco (direita).

Desta forma, os pixels não descartados nas etapas apresentadas são considerados pixels de borda da imagem (Figura 13).

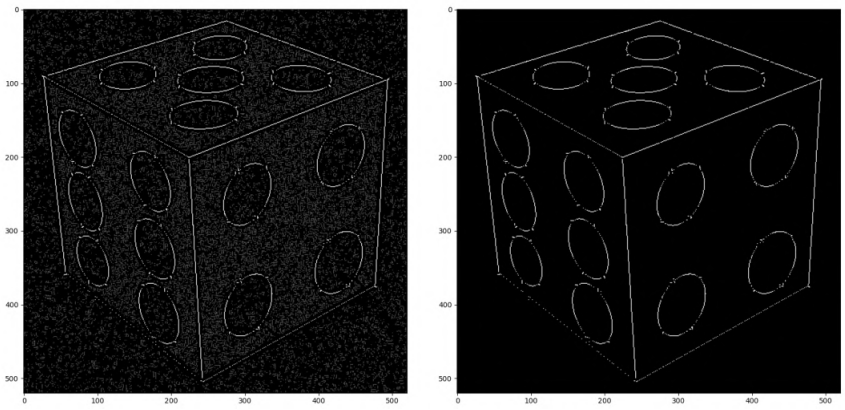


Figura 13: Resultado final do método de detecção de bordas.

2.2 Transformada de Hough

Uma das maneiras mais conhecidas de detectar alinhamentos de dados, formas geométricas ou padrões que melhor se ajustam a determinados conjuntos de dados não ordenados é a HT.

Nomeada em homenagem ao seu inventor original, a HT foi inicialmente proposta para detectar retas em imagens binárias (HOUGH, 1962; DUDA; HART,

1972). Posteriormente, a HT foi aprimorada para detectar outros tipos de formas facilmente parametrizadas, como círculos (DUDA; HART, 1972), parábolas (SKLANSKY, 1978), elipses (BENNETT; BURRIDGE; SAITO, 1999), formas não analíticas em imagens (BALLARD, 1981) e generalizadas para detecção de quaisquer formas analíticas a partir da combinação de qualquer tipo analítico de entrada que possa ser codificado por um subespaço linear em um modelo de geometria linear ou não linear (FERNANDES; OLIVEIRA, 2014).

De modo geral, uma HT pode ser definida por qualquer forma que possa ser representada pela **função de modelo**:

$$f(X, \boldsymbol{p}) = 0, \tag{2.23}$$

onde $X = x_1, x_2, \cdots, x_d$ é um dado do conjunto de entrada, normalmente um ponto em $\mathbb{R}^d$, e $\boldsymbol{p} = p_1, p_2, \cdots, p_m$ é um vetor do espaço de parâmetros $\mathbb{P}^m$ que representa uma instância da forma pretendida.

O espaço de parâmetros na HT é uma representação abstrata onde as características geométricas da forma a ser detectada são expressas em termos de parâmetros. Para cada forma geométrica, o espaço de parâmetros pode ter diferentes valores de dimensão m e os parâmetros podem representar diferentes propriedades da forma. A sua discretização define um conjunto de células acumuladoras, onde será realizado o processo de votação.

Para realizar a votação, é definido um subconjunto de k valores de parâmetros de $\boldsymbol{p}$ (para $k < m$) arbitrariamente e os valores de parâmetros não arbitrados $(m-k)$ são calculados usando a **função de mapeamento**:

$$(p_{k+1}, p_{k+2}, \cdots, p_m) = g(X; p_1, p_2, \cdots, p_k). \tag{2.24}$$

A Equação 2.24 é avaliada para todo $p_1, p_2, \cdots, p_k \in \mathbb{P}^k$ e a escolha de quais parâmetros são arbitrados e quais são calculados depende de como as Equações 2.23 e 2.24 são definidas.

De forma análoga à HT original, cada entrada X vota para todas as células acumuladoras possíveis da forma que passam por ela e as células que recebem muitos votos correspondem aos parâmetros das formas mais prováveis no conjunto de entrada.

A seguir, veremos com mais detalhes como é feito o processo de votação para a detecção de linhas retas.

2.2.1 Transformada de Hough para linhas retas

Na detecção de linhas retas, a função de modelo pode ser dada, por exemplo, pela equação **reduzida** da reta:

$$ax + b - y = 0, \tag{2.25}$$

onde $X = x, y$ (coordenadas do ponto) e $\boldsymbol{p} = a, b$ (coeficientes angular e linear, respectivamente). Definindo a como parâmetro arbitrado, a função de mapeamento é dada por

$$b = y - ax, \tag{2.26}$$

porém, estes valores de parâmetros possuem a desvantagem de não possuírem um valor limite (a,$b \in (-\infty, +\infty)$), e não parametrizarem retas verticais, onde $a = \pm\infty$.

Outra candidata à função de modelo é a equação **normal** da reta:

$$x\cos\theta + y\sin\theta - \rho = 0, \tag{2.27}$$

onde $X = x, y$ e $\boldsymbol{p} = \rho, \theta$, sendo ρ a menor distância da reta a origem do sistema de coordenadas da imagem e $\theta \in (-\pi/2, \pi/2]$ o ângulo entre o eixo x e a normal à linha, de forma que a reta seja definida por um único valor de θ. Definindo θ como parâmetro arbitrado, a função de mapeamento é dada por

$$\rho = x\cos\theta + y\sin\theta. \tag{2.28}$$

Os parâmetros das Equações 2.27 e e 2.28 são mais adequados por possuírem um valor limite[1] e parametrizarem todas as retas.

Com a escolha dos parâmetros, os valores de θ são definidos de acordo com a discretização do espaço de parâmetros. É interessante notar que o espaço de parâmetros $\rho\theta$ consiste em uma projeção bidimensional de um hemisfério, ou seja, um mapeamento de um hemisfério sobre um plano Cartesiano. De maneira análoga às distorções observadas na representação da superfície terrestre em mapas, este espaço

[1] O parâmetro ρ se torna limitado no contexto de detecção de formas em imagens pois a imagem em si é limitada, logo, se a origem estiver em um dos cantos de uma imagem de dimensões $M \times N$, o valor máximo de $|\rho|$ será $(M^2 + N^2)^{1/2}$.

também apresenta distorções, uma vez que o mapeamento não preserva distâncias geodésicas, resultando em deformações nas formas e tamanhos das áreas.

Para cada θ_i possível, é calculado ρ_i através da função de mapeamento, de forma que todas as retas que passam por X depositem 1 (um) voto na célula acumuladora correspondente. Uma representação deste processo pode ser observada na Figura 14, onde é possível identificar a formação da senoide.

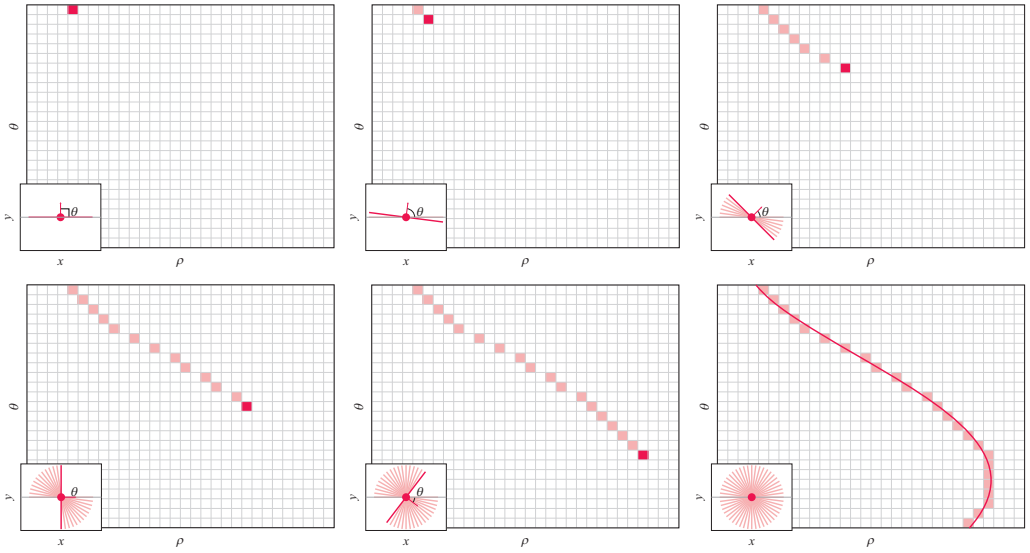


Figura 14: Processo de votação de um ponto da imagem.

O processo se repete para todos os pontos de entrada e, quando todos os votos são depositados, o processo de votação se encerra.

A partir então, inicia-se a busca por máximos locais, pois suas localizações indicam prováveis retas na imagem. Por conta da discretização do espaço de parâmetros e da imagem, que faz com que os pixels pertencentes a uma reta não sejam perfeitamente colineares (Figura 15), muitos máximos locais acabam correspondendo a mesma reta da imagem, sendo necessário um processo de agrupamento ou cancelamento a fim de encontrar as retas que mais se adéquam ao conjunto de pontos de entrada.

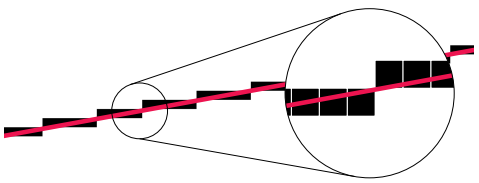


Figura 15: Pixels de borda correspondentes à reta vermelha.

A Figura 16 ilustra as etapas da HT para a detecção da reta que melhor se ajusta a três pontos de entrada.

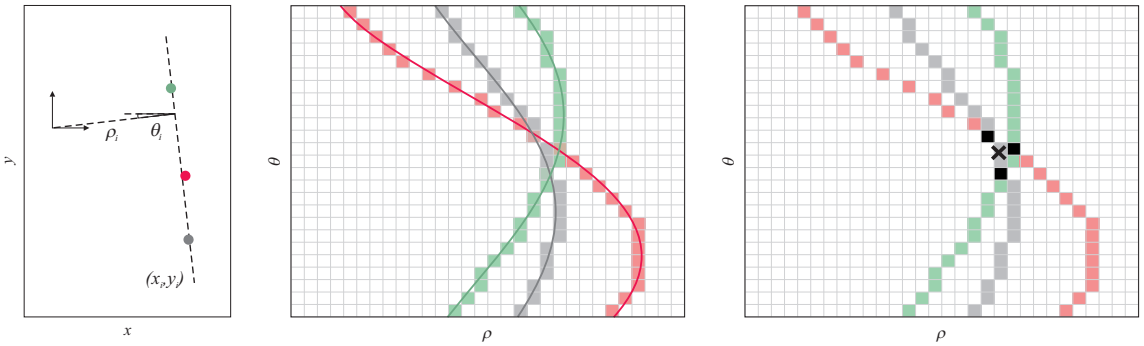


Figura 16: Etapas da HT para detecção de retas. Cada ponto da imagem (esquerda) vota para todas as linhas possíveis $\rho = x\cos\theta + y\sin\theta$, formando senoides no espaço de parâmetros (centro), e o ponto médio dos máximos locais fornece a equação de reta desejada (direita), representada pela linha tracejada.

2.2.2 Transformada de Hough orientada

A Transformada de Hough orientada (OHT, do inglês *oriented Hough Transform*) utiliza um conjunto de m entradas para definir uma forma arbitrária e votar em uma única célula acumuladora no espaço de parâmetros (SZELISKI, 2010). Dessa forma, os k valores de parâmetros deixam de ser definidos arbitrariamente e passam a ser calculados pelos valores das m entradas selecionadas através da **segunda função de mapeamento**:

$$(p_1, p_2, \cdots, p_k) = h(X_1, X_2, \cdots, X_m). \tag{2.29}$$

Com isso, a Equação 2.24 pode ser aplicada utilizando qualquer uma das m entradas selecionadas.

Por exemplo, como o valor de θ corresponde ao ângulo do gradiente da imagem, podemos escrever a Equação 2.29 da seguinte maneira para determinar o valor do parâmetro θ da Equação 2.27:

$$\theta = \arctan\left(\frac{f_y}{f_x}\right), \tag{2.30}$$

onde f_y e f_x são as derivadas parciais de primeira ordem nas direções y e x, respectivamente.

