

Estudo e implementação de um algoritmo eficaz de detecção de retas

André M L G Cerqueira
Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
andre.cerqueira@gmail.com

Resumo

Este trabalho é um estudo das idéias apresentadas em "Algorithmic Evaluation of Line Detection Problem," de Tetsuo Asano, que trata o problema de detecção de reta do ponto de vista de complexidade algorítmica, e apresenta algumas idéias de novos algoritmos. Apresenta os resultados obtidos na detecção de retas da implementação de um dos algoritmos propostos, que se propõe a ser eficaz, detectando todas as retas existentes em uma imagem.

1. Introdução

Detecção de retas é um problema recorrente em Visão Computacional. O algoritmo conhecido como Transformada de Hough é um dos mais conhecidos. Ele aplica o paradigma de dualidade, transformando pontos no espaço primal em curvas no espaço dual e calculando suas interseções.

Sua fácil implementação e bom desempenho explicam sua popularidade. No entanto, a definição do de seus parâmetros forçam um compromisso entre a qualidade do resultado e o desempenho.

[1] propõe outra dualidade, onde um ponto é transformado em um par de paralelas. O conjunto de pontos de borda é transformado, então, em um arranjo de retas no plano. O problema passa a ser encontrar neste arranjo as faces máximas, que correspondem as retas detectadas.

2. Revisão Bibliográfica

Este trabalho é completamente baseado em [1]. Lá é feito um estudo algorítmico do problema de detecção de retas, calculando sua complexidade e definindo os limites de algoritmos de votação como de Hough. São propostos três novos algoritmos, um de votação, porém mais eficaz do que Hough, um baseado em arranjo de

retas, que foi o objeto deste trabalho, e um aleatório, que faz uso do algoritmo anterior.

[1] é uma continuação do trabalho em [2], onde, especialmente, o algoritmo aqui implementado foi inicialmente sugerido. No entanto, em [2] o paradigma utilizado foi o de varredura angular.

[3] e [4] tratam do Caminhamento Topológico, que é uma solução mais adequada a problemas onde o arranjo é apenas um passo na solução do problema, como o aqui apresentado.

[5] faz uma revisão extensa dos problemas em Geometria Computacional, inclusive o de reconhecimento de padrões, onde a detecção de retas foi incluída, e o artigo [2] mencionado.

[6] e [7] são literatura fundamentais de suas áreas, de Visão e Geometria Computacional, assim com [8] e [9] são suas bibliotecas padrão.

3. Detecção de Retas

Um processo recorrente em Visão Computacional é o de extração de retas, ou mais precisamente, segmentos de retas, de uma imagem. O processo, normalmente, utiliza a saída de um detector de bordas. Um detector de bordas transforma a imagem original em uma outra onde, para cada *pixel* pertencente a uma borda existe um *pixel* com valor 1, e para todos outros um *pixel* com valor 0.

3. Transformada de Hough Padrão

A transformada de Hough é um método amplamente utilizado para detecção de retas em uma imagem. Ele utiliza-se do paradigma de dualidade, mapeando pontos da imagem do plano de imagem para curvas do plano de parâmetros.

Em cada ponto de borda (x_i, y_i) passam infinitas retas (na verdade não são infinitas, devido ao fato do espaço em questão ser discretizado). Podemos descrever cada uma destas retas por uma equação da forma $\rho = x_i \cdot \cos(\theta) + y_i \cdot \sin(\theta)$, onde ρ é a distância do centro de coordenadas à reta, e θ é o ângulo que a normal à reta faz com o eixo de coordenadas x (veja figura a seguir).

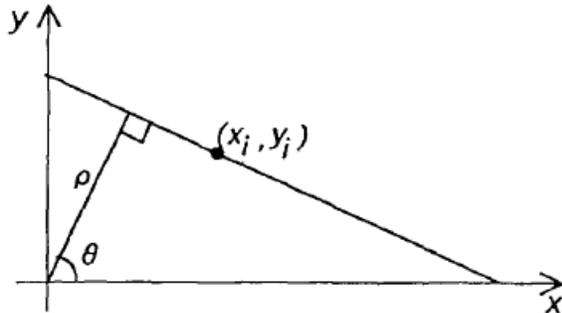


Figura 1 (retirada de [1]).

