

Transformada Hough

Aplicação da transformada Hough para detecção de linhas e círculos na imagem – Princípios de Visão Computacional – 14/10/2014

Vitor Quaresma Silveira de Hollanda Ramos
110143566

Departamento de Ciência da Computação-CIC
UnB – Universidade de Brasília
Brasília, Brasil
vitor.q.s.h.r@gmail.com

Mateus Mendelson Esteves da Silva
11/0017579

Departamento de Ciência da Computação-CIC
UnB – Universidade de Brasília
Brasília, Brasil
m.mendelson.unb@gmail.com

I. OBJETIVOS

Este projeto tem como objetivo prático criar um programa usando OpenCV que utilize a transformada Hough para detectar linhas e círculos em uma imagem.

O objetivo acadêmico deste projeto é aprender na prática os conceitos teóricos da transformada Hough aprendidos em sala de aula, utilizando a transformada Hough em um programa para resolver um problema prático, a detecção de linhas e círculos.

II. INTRODUÇÃO

A. Transformada Hough

A transformada Hough é um método utilizado para detecção de formas geométricas em uma imagem. A transformada Hough tem como função principal a detecção de linhas em uma imagem, porém pode ser generalizada para detectar outras formas geométricas. É um método robusto onde não é necessário conhecer a posição das linhas para detectá-las. A transformada Hough considera que cada ponto (X_i, Y_i) na imagem possui infinitas retas que obedecem a equação geral da reta, com a e b variáveis.

$$y_i = ax_i + b$$

Fig. 1. Equação geral da reta

Podemos escrever a equação de outra forma:

$$b = -x_i a + y_i$$

Fig. 2. Equação geral da reta reescrita

A partir desta segunda equação definimos um plano ab com inclinação $-x_i$ e intersecção y_i .

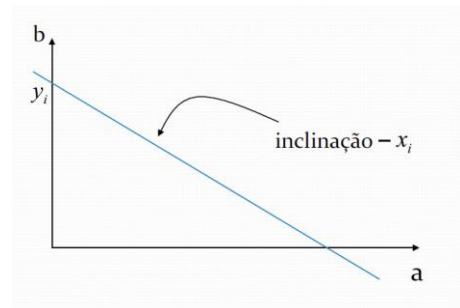


Fig. 3. Plano ab

Definimos então um outro ponto no plano xy de coordenadas (x_j, y_j) . Podemos definir então a equação da reta para este ponto e a equação da reta reescrita. A reta reescrita no plano ab intersecciona a reta do ponto (x_i, y_i) no ponto (a', b') , sendo a' e b' a inclinação e intersecção da reta da equação reescrita de (x_j, y_j) . Logo, todos os pontos pertencentes a uma mesma reta no plano xy tem intersecção no plano ab , no ponto (a', b')

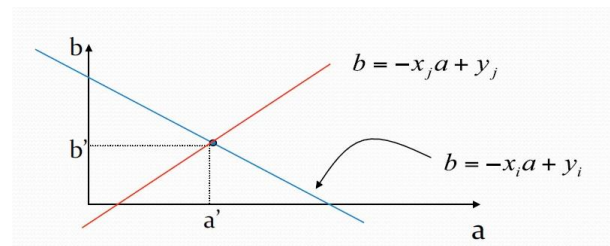


Fig. 4. Plano ab com as duas retas se interseccionando

B. Implementação

Dividimos o espaço de parâmetros em células acumuladoras, onde cada célula (i, j) possui um acumulador $A(i, j)$ que guarda o número de ocorrências de a_i, b_j , devemos definir também (a_{\min}, a_{\max}) e (b_{\min}, b_{\max}) como os valores mínimos e máximos permitidos para a inclinação e intersecção das retas. Para cada ponto x_k, y_k no plano xy ,

devemos considerar o parâmetro a como todos os valores possíveis de a e calcular o parâmetro b na equação:

$$b = -x_k a + y_k$$

Fig. 5. Equação da reta usada na implementação

Então, se um valor de a_i resulta em b_j devemos somar 1 ao acumulador $A(i,j)$. Ao final do algoritmo, cada acumulador $A(i,j)$ vai ter a quantidade de pontos em uma reta com parâmetro a_i , b_j .

Quando uma reta tende a se tornar vertical, os valores de a e b tendem a infinito, isto porque tendemos a ter infinitos valores de y para um mesmo valor de x . Para contornar este problema, podemos utilizar a equação da reta em coordenadas polares, da seguinte forma:

$$x \cos \theta + y \sin \theta = \rho$$

Fig. 6. Equação da reta com coordenadas polares

Utilizando esta representação teremos um espaço $\rho\theta$ no lugar do espaço ab . E são obtidas curvas senoidais neste espaço, ao invés de retas.