Assim sendo, cada ponto de coordenadas X deposita um único voto apenas na célula acumuladora que representa a reta que passa por X cuja normal corresponde ao ângulo do gradiente naquele ponto. Por fim, as células com mais votos indicam prováveis retas na imagem, sendo necessário um processo de agrupamento ou cancelamento.

A Figura 17 ilustra as etapas da OHT para a detecção da reta que melhor se ajusta a três pontos de entrada.

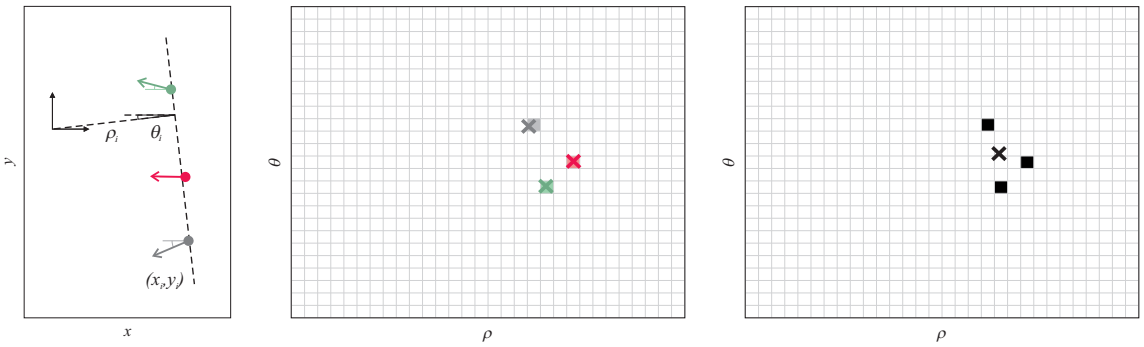


Figura 17: Etapas da OHT para detecção de retas. Cada ponto da imagem (esquerda) vota em apenas uma célula acumuladora (centro), e o ponto médio das células votadas fornece a equação de reta desejada (direita), representada pela linha tracejada.

Há ainda um método híbrido entre a HT e a OHT, que utiliza um conjunto de m entradas para definir uma célula acumuladora como ponto central de uma porção de votos (O'GORMAN; CLOWES, 1976). No caso da detecção de retas, o gradiente θ é utilizado como ponto central da porção de retas votantes. O tamanho desta porção pode ser definido como parâmetros (+/- 10 graus, por exemplo) ou calculado considerando as incertezas na discretização da imagem e na estimativa do gradiente. Como na HT, os máximos locais indicam prováveis retas na imagem, sendo necessário também um processo de agrupamento ou cancelamento.

A Figura 18 ilustra as etapas do método híbrido para a detecção da reta que melhor se ajusta a três pontos de entrada.

Novamente, a escolha de quais parâmetros são calculados pelos valores de entrada e quais são calculados a partir de outros parâmetros depende de como as Equações 2.23 e 2.24 são definidas.

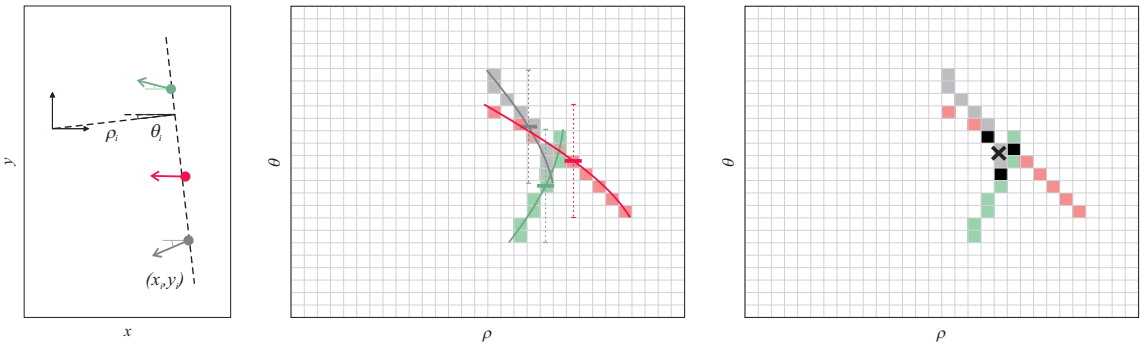


Figura 18: Etapas do método híbrido para detecção de retas. Cada ponto da imagem (esquerda) vota um conjunto de células acumuladoras, centrado em θ e de tamanho $2\delta\theta$ (centro), e o ponto médio dos máximos locais fornece a equação de reta desejada (direita), representada pela linha tracejada.

2.3 Topologia algébrica

A topologia algébrica é o ramo da matemática que combina conceitos da topologia e álgebra para estudar propriedades geométricas e topológicas dos espaços. É responsável por associar estruturas algébricas a espaços topológicos de forma a identificar propriedades que permanecem invariantes sob deformações contínuas, transformações e mapeamentos entre espaços. Isso permite a criação de invariantes topológicos que são robustos em relação a deformações contínuas, preservando características essenciais dos espaços estudados.

Nesta seção, introduziremos conceitos elementares da topologia algébrica que serão utilizados na nova abordagem da HT descrita no Capítulo 3, como espaço topológico, complexo simplicial e triangulação (MATOUSEK, 2003), além de apresentar uma forma de encontrar pontos críticos geométricos.

2.3.1 Espaço topológico

O espaço topológico é uma estrutura matemática que formaliza a noção intuitiva de **proximidade** e **continuidade** em espaços abstratos. Sua definição é baseada em um conjunto de axiomas que especificam quais subconjuntos desse espaço são considerados abertos, refletindo assim as propriedades topológicas do conjunto.

Matematicamente, um **espaço topológico** é um par $(X, \mathcal{T})$, onde X é um conjunto fundamental (tipicamente infinito) e $\mathcal{T} \subseteq 2^X$ é um sistema de conjuntos, cujos membros são chamados de **conjuntos abertos**, tais que:

1. $\varnothing \in \mathcal{T}$e $X \in \mathcal{T}$, ou seja, o conjunto vazio e o conjunto completo são abertos.

2. Se $U_1, U_2, \ldots, U_n \in \mathcal{T}$ são abertos, então a sua interseção $\bigcap_{i=1}^{n} U_i$ também é um conjunto aberto.

3. Se $\{U_i\}$ é uma coleção arbitrária de conjuntos abertos, então a sua união $\bigcup_i U_i$ também é um conjunto aberto.

Um conjunto U em um espaço topológico X é **fechado** se o complemento de U em X, denotado por $X \backslash U$, for aberto. O **fechamento** de $U \subseteq X$, denotado por $cl(U)$, é a interseção de todos os conjuntos fechados em X contendo U.

A **fronteira** de U é dado por

$$U^{\circ} = \{cl(U) \cap cl(X \backslash U)\} \tag{2.31}$$

e o **interior** de U é definido como

$$U^{\bullet} = U \backslash U^{\circ}. \tag{2.32}$$

É possível que dois espaços $(X_1, \mathcal{T}_1)$ e $(X_2, \mathcal{T}_2)$ que aparentam ter diferenças significativas em termos de métrica, sejam topologicamente iguais como, por exemplo, um torus e uma caneca (Figura 19). A noção de "ser igual" para espaços topológicos é tradicionalmente chamada de **homeomorfismo**, que é uma bijeção $f : X_1 \to X_2$ tal que para todo $U \subseteq X_1$, $f(U) \in \mathcal{T}_2$ se e somente se $U \in \mathcal{T}_1$. Em termos simples, dois espaços são homeomorfos se puderem ser transformados um no outro de uma maneira contínua e sem rasgaduras.

2.3.2 Complexo simplicial

Os complexos simpliciais são estruturas que fornecem uma maneira abstrata de representar e estudar a topologia de espaços topológicos mais complexos por meio de elementos geométricos simples chamados simplexos, que incluem pontos, segmentos de retas, triângulos e suas generalizações para dimensões superiores, respeitando regras simples.

Podemos definir um **simplexo** σ como o fecho convexo de um conjunto de pontos A em $\mathbb{R}^d$. Os pontos de A pertencentes ao fecho convexo de σ são chamados

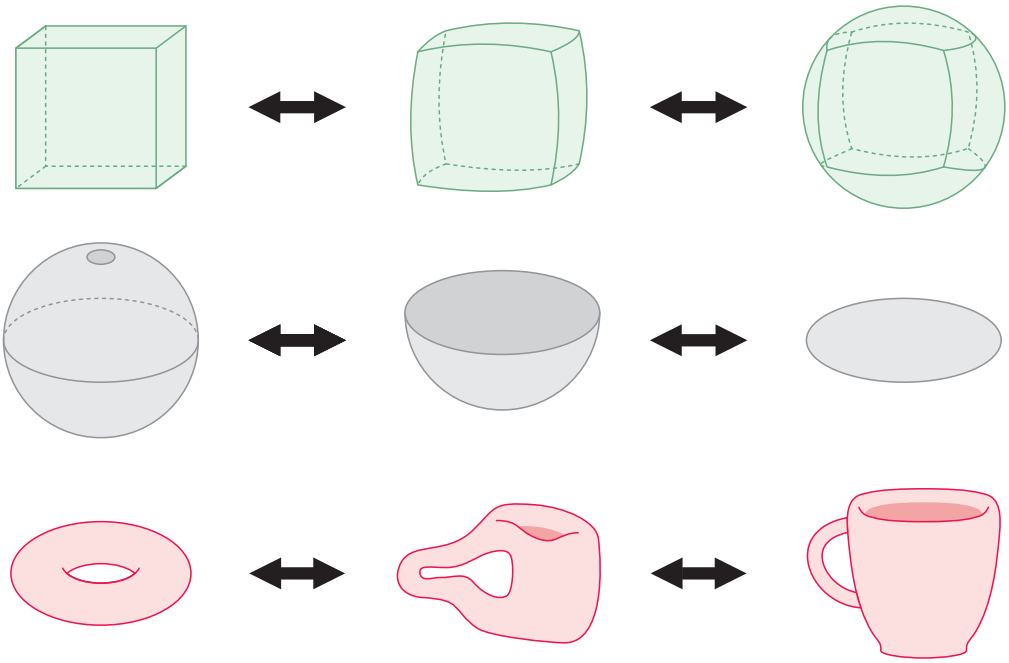


Figura 19: Homeomorfismo entre cubo e esfera (verde); esfera furada, hemisfério, e disco aberto (cinza); e torus e caneca (vermelho).

de vértices de σ, denotados por v, e a dimensão de σ é dada por

$$k = |V| - 1, \tag{2.33}$$

onde V é o conjunto de vértices $v \in A$ e $|V|$ é a cardinalidade do conjunto.

Assim, todo k-simplexo (ou simplexo k-dimensional) possui $k+1$ vértices. Por exemplo, um simplexo 0-dimensional é um ponto, um 1-dimensional é um segmento de linha, um 2-dimensional é um triângulo, e assim em diante (Figura 20).

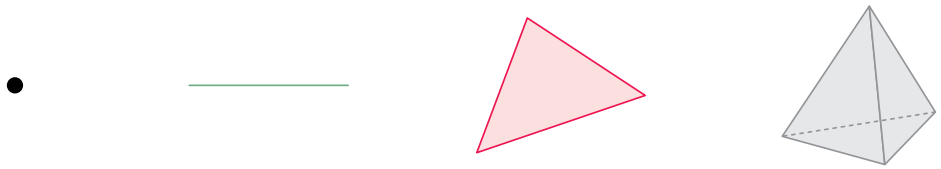


Figura 20: Simplexos de dimensões 0 (ponto), 1 (segmento de linha), 2 (triângulo) e 3 (tetraedro), respectivamente.

Além disso, os vértices $v_0, v_1, ..., v_k$ de σ precisam ser **afins independentes**, ou seja, $\sum_{i=0}^{k} \alpha_i v_i = 0$ somente se $\alpha_1 = \alpha_2 = \ldots = \alpha_k = 0$, onde α_i são números reais. Por exemplo, afim independência de dois pontos significa serem diferentes, de três pontos significa não pertencerem a uma linha comum, de quarto pontos significa não

pertencerem a um plano comum, e assim em diante.