Assim, pode-se fazer uma transformação de dualidade do plano primal xy para o plano dual $\rho\theta$, de forma que cada ponto primal transforma-se em uma curva senoidal no plano dual. Quando duas curvas se interceptam no plano dual significa que existe uma reta passando pelos pontos correspondentes no plano primal.

O problema de encontrar retas passa a ser, então, de encontrar interseções no plano dual. Porém, como se trata de um espaço discretizado, e, também, os pontos foram gerados a partir de uma imagem, é pouco provável que várias curvas se interceptem em um mesmo ponto. É feita, então, uma quantização dos parâmetros ρ e θ , criando então regiões no plano dual. Define-se então um mínimo de interseções τ tal que todas as regiões que contenham este mínimo definem uma reta. Veja a figura 2, onde uma série de curvas é mostrada, assim como seus pontos correspondentes.

Necessários para o funcionamento do algoritmo, a definição dos parâmetros ρ , θ e τ embutem uma solução de compromisso entre a qualidade da detecção e o desempenho em tempo e espaço. Pequenos valores de τ podem gerar retas falsas, enquanto valores muito altos podem excluir retas legítimas. Uma definição que funcione bem para determinada figura, pode não funcionar bem para outra. Muitas vezes, dentro de uma mesma figura, valores diferentes dos parâmetros podem ser mais adequados.

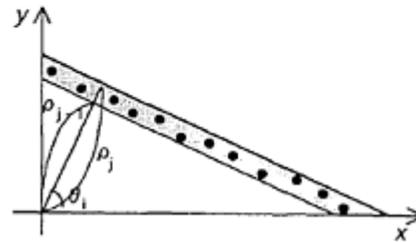
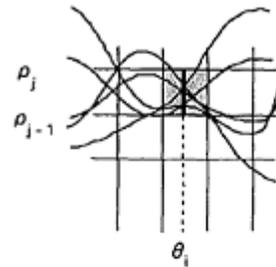


Figura 2 (retirada de [2]).

4. Representação digital de uma reta

Uma reta é uma entidade geométrica, definida no plano contínuo. Ao ser representada em um espaço discretizado, como o de uma imagem, deixa de ser, formalmente, uma reta. Assim define em [1], então, o que chamou de **Representação digital da reta**. Define-se, então, $G(a,b)$ como a representação digital da reta representada por $y = ax+b$. $G(a,b)$ contém todos os *pixels* que estão a uma distância, no máximo, de $0,5 \text{ pixel}$ da reta teórica, como pode ser visto na figura abaixo:

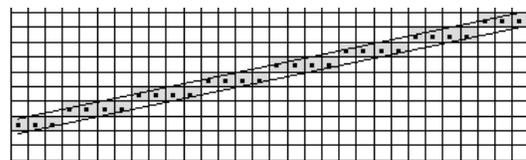


Figura 3

Um conjunto P de pontos de borda é chamado de **componente (digital) de reta** se existem a e b tais que $P \subseteq G(a,b)$. Se a adição de mais um ponto viola a condição anterior, P é dito **máximo**.

5. Definição do problema

Dado um arquivo de bordas de uma imagem, de tamanho $N \times N$, contendo n pontos de borda e um limite τ , reportar todos componentes de reta com tamanho maior do que τ . Adicionalmente, apenas os componentes máximos devem ser listados.

A restrição de imagens com altura e largura iguais é apenas para facilitar a análise, não havendo perda de generalidade. Outra restrição imposta, que também não afeta a generalidade, é de que serão considerados apenas retas com inclinação (parâmetro a) entre 0 e 1. Para as detectar as outras retas, basta efetuar transformações simples na imagem, como inversão da coordenada x e/ou troca das coordenadas x e y . Desta forma, todas as retas são consideradas, inclusive as verticais.