Desta forma, temos que:

- Os k vetores $v_1 - v_0, v_2 - v_0, \ldots, v_k - v_0$ são linearmente independentes.
- Os vetores de dimensão d+1 $(1, v_0), (1, v_1), \ldots, (1, v_k) \in \mathbb{R}^{d+1}$ são linearmente independentes.

O fecho convexo de um subconjunto arbitrário de vértices de σ é uma **face** de σ e, consequentemente, cada face é em si um simplex. O que sobra quando removemos todas as faces de dimensão menor que k é chamado de **interior relativo** de σ. O simplexo em si é uma união disjunta dos interiores relativos de suas faces.

Na Figura 21, podemos observar que o triângulo fechado é a união de um triângulo aberto (seu interior relativo) τ, 3 arestas $\epsilon_0, \epsilon_1, \epsilon_2$, 3 vértices v_0, v_1, v_2 e o conjunto vazio.

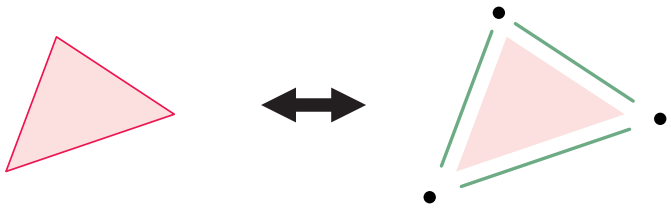


Figura 21: Triângulo fechado e seus componentes.

Uma família não vazia Σ de simplexos é o que chamamos de **complexo simplicial** somente se:

1. Cada face de qualquer simplexo $\sigma \in \Sigma$ também é um simplexo de Σ.
2. A interseção de quaisquer dois simplexos $\sigma_0, \sigma_1 \in \Sigma$ é uma face tanto de σ_0 quanto de σ_1.

A dimensão de Σ é a maior dimensão de um simplexo:

$$\dim \Sigma = \max\{k_\sigma : \sigma \in \Sigma\}. \tag{2.34}$$

Dado um vértice $v \in \Sigma$, o conjunto de todos os simplexos de Σ que possuem v como vértice (incluindo simplexo v em si) é chamado de **estrela** de v (Figura 22):

$$St(v) = \{\sigma \in \Sigma \mid v \subseteq \sigma\}. \tag{2.35}$$

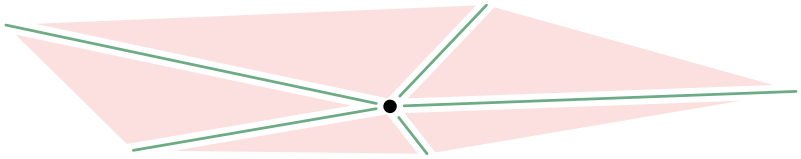


Figura 22: Estrela do vértice.

Se um vértice v_1 está conectado a v_0 por uma aresta $\epsilon \in St(v_0)$, dizemos que v_1 é um **vértice adjacente** a v_0, e a **distância** entre v_1 e v_0 é 1 (lembrando que $v_1 \notin St(v_0)$, pois $v_0 \not\subseteq v_1$). Se um vértice v_2 é adjacente a v_1, mas não a v_0, a distância entre v_2 e v_1 é 2, e assim por diante.

Um complexo simplicial Σ finito e homeomorfo a um espaço topológico X é chamado de **triangulação** de X. Por exemplo, a triangulação mais simples da esfera S^{k-1} é o limite de um k-simplexo, ou seja, o subcomplexo de σ^k obtido pela exclusão do simplexo k-dimensional único (Figura 23). Consequentemente, limite de um k-simplexo é homeomórfico a S^{k-1}.

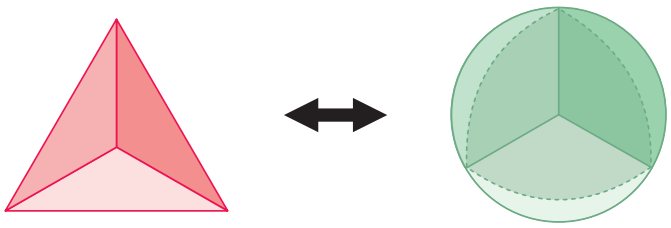


Figura 23: Homeomorfismo entre um 3-simplexo e uma esfera S^2.

2.3.3 Variedade linear por partes

Uma forma de representar superfícies e outros espaços topológicos é através de estruturas topológicas formadas por complexos simpliciais definidos pela triangulação de pontos. Tais estruturas são conhecidas como variedades lineares por partes.

Primeiramente, uma **variedade topológica** $\mathcal{V}$ d-dimensional (ou d-variedade) é um espaço topológico onde os pontos são representados por um vetor d-dimensional e a vizinhança de cada ponto é homeomorfa a $\mathbb{R}^d$. Em outras palavras, é um espaço topológico que localmente se assemelha a um espaço Euclidiano.

Superfícies bidimensionais, como esferas e planos, são exemplos de 2-variedades, enquanto uma curva unidimensional, como um círculo, é uma 1-variedade. Uma

lemniscata, por exemplo, não representa uma variedade, pois possui pontos de cruzamento.

Uma variedade é dita LPP se existe um complexo simplicial Σ tal que cada ponto admita uma vizinhança aberta U de forma que haja uma triangulação de U e um homeomorfismo LPP $f : U \to \mathbb{R}^d$. Ou seja, se uma variedade topológica $\mathcal{V}$ possui uma estrutura LPP, dizemos que $\mathcal{V}$ é uma variedade LPP.

2.3.4 Pontos críticos geométricos

Para encontrar os pontos críticos de uma função de altura h para 2-variedades LPP definidas por uma triangulação, Banchoff (1970) definiu as **estrelas inferior** e **superior** de v da seguinte forma:

$$St^+(v) = \{\sigma \in St(v) \mid h(u) > h(v), v \subseteq \sigma\}, \tag{2.36}$$

$$St^-(v) = \{\sigma \in St(v) \mid h(u) < h(v), v \subseteq \sigma\}, \tag{2.37}$$

onde u é um vértice adjacente a v.

Se um triângulo aberto $\tau(u,v,w)$ pertence a $St(v)$, mas não pertence nem a $St^+(v)$ nem a $St^-(v)$, significa que $h(u) \leq h(v) \leq h(w)$, e dizemos que τ tem um *meio* para v.

$$\tau^0(v) = \{\tau \mid \tau \in St(v), \tau \notin St^+(v), \tau \notin St^-(v)\}. \tag{2.38}$$

Esses triângulos são utilizados para calcular o índice de Banchoff de acordo com a Equação 2.39:

$$i(v,h) = 1 - \frac{1}{2}\left|\tau^0(v)\right|. \tag{2.39}$$

Conforme ilustrado na Figura 24, se $St(v) = St^+(v)$, então $i(v,h) = 1$, e v é um mínimo. Da mesma forma, se $St(v) = St^-(v)$, então $i(v,h) = 1$, e v é um máximo. O vértice v é considerado regular se $i(v,h) = 0$. Ele é um ponto de sela se $i(v,h) = -1$, e um ponto de sela degenerado de ordem k se $i(v,h) = -k$ (um ponto de sela de ordem 2 é conhecido como ponto de sela de macaco).

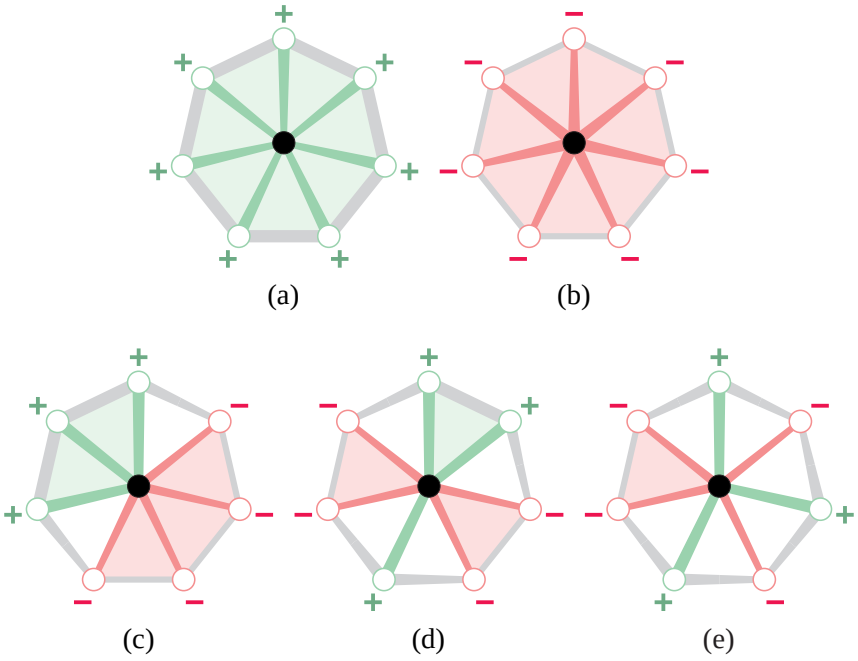


Figura 24: Classificação dos pontos em uma superfície em mínimo (a), máximo (b), regular (c), sela (d) e sela de macaco (e).

2.4 Discussões

Em nossa abordagem, utilizaremos o gradiente e a curvatura dos pixels de borda da imagem para realizar a votação nas células acumuladoras do espaço de parâmetros da HT orientada.

A triangulação das células acumuladoras definirá a variedade LPP e a função de altura h e será dada pela quantidade de votos de cada célula.

Os máximos locais definirão as células candidatas a equação da forma desejada, que poderão ser encontrados através da definição de Banchoff.

3 Transformada de Hough Linear por Partes para Linhas Retas

Neste capítulo, apresentaremos nossa abordagem da HT baseada em Dicionário (DHT) para detecção de linhas retas em três etapas: na Seção 3.1, atribuiremos votos nas células acumuladoras do espaço de parâmetros; na Seção 3.2, identificaremos os máximos locais aplicando a definição de Banchoff dentro da 2-variedade LPP formada pela triangulação das células acumuladoras; e na Seção 3.3, desenvolveremos uma técnica para agrupar os máximos locais que possam indicar a mesma forma dentro da imagem.

Embora ilustremos esse método utilizando a detecção de retas como exemplo, na Seção 3.4 demonstraremos como ele pode ser adaptado para qualquer forma parametrizada.

3.1 Votação nas células acumuladoras

A primeira etapa envolve a definição de retas votantes em células acumuladoras através de pixels de borda selecionados. Embora o processo de seleção dos pixels de aresta não seja obrigatório, é altamente recomendado, pois reduz o número de células votadas, diminuindo significativamente o tempo de processamento e o uso de memória (Seção 4.2.1).

Um método eficaz de seleção envolve a definição de um limiar máximo de curvatura, κ_{max}. Um pixel de borda na imagem com um valor de curvatura muito elevado pode indicar curvas acentuadas ou junções de segmentos (cantos). Em contraste, um valor de curvatura mais baixo sugere que o pixel faz parte de um segmento reto ou de uma curva suave, que é exatamente o que buscamos.

Assim, cada pixel de borda com curvatura menor ou igual a κ_{max} define uma reta

(ρ, θ) que vota em uma célula acumuladora. As coordenadas da célula são determinadas como os valores arredondados de ρ e θ, com d_ρ e d_θ casas decimais, respectivamente (Algoritmo 1). Por exemplo, se $d_\rho = 0$ e $d_\theta = 2$, as retas (35,874 , 1,2584) e (35,163 , 1,2506) irão votar na célula de coordenada (35 , 1,25), enquanto a reta (62,413 , $-$ 0,3144) irá votar na célula de coordenada (62 , $-$ 0,31).

As células que recebem votos são armazenadas em um dicionário durante o processo, onde a chave é dada pela coordenada da célula e o seu valor é a quantidade de votos recebidas. Assim, as células não são armazenadas enquanto não receberem algum voto, portanto não ocupam memória, e as células votadas podem ser facilmente acessadas e atualizadas a medida que o processo de votação ocorre.

Algoritmo 1: Processo de votação.