6. Complexidade

Se P é um componente de reta, para todo ponto (x_i, y_i) em P existem constantes a e b tais que:

$$y_i - a \cdot x_i - b \geq -\frac{1}{2}$$

$$y_i - a \cdot x_i - b \leq \frac{1}{2}$$

É possível considerar apenas a distância vertical à reta pois estão sendo considerada apenas retas com inclinações entre 0 e 1, e nestas retas a distância vertical a reta teórica sempre que a distância horizontal.

Com isto em conta, pode-se definir a representação digital como:

$$G(a,b) = \{(x_i, y_i) \in G \mid -\frac{1}{2} \leq y_i - ax_i - b \leq \frac{1}{2}\}$$

Asano prova em [1] que existem a e b racionais:

$$a = \frac{q}{p}$$

$$b = y - \frac{q}{p} \cdot x \pm \frac{1}{2},$$

$$0 \leq q \leq p \leq N-1 \text{ e } 0 \leq x; y \leq N-1$$

a , então, é um membro de uma seqüência de Farey de ordem $N-1$, e pode assumir $3N^2/\pi^2$ valores diferentes, ou seja, sua complexidade é $O(N^2)$.

Para cada valor de a , b pode assumir $O(N^2)$, já que tanto x quanto y são $O(N)$.

Assim, um algoritmo de votação, como o de Hough, que considerasse todos os possíveis valores de a e b para cada ponto de borda teria complexidade $O(nN^4)$.

Asano propôs em [1], e foi implementado como parte deste trabalho, uma algoritmo baseado em arranjo de retas, de complexidade de tempo $O(N^2)$ e de espaço $O(N)$.

7. Arranjo de retas

Um conjunto L de n retas no plano induz uma subdivisão deste em vértices, arestas e faces. Esta subdivisão é chamada de **Arranjo** induzido por L , ou $A(L)$.

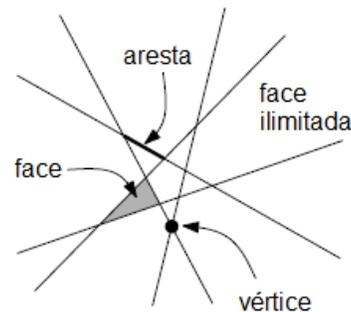


Figura 4 (retirada de [7])

Um arranjo é dito simples se não contém retas paralelas e não mais do que duas retas se interceptam em um mesmo ponto.

Arranjo de retas é uma das estruturas mais importantes em Geometria Computacional.

Para qualquer arranjo, temos os seguintes limites:

$$\text{Número de vértices} \leq n \cdot (n-1) / 2$$

$$\text{Número de arestas} \leq n^2$$

$$\text{Número de faces} \leq n^2 / 2 + n / 2 + 1$$

Para um arranjo simples, as relações acima são de igualdade.

Existe algoritmo ótimo para construção de arranjos, a partir de n retas, com tempo $O(n^2)$.

8. Algoritmo baseado em arranjo de retas

Vimos que um ponto $p=(x,y)$ pertence a componente de reta $G(a,b)$ se:

$$y - ax - b \geq -\frac{1}{2}$$

$$y - ax - b \leq \frac{1}{2}$$

Reescrevendo, temos:

$$b \geq -x \cdot a + y - \frac{1}{2}$$

$$b \leq -x \cdot a + y + \frac{1}{2}$$

Isto define, no plano dual **ab**, um tubo limitado por duas retas paralelas. Cada uma delas tem com inclinação $-x$ e a reta superior intercepta o eixo **b** no ponto $y+\frac{1}{2}$ e a inferior no ponto $y-\frac{1}{2}$. Na figura abaixo, vemos a representação dual do ponto $P=(2,1)$:

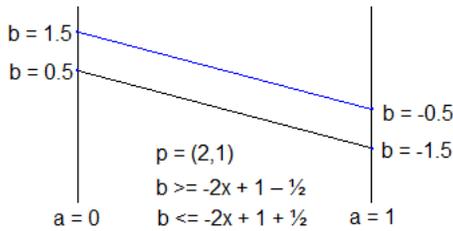


Figura 5

Todos os pontos **(a,b)** contidos em um tubo, definem componentes de reta **G(a,b)** que contem (x,y) .

Da mesma forma, todos os pontos **(a,b)** contidos na interseção de vários tubos definem componentes de reta **G(a,b)** que contem todos os pontos (x,y) correspondentes. Abaixo vemos a interseção dos tubos correspondentes aos pontos $(0,0)$ e $(4,3)$:

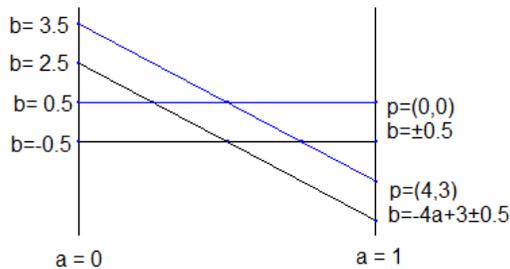


Figura 6

O problema de detecção passa a ser o de encontrar as faces máximas no arranjo dual. Uma face é máxima se é delimitada por cima apenas por segmentos superiores e por baixo apenas por segmentos inferiores. Se uma face é delimitada por cima, por exemplo, por um segmento inferior, significa que a face adjacente por esta aresta tem, pelo menos, um ponto a mais, relativo a aresta comum. A figura a seguir ajuda a entender isto.

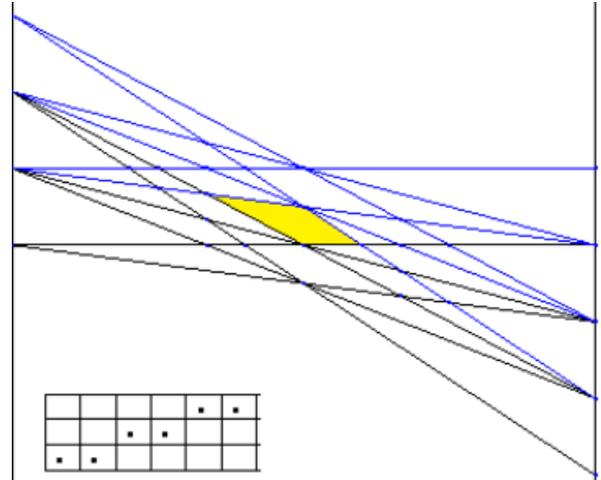


Figura 7

9. Implementação

Asano propõe o “Caminhamento Topológico” para efetuar esta busca. A complexidade de tempo seria de $O(n^2)$ e a de espaço $O(n)$. Porém, como o objetivo do trabalho era experimentar o conceito, foi feita uma implementação utilizando a CGAL. A construção do arranjo, com vimos, gasta $O(n^2)$. No entanto, é utilizada uma estrutura complexa (DCEL) para armazená-lo, que gasta $O(n^2)$ para armazená-lo.

Na implementação, a fase de caminhamento no arranjo gastou $O(n^2 \cdot \log n)$ e também, $O(n^2)$ de espaço.

Lembrando que a complexidade do algoritmo de votação seria $O(nN^4)$, onde N é a dimensão da imagem e n é o número de pontos de borda, e, normalmente, $n \ll N^2$.

A robustez numérica necessária para correto processamento das interseções das retas foi alcançada pela utilização de coordenadas homogêneas, que postergam a divisão.

9. Casos degenerados

Existem dois caso degenerados a considerar:

a) O encontro de várias retas em um único vértice. Isto é tratado corretamente tanto pelo Caminhamento Topológico, quanto pela implementação de arranjos pela CGAL.

b) A sobreposição de um segmento superior de um tubo com o inferior de outro. Isto foi tratado, sem aumento da complexidade, pela introdução de um terceiro tipo de segmento, o duplo. Porém, é necessário que os pontos sejam processados ordenados por x,y . Como os pontos são obtidos de uma matriz, isto não é um problema.