Dados: $imgCinza$, $imgBorda$, κ_{max}, d_ρ, d_θ
Resultado: $celulas$
$\theta \leftarrow$ direção do gradiente de $imgCinza$;
$\kappa \leftarrow$ curvatura de $imgCinza$;
$celulas \leftarrow$ dicionário vazio onde a chave será a coordenada da célula e o valor serão os votos;
para p *de borda em* $imgBorda$ **faça**
  se $\kappa_p \leq \kappa_{max}$ **então**
    $\rho_p \leftarrow x_p \cos\theta_p + y_p \sin\theta_p$;
    $celula\,[arredonda(\rho_p, d_\rho), arredonda(\theta_p, d_\theta)] + +$;
  fim
fim

3.2 Detecção de máximos

Como é esperado que as células acumuladoras que representem uma reta na imagem recebam mais votos, podemos descartar com segurança todas as células acumuladoras com somente um voto, a fim de minimizar ainda mais o uso de memória e o tempo de processamento. Desta forma, definimos espaço de parâmetros como uma 2-variedade LPP formada pela triangulação das células restantes a fim de determinar a relação de adjacências dessas células e encontrar os máximos locais através da definição de Banchoff.

Foi utilizada a triangulação de Delaunay (DELAUNAY, 1934) neste processo por ser consolidada na literatura e conter as seguintes propriedades:

- *Circunferência vazia.* A circunferência circunscrita de cada triângulo da triangulação não contém pontos interiores;
- *Otimalidade local.* A conexão entre pontos adjacentes na triangulação é a mais curta possível, ou seja, o comprimento das arestas dos triângulos é minimizado;

o que garante a minimização dos comprimentos das arestas e evita, sempre que possível, triângulos excessivamente alongados, o que resulta em estrelas mais uniformes nos locais onde as células estão mais aglomeradas.

Com a 2-variedade LPP definida, os máximos locais podem ser identificados usando a definição de Banchoff seguindo o Algoritmo 2.

Algoritmo 2: Detecção de máximos.

Dados: *celulas*
Resultado: *maxCelulas*
tri ←Triangulação de Delaunay formada pelas chaves de *celulas* com valor maior que 1;
maxCelulas ← dicionário vazio onde a chave será a coordenada da célula e o valor serão os votos;
para *coord em tri* **faça**
  max ← verdadeiro;
  $coords_{adj}$ ← células adjacentes à *coord*;
  para $coord_{adj}$ *em* $coords_{adj}$ **faça**
    se $celulas[coord_{adj}] > celulas[coord]$ **então**
      max ← falso;
      quebra;
    fim
  fim
  se *max* **então**
    inserir *celula* em *maxCelulas*;
  fim
fim

A Figura 25 ilustra o espaço de parâmetros LPP formado pela triangulação de Delaunay e as retas correspondentes na imagem (células de máximos destacadas em vermelho). Embora haja triângulos bem alongados por conta da configuração do espaço de parâmetros, é possível observar que a triangulação de Delaunay forma estrelas uniformes nas regiões de maior concentração de células acumuladoras, onde geralmente estão localizados os máximos locais de interesse.

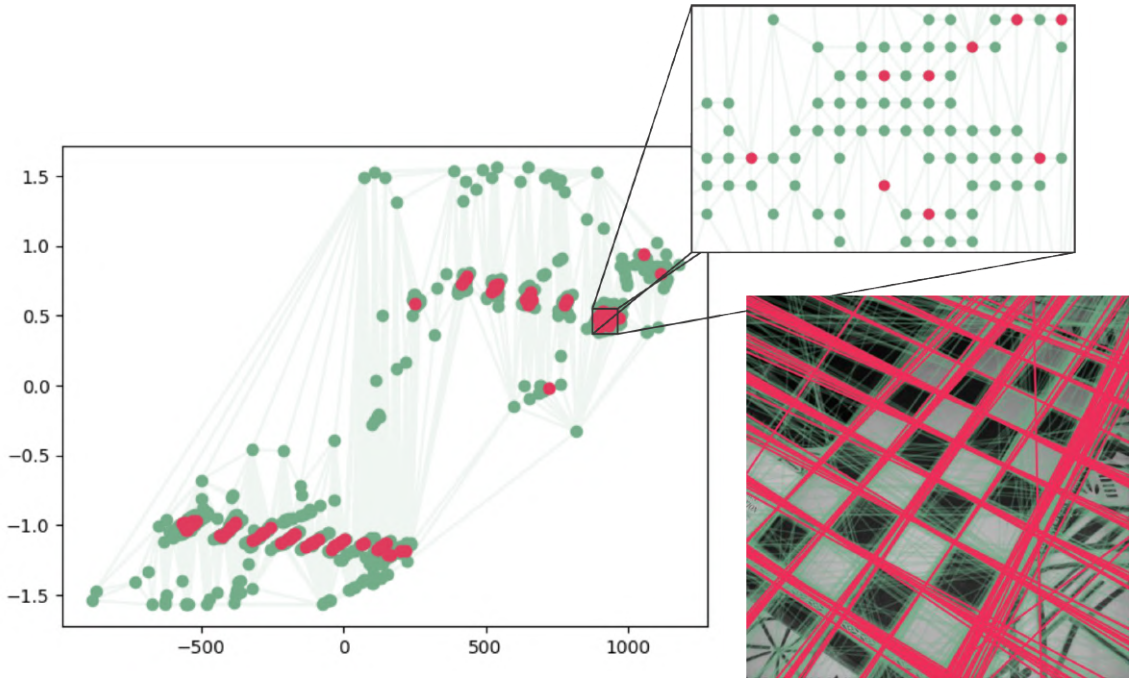


Figura 25: Espaço de parâmetros LPP (esquerda) e retas correspondentes na imagem (direita).

3.3 Agrupamento de máximos

Conforme é possível observar na Figura 25, muitas células de máximos correspondem à mesma reta. Como vimos na seção 2.2, isso ocorre devido à discretização da imagem e do espaço de parâmetros. Desta forma é necessário desenvolver um método para agrupar as células espacialmente próximas.

Uma vez que as células de máximos são identificadas, podemos estabelecer uma nova triangulação, considerando apenas essas células e descartando as outras.

A célula com mais votos é escolhida para definir o primeiro agrupamento. Em seguida, verifica-se se cada célula com distância 1 (células adjacentes) está dentro de uma determinada área ao redor do centro de algum agrupamento; se estiver, a célula passa a fazer parte do agrupamento; se não estiver, define um novo agrupamento. Após verificar todas as células adjacentes, a verificação continua com as células com distância 2, e assim por diante, até que todas as células tenham sido verificadas. O Algoritmo 3 fornece um exemplo de implementação deste processo.

A posição do centro do agrupamento é a média ponderada das posições das células que o compõem, e o número de votos é a soma dos votos dessas células. Tais

Algoritmo 3: Agrupamento de máximos.

Dados: $maxCelulas, r_\rho, r_\theta, v_{min}$
Resultado: $retas$
$tri \leftarrow$Triangulação de Delaunay formada pelas chaves de $maxCelulas$;
$retas \leftarrow$ lista vazia;
$agrCentro \leftarrow$ lista vazia;
$agrVotos \leftarrow$ lista vazia;
$celulasVerif \leftarrow$ lista vazia;
$proxCelulas \leftarrow$ lista contendo a chave de $maxCelulas$ com maior valor;
enquanto *proxCelulas não for vazio,* **faça**
  inserir $proxCelulas$ em $celulasVerif$;
  para *coord em proxCelulas* **faça**
    se *coord está dentro da área* $2r_\rho \times 2r_\theta$ *ao redor de algum centro em* $agrCentro$ **então**
      $i \leftarrow$ posição de $centro$ em $agrCentro$;
      $agrCentro\,[i] \leftarrow$ média ponderada entre $agrCentro$ e $coord$;
      $agrVotos\,[i] \leftarrow agrVotos\,[i] + maxCelulas[coord]$;
    senão
      inserir $coord$ em $agrCentro$;
      inserir $maxCelulas[coord]$ em $agrVotos$;
    fim
    inserir células adjacentes à $coord$ em $proxCelulas$;
  fim
  remover $celulasVerif$ de $proxCelulas$;
fim
$retas \leftarrow agrCentro$ com $votos \geq v_{min}$

valores devem ser ajustados sempre que uma nova célula é integrada ao agrupamento.

É importante observar que definir o centro dessa maneira pode não corresponder necessariamente à posição de uma célula acumuladora. Isso é vantajoso, considerando os argumentos apresentados anteriormente.

Ao final do processo, o centro de cada agrupamento com um número de votos igual ou superior ao limiar v_{min} corresponde a uma reta na imagem (Figura 26).

3.4 Outras formas geométricas

A generalização da abordagem proposta para detecção de outras formas geométricas é relativamente simples, visto que as técnicas utilizadas no espaço de parâmetros de dimensão 2 é aplicável em qualquer dimensão.

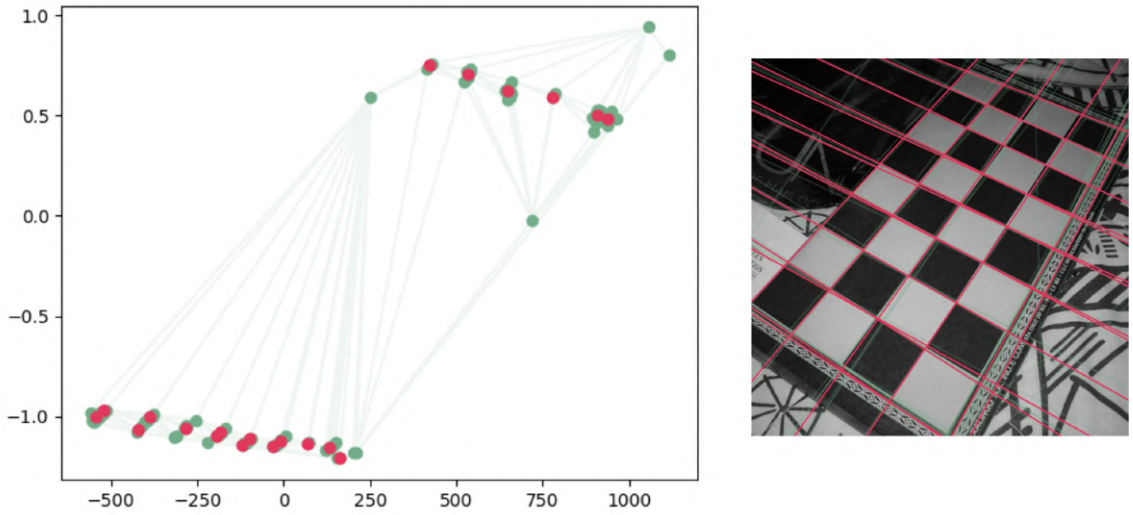


Figura 26: Espaço de Parâmetros LPP de máximos em verde e centros de agrupamentos com $votes \geq v_{min}$ em vermelho (esquerda), e linhas correspondentes na imagem (direita).

Dado um conjunto de m entradas, onde m é a dimensão do espaço de parâmetros, e a função modelo da forma geométrica, é possível realizar a votação nas células acumuladoras conforme descrito na Seção 2.2.2.

A triangulação de Delaunay pode ser generalizada à sua forma m-dimensional, que leva a geração de uma malha de triângulos (espaço de parâmetros 2-D), ou de tetraedros (espaço de parâmetros 3-D) e assim por diante. Porém, torna-se mais complexa em dimensões mais altas devido a maior diversidade de estruturas, o que pode aumentar consideravelmente o tempo de processamento e a memória utilizada.

A definição de Banchoff também pode ser utilizada para encontrar pontos de máximo em dimensões mais altas, pois a Equação 2.37 permanece a mesma, ou seja, somente comparar a quantidade de votos de uma célula com suas células adjacentes já basta.

Por fim, a generalização do método de agrupamento de máximos também é trivial, pois o conceito de adjacência e média ponderada se mantém, e a área ao redor do centro de um agrupamento é generalizada a um m-volume.

4 Experimentos

Neste capítulo, relataremos os testes feitos utilizando o algoritmo proposto no Capítulo 3. Na Seção 4.1, falaremos sobre a metodologia utilizada; e na Seção 4.2, apresentaremos os resultados e as discussões sobre os testes de desempenho realizados para definir e verificar a eficiência do limiar κ_{max}, o tempo de cada etapa do processo e a eficiência da técnica em si, e sobre as retas detectadas nas imagens.

4.1 Metodologia

A Transformada de Hough padrão (SHT) e a Transformada de Hough baseada em Kernel (KHT) (FERNANDES; OLIVEIRA, 2008) (a versão mais rápida da HT relatada na literatura) foram selecionadas para comparação com a abordagem proposta (DHT). Desta forma, todas as técnicas foram implementadas usando C++ e testadas em um conjunto de imagens idênticas de diversos tamanhos.

As imagens foram geradas a partir da vetorização da imagem original a fim de manter a qualidade das bordas ao escalar o tamanho. Antes do processamento, foi utilizado um filtro Gaussiano para suavizar as bordas rígidas e minimizar as imperfeições provenientes da vetorização (Figura 27).

Cada valor apresentado representa uma média de 30 testes realizados, e os picos de memória não consideram a memória para armazenamento das imagens (Tabela 1).

Os experimentos relatados foram realizados em um processador CPU AMD Ryzen 7 1800X Octa-Core com 3,6 GHz e 16 GB de RAM.

Figura 27: Processo de geração das imagens. Imagem original (esquerda), imagem vetorizada (centro) e imagem final (direita).

Tabela 1: Memória de armazenamento utilizada para cada tamanho de imagem.

Tamanho (px)	Memória (MiB)
500×500	141,367
1000×1000	151,367
2000×2000	176,945
4000×4000	288,797
8000×8000	739,113
16000×16000	1659,382
32000×32000	3984,770
64000×64000	11620,930

4.2 Resultados e discussões

Nesta seção, relataremos e discutiremos o resultado dos experimentos realizados utilizando os algoritmos propostos no Capítulo 3 para testar o desempenho da DHT. Na Seção 4.2.1, definiremos um valor padrão para o limiar κ_{max} e discutiremos a sua importância no desempenho da técnica; na Seção 4.2.2, veremos o tempo de processamento dos algoritmos em relação ao tempo de processamento total; na Seção 4.2.3 exibiremos o resultado da detecção de retas da DHT para algumas imagens; e, por fim, na Seção 4.2.4, compararemos a DHT com as outras técnicas selecionadas a fim de analisar o desempenho e a qualidade do resultado da técnica proposta.

4.2.1 Teste dos limiares

Diversos testes foram realizados para estabelecer um valor padrão para o limiar κ_{max}. Os resultados mais satisfatórios foram obtidos com $\kappa_{max} = 0{,}02$, onde algumas

retas mais pronunciadas podem ser identificadas mais de uma vez, mas retas menos pronunciadas possuem maior chance de serem identificadas (Figura 28). Note que, ao utilizar o valor padrão de κ_{max}, quanto mais precisas forem as retas da imagem, maior será o valor v_{min}.

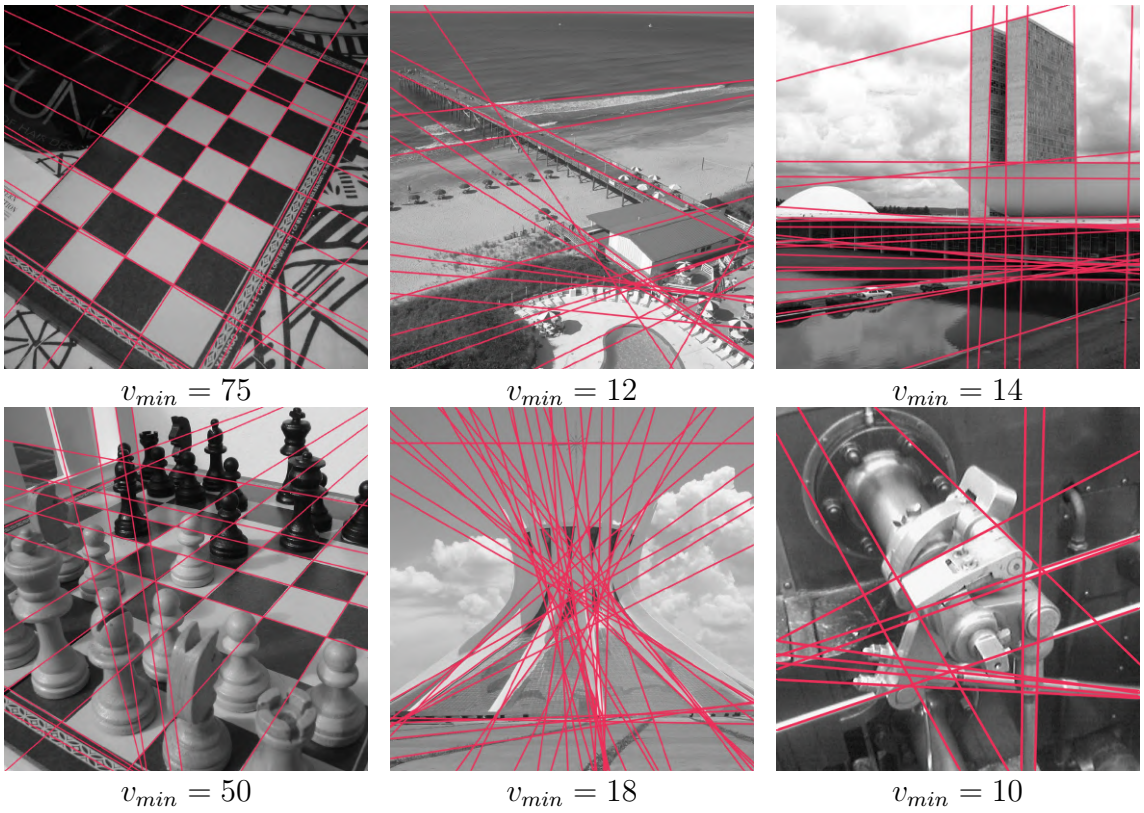


Figura 28: Retas identificadas pela abordagem proposta usando os valores padrão definidos para os limiares κ_{max} e v_{min}.

Para testar a eficiência do limiar de curvatura, foi desenvolvida uma versão da abordagem proposta sem o uso de κ_{max} e sem quaisquer cálculos utilizados para obter o valor de curvatura da imagem. A versão sem o limiar foi comparada com a versão inicialmente proposta utilizando o $\kappa_{max} = 0{,}02$ (definido como padrão), e os resultados podem ser observados nas Figuras 29 e 30. Notavelmente, todos os casos mostraram melhoria de desempenho, potencialmente alcançando um pico de uso de memória 6 vezes menor e um tempo de processamento 7 vezes mais rápido (Tabela 2).

Isso é esperado, pois, embora haja o cálculo adicional da curvatura, o limiar κ_{max} reduz a quantidade de células logo no início do processo, fazendo que todas as etapas futuras sejam mais rápidas e mais leves.

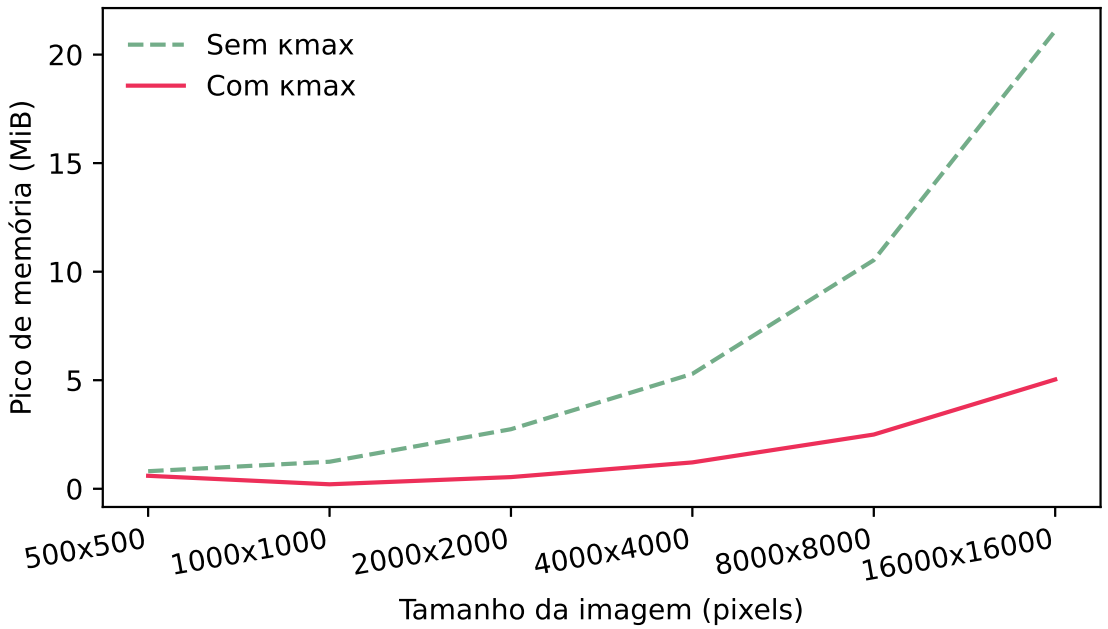


Figura 29: Pico de memória vs. tamanho da imagem para a DHT com e sem o limiar κ_{max}.

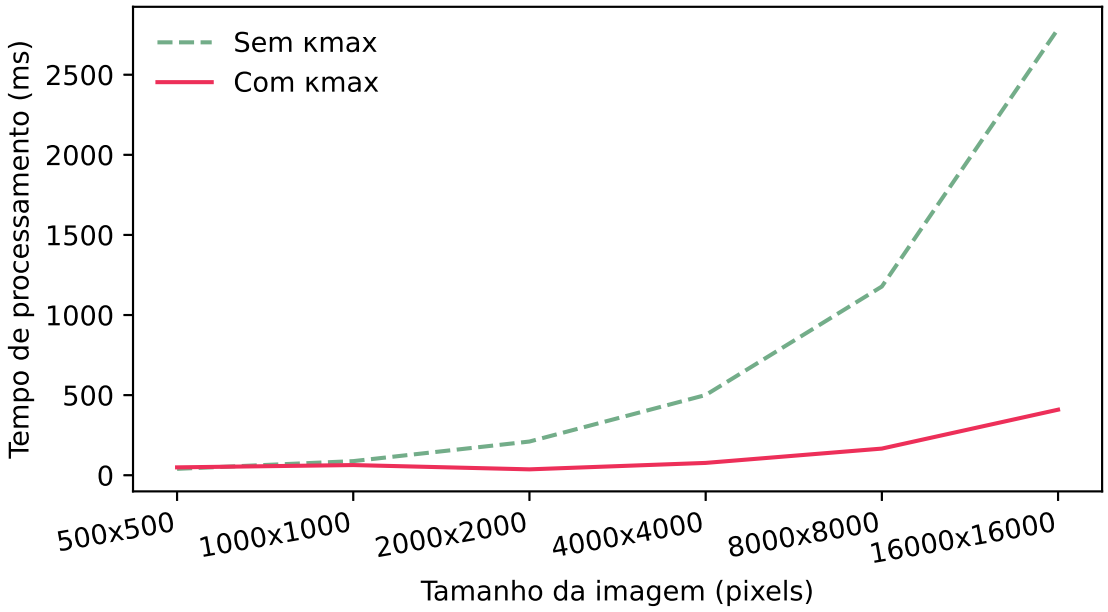


Figura 30: Tempo total de processamento vs. tamanho da imagem para a DHT com e sem o limiar κ_{max}.

Tabela 2: Valores médios de pico de memória e tempo de processamento obtidos nos testes da DHT com e sem o limite κ_{max} para a mesma imagem de tamanhos variados.

Tamanho (px)	Pico de memória (MiB)		Processamento (ms)	
	Com κ_{max}	Sem κ_{max}	Com κ_{max}	Sem κ_{max}
500×500	**0,597**	0,808	**49,207**	88,136
1000×1000	**0,206**	1,245	**63,569**	67,985
2000×2000	**0,538**	2,743	**36,809**	210,432
4000×4000	**1,214**	5,294	**77,019**	500,752
8000×8000	**2,498**	10,531	**166,400**	1178,026
16000×16000	**5,034**	21,098	**409,192**	2786,379

4.2.2 Tempo por etapa

Com o cálculo do limiar κ_{max}, é possível observar na Figura 31 que a etapa de votação nas células acumuladoras é a que possui maior tempo de processamento, chegando a compor mais de 98% do tempo de processamento da técnica para imagens com mais de 250 milhões de pixels (Tabela 3).