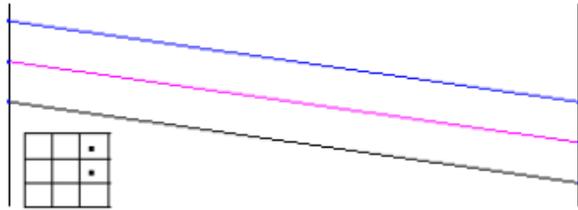


Figura 8

10. Experimentos

Inicialmente, foram feitos testes com a implementação da transformada de Hough da OpenCV, para testar os problemas de detecção. Consegui-se verificar os questões alegadas em relação a definição dos parâmetros ρ , θ e τ . Uma reta que foi detectada com a discretização de 1° , já não foi mais apenas com a mudança de sua interseção com o eixo y (parâmetro b na equação $y=ax+b$). Curiosamente, a reta $y=0.5x+1$ não foi detectada nem com divisores muito altos para θ . Os teste foram feitos em imagens geradas artificialmente de 128×128 .

a	b	Ptos	τ	$\Delta\theta$	Retas	Tempo	a	b
0,46	0	50	49	$\pi/180$	1	4ms	0,445	0
0,46	1	50	49	$\pi/180/8$	1	31ms	0,443	1,09
0,5	0	50	49	$\pi/180/100$	0	690ms		
0,5	0	50	49	$\pi/180/1000$	0	8509ms		
0,46	0	50	49	$\pi/180/8$	3	44ms	0,44-0,46	0 - 2
0	0	50	49	$\pi/180/8$	3	39ms	-0,02	0 - 1
0	0	50	80	$\pi/180/8$	1	33ms	0	0

Tabela 1

Vemos também que aumentando o valor de τ há a detecção de retas espúrias.

Já com a implementação feita, utilizando arranjo de retas, obteve-se sucesso em todos os casos, ainda que com um tempo maior.

Reta gerada			Parâmetro	Detecção			
a	b	Ptos	τ	Retas	Tempo	a	b
0,46	0	50	49	1	75ms	0,447	-0,05
0,46	1	50	49	1	76ms	0,447	0,97
0,5	0	50	49	1	61ms	0,496	0,755
0,46	0	50	49	1	367ms	0,456	-0,03
0	0	50	49	1	115ms	0	0

Tabela 2

Conclusões

Não só foi verificada a melhor qualidade da detecção, em imagens artificiais, do algoritmo proposto em relação a implementação de Hough na OpenCV, como em outros testes todas as retas geradas foram detectadas. Assim, algum problema que necessite de uma detecção mais precisa de retas pode se beneficiar deste algoritmo

Porém, ainda que eficaz, é necessário aumentar a eficiência da implementação. Algumas opções são mostradas em "Trabalhos Futuros".

Referências

- [1] Asano, T., (2002) "Algorithmic Evaluation of Line Detection Problem," Interdisciplinary Information Sciences, Vol. 8, No.2, pp. 137-145, December 2002.
- [2] Asano, T., and Katoh, N., (1996), Variants for the Hough Transform for Line Detection, Computat. Geom. Theory Appl., 6: 231-252.
- [3] Asano, T., and Tokuyama, T., (1998), Topological Walk Revisited, IEICE Trans. Fundam., E81-A: 751-756.
- [4] Asano, T., Guibas, L., and Tokuyama, T., (1994), Walking in an Arrangement Topologically, Int. J. Comput. Geom. Appl., 4: 123-151.
- [5] J.E. Goodman, J. O'Rourke (Eds.), Handbook of Discrete and Computational Geometry (2nd edition), CRC Press, 2004
- [6] Trucco E., Verri A., "Introductory Techniques for 3-D Computer Vision", Prentice Hall, 1998.
- [7] de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O., Computational Geometry: Algorithms and Applications, Springer-Verlag Berlin and Heidelberg GmbH & Co., January 2000. ISBN: 3540656200.
- [8] The Computational Geometry Algorithms Library (CGAL) - <http://www.cgal.org/>
- [9] OpenCV Open Computer Vision Library <http://sourceforge.net/projects/opencvlibrary>