Este valor aumenta com o tamanho da imagem por conta da maior quantidade de informações presentes, o que aumenta a quantidade de pixels colineares nas retas diagonais, tornando o cálculo da curvatura mais preciso.

Como consequência, mais pixels são barrados no limiar κ_{max}, reduzindo proporcionalmente o tempo de processamento das etapas de detecção e agrupamento de máximos. Este detalhe faz com que o pico de memória da imagem de tamanho 500×500 seja superior ao da imagem de tamanho 1000×1000 para a imagem selecionada.

Porém, é possível observar que a imagem de tamanho 2000×2000 leva proporcionalmente menos tempo na primeira etapa que a imagens menores, o que resultou em um tempo de processamento total inferior, visto que esta etapa é a mais demorada. Tal irregularidade pode ocorrer de acordo com o conteúdo da imagem.

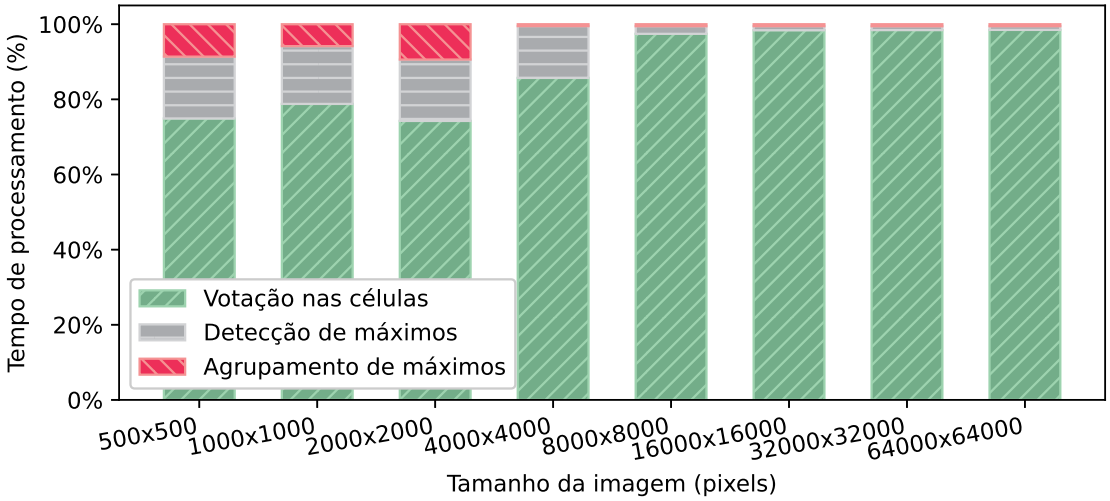


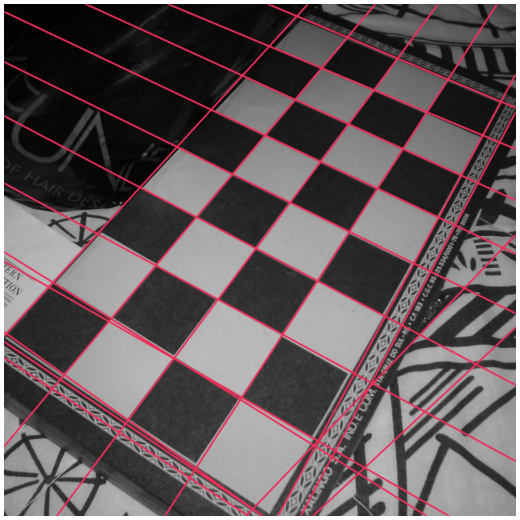
Figura 31: Tempo percentual cada etapa do processamento vs. tamanho da imagem para a DHT.

Tabela 3: Valores médios de porcentagem do tempo de processamento de cada etapa em relação ao tempo total de processamento obtidos nos testes da DHT para a mesma imagem de tamanhos variados.

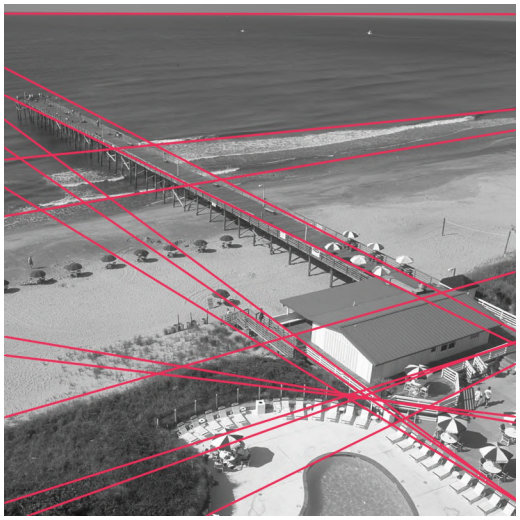
Tamanho (px)	Tempo de processamento por etapa (%)		
	Votação nas cél.	Detecção de máx.	Agrupam. de máx.
500×500	74,90	16,48	8,62
1000×1000	78,79	15,38	5,83
2000×2000	74,26	16,35	9,39
4000×4000	85,71	13,79	0,50
8000×8000	97,45	2,03	0,52
16000×16000	98,40	1,09	0,51
32000×32000	98,45	1,02	0,53
64000×64000	98,50	0,99	0,51

4.2.3 Detecção de retas

Ajustando os limiares da DHT, é possível observar na Figura 32 que a técnica é capaz de identificar as retas presentes na imagem, principalmente as bem definidas, porém é imprecisa em alguns casos e não agrupa devidamente em outros, gerando falsos positivos. Isso acontece porque a triangulação e o agrupamento de máximos não leva em conta a distorção do espaço de parâmetros $\rho\theta$.

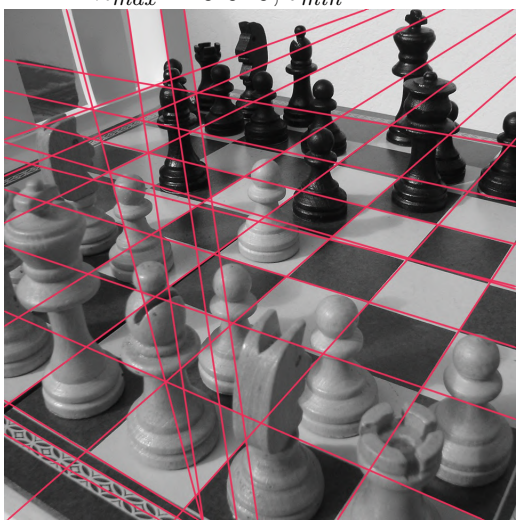


$\kappa_{max} = 0.008, v_{min} = 30.$

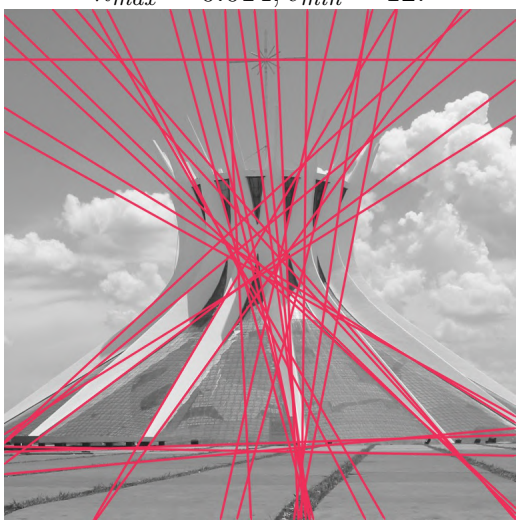


$\kappa_{max} = 0.020, v_{min} = 14.$

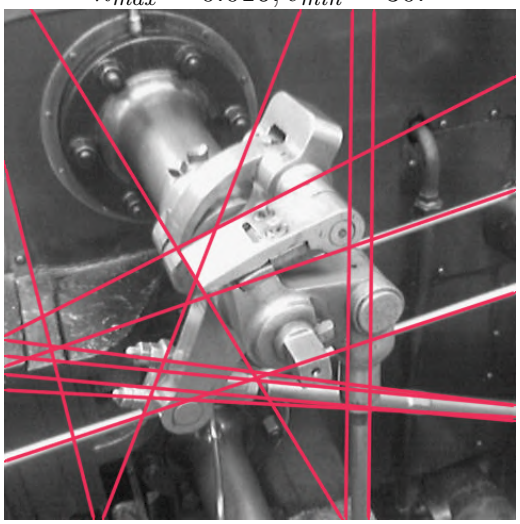
$\kappa_{max} = 0.014, v_{min} = 12.$



$\kappa_{max} = 0.016, v_{min} = 30.$



$\kappa_{max} = 0.018, v_{min} = 20.$



$\kappa_{max} = 0.023, v_{min} = 14.$

Figura 32: Resultado da DHT com os limiares ajustados de acordo com a legenda de cada imagem.

4.2.4 Comparação entre técnicas

Os valores obtidos ao comparar as três técnicas são apresentados na Tabela 4. Pode-se observar que a abordagem proposta (DHT) possui um requisito máximo de memória, em média, 10,9 vezes menor do que a Transformada de Hough padrão (SHT) e 7,7 vezes menor do que a Transformada de Hough baseada em Kernel (KHT) (Figura 33).

Em relação ao tempo de processamento, a DHT inicializa mais lentamente do que a KHT. No entanto, essa diferença diminui com o aumento do tamanho da imagem, e a DHT torna-se mais rápida em imagens com mais de 250 milhões de pixels. Apesar disso, a DHT mostra-se 5,9 vezes mais rápida que a SHT (Figura 34). Por fim, a DHT foi a única técnica capaz de processar a última imagem por conta da memória necessária.

As retas encontradas nas três técnicas durante os testes das imagens de até 1 bilhão de pixels podem ser analisadas nas Figuras 35 a 40. A SHT e a DHT possuem um limiar de votações mínimas v_{min}, enquanto a KHT possui um limiar de número de retas n_{retas}. Tais limiares foram ajustados manualmente a fim de obter o melhor resultado para cada imagem e este ajuste não influencia nos resultados obtidos de tempo de processamento e uso de memória das técnicas, apenas limita a quantidade de retas retornadas.

É possível observar que a KHT é a mais precisa dentre as técnicas, porém o resultado obtido pela DHT assemelha-se ao da KHT e é mais preciso que a SHT.

Tabela 4: Valores médios de pico de memória e tempo de processamento obtidos nos tesdes da DHT, KHT e GHT para a mesma imagem de tamanhos variados.

Tamanho (px)	Pico de memória (MiB)			Processamento (ms)		
	DHT	KHT	SHT	DHT	KHT	SHT
500×500	**0,597**	3,000	2,634	49,207	**8,537**	56,635
1000×1000	**0,206**	2,694	3,875	63,569	**13,103**	136,222
2000×2000	**0,538**	3,906	5,893	36,809	**21,711**	298,638
4000×4000	**1,214**	7,816	4,415	77,019	**54,491**	767,343
8000×8000	**2,498**	15,628	38,578	166,400	**158,772**	1526,240
16000×16000	**5,034**	39,307	77,152	**409,192**	559,647	3135,118
32000×32000	**20,746**	166,699	161,620	**1012,371**	2510,929	6937,902
64000×64000	**66,220**	-	-	**2587,412**	-	-

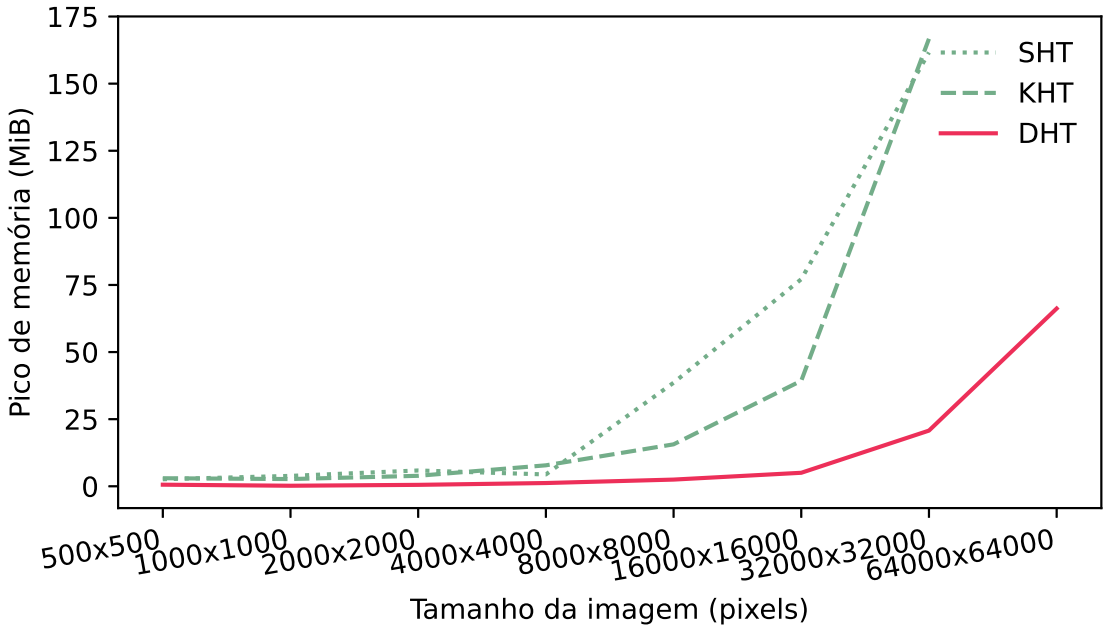


Figura 33: Pico de memória vs. tamanho da imagem para as técnicas SHT, KHT e DHT.

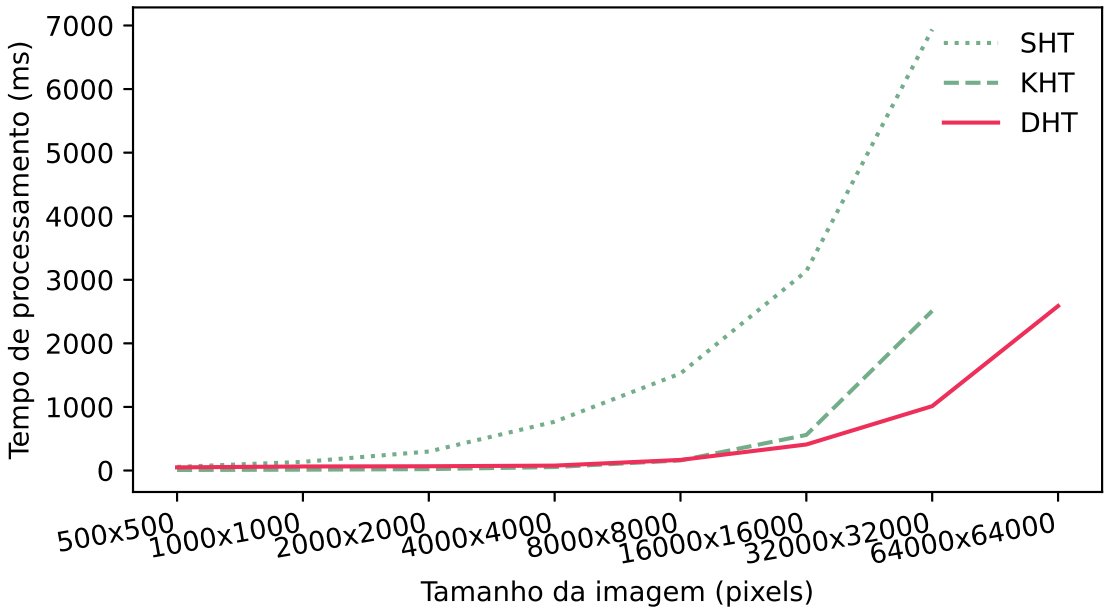
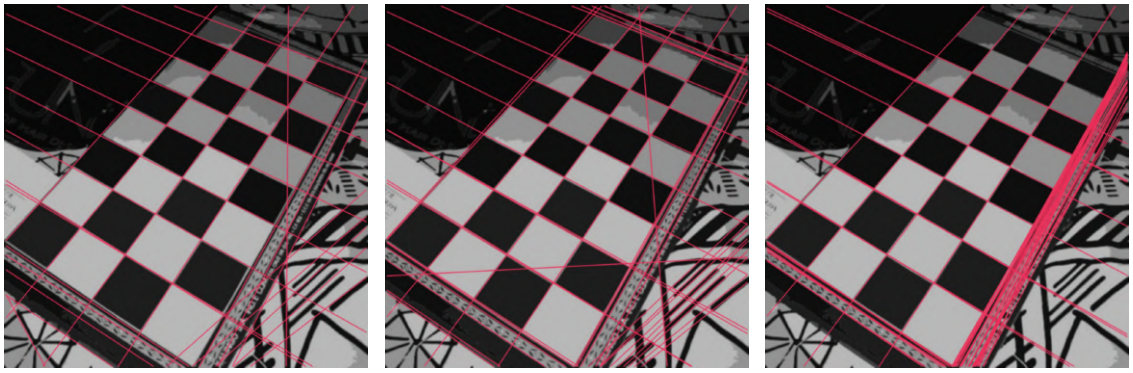
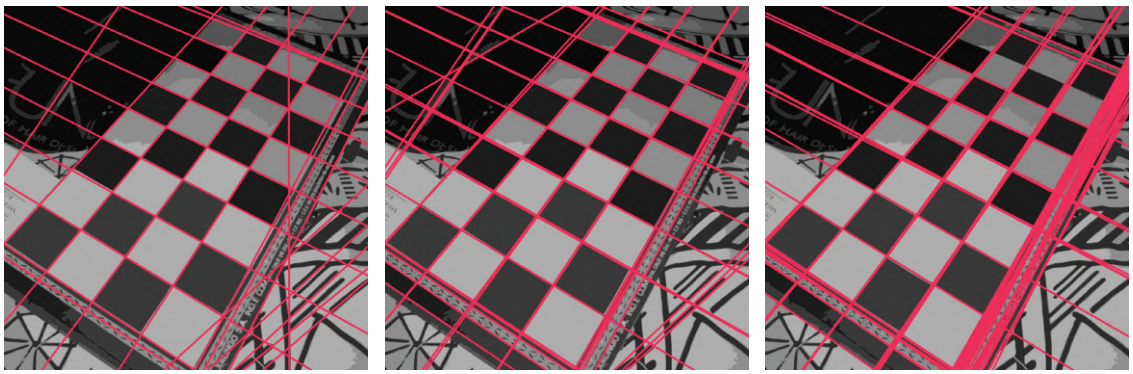


Figura 34: Tempo total de processamento vs. tamanho da imagem para as técnicas SHT, KHT e DHT.



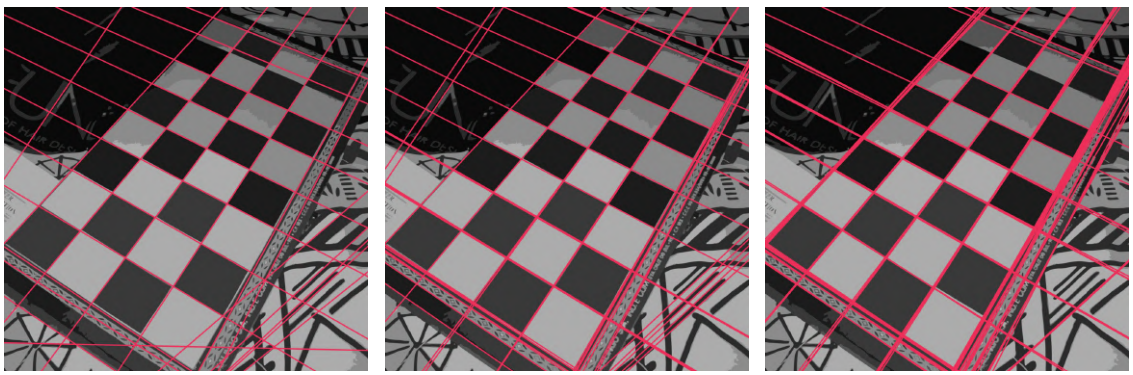
DHT ($v_{min} = 35$) KHT ($n_{retas} = 40$) SHT ($v_{min} = 150$)

Figura 35: Comparação das técnicas para imagem de tamanho 500×500.



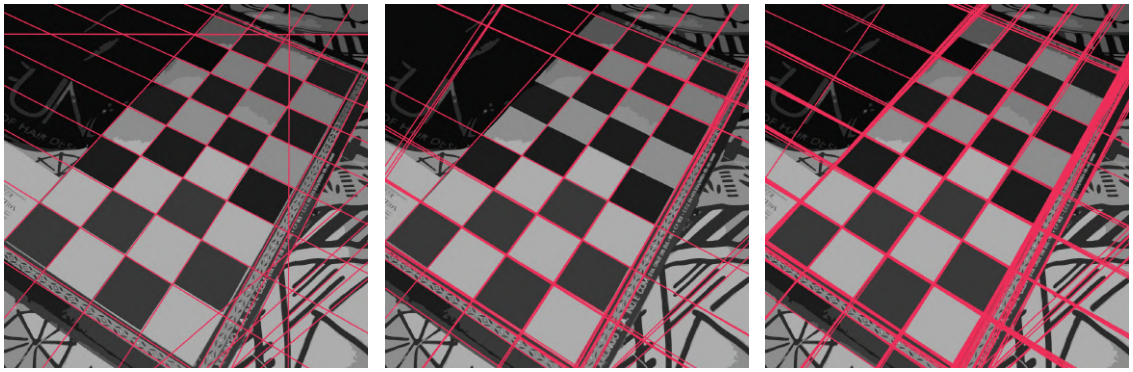
DHT ($v_{min} = 80$) KHT ($n_{retas} = 40$) SHT ($v_{min} = 200$)

Figura 36: Comparação das técnicas para imagem de tamanho 1000×1000.



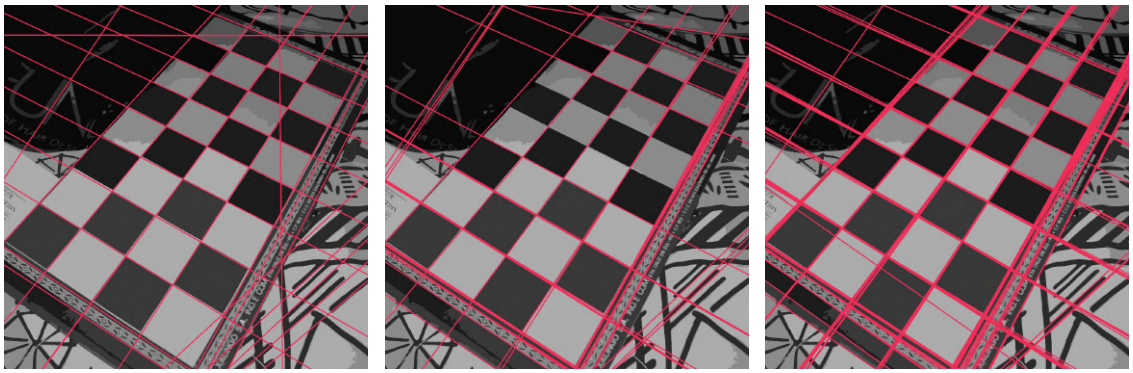
DHT ($v_{min} = 190$) KHT ($n_{retas} = 40$) SHT ($v_{min} = 300$)

Figura 37: Comparação das técnicas para imagem de tamanho 2000×2000.



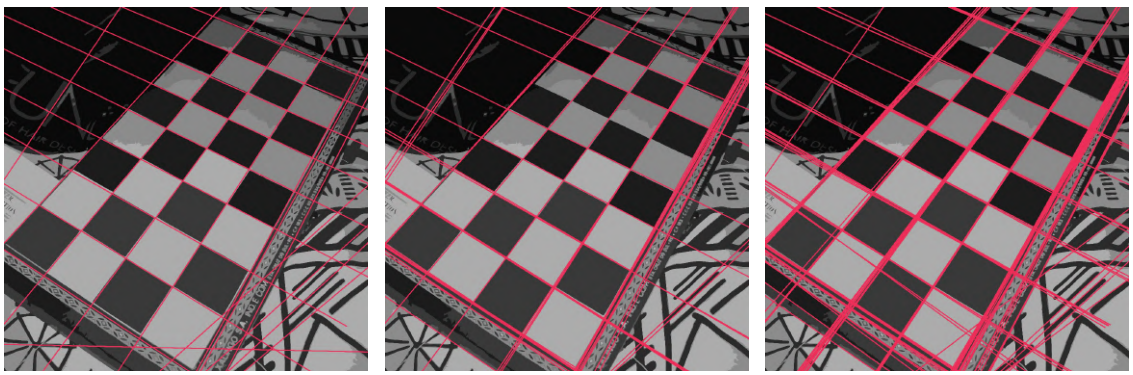
DHT ($v_{min} = 380$) KHT ($n_{retas} = 40$) SHT ($v_{min} = 350$)

Figura 38: Comparação das técnicas para imagem de tamanho 4000×4000.



DHT ($v_{min} = 750$) KHT ($n_{retas} = 40$) SHT ($v_{min} = 500$)

Figura 39: Comparação das técnicas para imagem de tamanho 8000×8000.



DHT ($v_{min} = 1700$) KHT ($n_{retas} = 40$) SHT ($v_{min} = 650$)

Figura 40: Comparação das técnicas para imagem de tamanho 16000×16000.

5 Conclusões e Trabalhos Futuros

Neste capítulo, apresentaremos as principais discussões deste trabalho, as limitações da técnica e os trabalhos futuros a serem desenvolvidos.

5.1 Discussões

Introduzimos uma técnica que armazena apenas as células acumuladoras mais relevantes do espaço de parâmetros, estabelece uma topologia por meio da triangulação de Delaunay dessas células e detecta máximos locais usando a definição de Banchoff para identificar retas em uma imagem. Com a pré-seleção de pontos de borda como votantes, o descarte de células acumuladoras menos relevantes e a seleção de pontos de máximos locais, o espaço de parâmetros LPP torna-se cada vez mais enxuto, resultando em uma melhoria na eficiência computacional.

Os resultados demonstraram que a DHT requer consideravelmente menos memória do que a SHT e a KHT, especialmente para imagens com mais de 1 bilhão de pixels. Além disso, é significativamente mais rápida que a SHT em todos os tamanhos de imagem e torna-se mais rápida que a KHT em imagens com mais de 250 milhões de pixels. Entre as técnicas analisadas, a DHT foi a única capaz de processar imagens com mais de 1 bilhão de pixels.

A importância do uso do dicionário, do armazenamento somente as células votadas no espaço de parâmetros e do limite de curvatura para redução do tempo de processamento e uso de memória também foi enfatizada, possibilitando a detecção de retas em imagens com mais de 4 bilhões de pixels em menos de 3 segundos.

Essa técnica pode ser facilmente adaptada para detectar formas geométricas com espaços de parâmetros em dimensões mais altas, já que todas as técnicas empregadas podem ser aplicadas em espaços n-dimensionais.

5.2 Limitações

A triangulação de Delaunay e o agrupamento de máximos não levam em consideração e distorção do espaço de parâmetros $\rho\theta$, fazendo tanto com que a adjacência quanto o agrupamento de alguns pontos não sejam calculados corretamente. Isso diminui a precisão das retas encontradas e gera falsos positivos devido a retas detectadas duplamente.

Por mais que a técnica possa ser facilmente adaptada para detecção de outras formas geométricas, não sabemos como será o desempenho em comparação com as técnicas já existentes na literatura por conta do aumento de complexidade da triangulação.

5.3 Trabalhos futuros

Futuramente, esperamos otimizar ainda mais a etapa de detecção de células a fim de reduzir o tempo de processamento da técnica.

Além disso, desejamos melhorar o agrupamento de máximos, melhorando a técnica abordada ou utilizando outras técnicas, como a detecção máxima iterativa para evitar que contagens votem em mais de uma reta e um ajuste pós-processamento por mínimos quadrados ortogonais apresentado em Dalitz, Schramke e Jeltsch (2017).

Também pretendemos implementar a técnica para outras formas geométricas, como círculos, parábolas e elipses, a fim de verificar se o ganho de memória e tempo de processamento sem mantém relevante nesses casos.

Por fim, planejamos utilizar a técnica desenvolvida para detectar quaisquer formas analíticas através da substituição do mapa de votos denso na transformada geral apresentada em Fernandes e Oliveira (2014) por um mapa de votos LPP.

REFERÊNCIAS

BALLARD, D. H. Generalizing the Hough transform to detect arbitrary shapes. **Pattern Recognition**, v. 13, n. 2, p. 111–122, jan. 1981. DOI: 10.1016/0031-3203(81)90009-1.

BANCHOFF, T. F. Critical points and curvature for embedded polyhedral surfaces. **The American Mathematical Monthly**, v. 77, n. 5, p. 475–485, 1970. DOI: 10.1080/00029890.1970.11992523.

BENNETT, N.; BURRIDGE, R.; SAITO, N. A method to detect and characterize ellipses using the Hough transform. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 21, n. 7, p. 652–657, jul. 1999. DOI: 10.1109/34.777377.

BOISSONNAT, J.-D. Geometric structures for three-dimensional shape representation. **ACM Transactions on Graphics**, v. 3, n. 4, p. 266–286, 1984. DOI: 10.1145/357346.357349.

BOLLAND, P. et al. Gradient-based Hough transform for the detection and characterization of defects during nondestructive inspection. In: PROCEEDINGS of IEEE/ASME International Conference on Advanced Intelligent Mechatronics. [S. l.: s. n.], 1997.

BRUZZONE, E. et al. Applying two-dimensional Delaunay triangulation to stereo data interpolation. In: SPRINGER. PROCEEDINGS of the European Conference on Computer Vision (ECCV). [S. l.: s. n.], 1992. P. 368–372.

CANNY, J. A computational approach to edge detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, PAMI-8, n. 6, p. 679–698, 1986. DOI: 10.1109/TPAMI.1986.4767851.

DALITZ, C.; SCHRAMKE, T.; JELTSCH, M. Iterative Hough Transform for Line Detection in 3D Point Clouds. **Image Processing On Line**, v. 7, p. 184–196, 2017. DOI: 10.5201/ipol.2017.208.

DEBEVEC, P. E.; TAYLOR, C. J.; MALIK, J. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In: PROCEEDINGS of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. [S. l.: s. n.], 1996. P. 11–20. DOI: 10.1145/237170.237191.

DELAUNAY, B. Sur la sphére vide. **Bulletin de l'Acad'emie des Sciences de l'URSS. Classe des sciences math'ematiques et na**, v. 1934, n. 6, p. 793–800, 1934.

DUDA, R. O.; HART, P. E. Use of the Hough transformation to detect lines and curves in pictures. **Communications of the ACM**, v. 15, n. 1, p. 11–15, jan. 1972. DOI: 10.1145/361237.361242.

EL BAHI, H.; ZATNI, A. Text Recognition in Document Images Obtained by a Smartphone Based on Deep Convolutional and Recurrent Neural Network. **Multimedia Tools Appl.**, v. 78, n. 18, p. 26453–26481, 2019. DOI: 10.1007/s11042-019-07855-z.

FAUGERAS, O. D. et al. Polyhedral approximation of 3-d objects without holes. **Computer Vision, Graphics, and Image Processing**, v. 25, n. 2, p. 169–183, 1984.

FERNANDES, L. A. F.; OLIVEIRA, M. M. Handling uncertain data in subspace detection. **Pattern Recognition**, v. 47, n. 10, p. 3225–3241, 2014. DOI: 10.1016/j.patcog.2014.04.013.

_____. Real-time line detection through an improved Hough transform voting scheme. **Pattern Recognition**, v. 41, n. 1, p. 299–314, 2008. DOI: 10.1016/j.patcog.2007.04.003.

FERREIRA, A. et al. Mixing images and sketches for retrieving vector drawings. In_____. **Eurographics Multimedia Workshop**. [S. l.: s. n.], 2004. DOI: 10.2312/EGMM/MM04/069-075.

FRANKE, R. Scattered data interpolation: tests of some method. **Mathematics of Computation**, v. 38, n. 157, p. 181–200, 1982.

GOLDMAN, R. Curvature formulas for implicit curves and surfaces. **Computer Aided Geometric Design**, v. 22, n. 7, p. 632–658, jul. 2005. DOI: 10.1016/j.cagd.2005.06.005.

GONZALEZ, R.; WOODS, R. **Processamento de Imagens Digitais**. [S. l.]: Pearson Universidades, 2010.

GROSSMANN, C.; ROOS, H.G.; STYNES, M. **Numerical treatment of partial differential equations**. [S. l.]: Springer Berlin Heidelberg, 2007. (Universitext).

HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. [S. l.]: Cambridge University Press, 2003. (Cambridge books online).

HOUGH, P. V. C. **Methods and means for recognizing complex patterns**. [S. l.: s. n.], dez. 1962. Patente 3.069.654.

KARNIELI, A. et al. Automatic extraction and evaluation of geological linear features from digital remote sensing data using a hough transform. **Photogrammetric Engineering and Remote Sensing**, v. 62, p. 525–531, mai. 1996.

KIM, S.-S.; CHO, M.-G.; CHO, H.-G. A geometric compression algorithm for massive terrain data using Delaunay triangulation. In: PROCEEDINGS of the International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media (WSCG). [S. l.: s. n.], 1998.

KOO, H. Text-Line Detection in Camera-Captured Document Images Using the State Estimation of Connected Components. **IEEE transactions on image processing : a publication of the IEEE Signal Processing Society**, PP, 2016. DOI: 10.1109/TIP.2016.2607418.

LEE, H.-J. et al. Hough transform for line and plane detection based on the conjugate formulation. In: MACHINE Vision Applications in Industrial Inspection IX. [S. l.: s. n.], 2001. v. 4301, p. 244–252. DOI: 10.1117/12.420918.

LEE, J. et al. Semantic Line Detection and Its Applications. In: 2017 IEEE International Conference on Computer Vision (ICCV). [S. l.: s. n.], 2017. P. 3249–3257. DOI: 10.1109/ICCV.2017.350.

LIEBEHERR, J.; NAHAS, M.; SI, W. Application-layer multicasting with Delaunay triangulation overlays. **IEEE Journal on Selected Areas in Communications**, v. 200, p. 1472–1488, 2002. DOI: 10.1109/JSAC.2002.803067.

MATOUSEK, J. **Using the Borsuk-Ulam Theorem: Lectures on Topological Methods in Combinatorics and Geometry**. [S. l.]: Springer, 2003. (Universitext (Berlin. Print)).

O'GORMAN, F; CLOWES, M. Finding Picture Edges Through Collinearity of Feature Points. **IEEE Trans. Computers**, v. 25, p. 449–456, 1976. DOI: 10.1109/TC.1976.1674627.

RODRÍGUEZ, A. et al. Delaunay surface reconstruction from scattered points. In: DISCRETE Geometry for Computer Imagery. [S. l.: s. n.], 2000. P. 272–283.

ROSENFELD, A. **Picture Processing by Computer**. [S. l.]: Academic Press, 1969. (Computer science and applied mathematics : a series of monographs and textbooks).

SIBSON, R. Locally equiangular triangulations. **The Computer Journal**, v. 21, n. 3, p. 243–245, 1978.

SKLANSKY, J. On the Hough technique for curve detection. **IEEE Transactions on Computers**, v. C-27, n. 10, p. 923–926, out. 1978. DOI: 10.1109/tc.1978.1674971.

SOBEL, I. **An isotropic 3x3 image gradient operator**. [S. l.: s. n.], fev. 1968. Presentation at Stanford A.I. Project.

SZELISKI, R. **Computer Vision: Algorithms and Applications**. [S. l.]: Springer London, 2010. (Texts in Computer Science).

WATSON, D. F. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. **The Computer Journal**, v. 24, n. 2, p. 167–172, 1981. DOI: 10.1093/comjnl/24.2.167.

WILLARD, S. **General Topology**. [S. l.]: Dover Publications, 2004. (Addison-Wesley Series in Mathematics).

YANG, X.; CUI, W. A novel spatial clustering algorithm based on Delaunay triangulation. **Journal of Software Engineering and Applications**, v. 3, n. 2, p. 141–149, 2010. DOI: 10.4236/jsea.2010.32018.

ZAHN, C. T. Graph-theoretical methods for detecting and describing gestalt clusters. **IEEE Transactions on Computers**, v. C-20, n. 1, p. 68–86, 1971. DOI: 10.1109/T-C.1971.223083.