C. Generalização

Podemos generalizar a transformada de Hough e aplicá-la a qualquer função da forma $g(v,c) = 0$, onde v é um vetor de coordenadas e c , de coeficientes. Neste projeto, foi feita a detecção de círculos, círculos possuem uma função da forma

$$(x - c_1)^2 + (y - c_2)^2 = c_3^2$$

Fig. 7. Equação do círculo

Para um círculo de raio c_3 e centro em (c_1, c_2) . Dessa forma, temos um plano $c_1c_2c_3$ para detecção de retas, ao invés do plano ab visto anteriormente.

III. MATERIAIS E MÉTODOS

Neste projeto foram implementadas duas classes principais, a classe *ImageGetter* é responsável por pegar a imagem da fonte escolhida pelo usuário, a classe *HoughDetector* é responsável pela detecção de linhas e círculos utilizando a transformada de Hough na imagem pega pela classe *ImageGetter*. O programa foi compilado por meio do script `build_all.sh`, e deve ser executado sem passar nenhum parâmetro por linha de comando.

A. ImageGetter

A classe *ImageGetter* é a classe responsável por obter do usuário a opção de onde este deseja pegar a imagem e pegar a imagem a partir desta opção. Primeiramente esta classe mostra

um menu ao usuário mostrando as opções de entrada, podendo então o usuário escolher entre um arquivo de imagem, um arquivo de vídeo ou a câmera. Se o usuário optar por um arquivo de imagem, o programa irá pedir o nome do arquivo a ser aberto, o usuário deve digitar o nome do arquivo com a extensão, e o programa irá então abrir este arquivo e salvar a imagem contida neste arquivo em uma *IpImage*. Se o usuário optar por utilizar um arquivo de vídeo, o programa irá pedir o nome do arquivo de vídeo a ser utilizado, o usuário deve então digitar o nome do arquivo de vídeo com extensão, o programa irá reproduzir o vídeo para o usuário até que este aperte a tecla *Esc*, quando o usuário apertar o *Esc*, o programa irá guardar a imagem do frame atual do vídeo que está sendo reproduzido e guardar em uma *IpImage*. Se o usuário optar pela opção de pegar a imagem pela câmera, o programa irá abrir a primeira câmera que encontrar, mostrando a imagem capturada por esta para o usuário até o usuário apertar *Esc*, quando o usuário aperta *Esc*, o programa pega o último frame de imagem capturado pela câmera e guarda em uma *IpImage*.

B. HoughDetector

A classe *HoughDetector* é a classe principal chamada pela *main*, primeiramente a *HoughDetector* cria e chama a classe *ImageGetter* para obter a imagem da fonte que o usuário optou. Depois a classe cria um *Mat*, passando a *IpImage* como parâmetro, esta *Mat* é criada para ser utilizada nas funções das transformadas de Hough. A classe então prepara e executa as transformadas de Hough para detecção de retas e círculos, executando primeiramente a transformada para detecção de retas e depois a transformada para detecção de círculos.

O programa então desenha os círculos e as linhas e mostra para o usuário três imagens diferentes, a original, a com a detecção de retas e com a detecção de círculos. Ao apertar qualquer tecla, o programa finaliza.

Esta classe, ainda, apresenta o método *houghCirclesTest*. Este método permite que o usuário avalie a quantidade de círculos que são detectados ao variar os parâmetros de detecção de círculo. A escolha de quais valores serão utilizados deve ser feita escolhendo os parâmetros que proporcionam o melhor resultado.

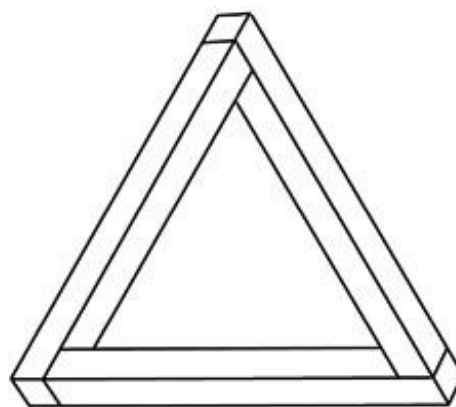


Fig. 8. Imagem utilizada no experimento.

Para a realização dos testes, foram utilizadas duas imagens, sendo elas Fig. 8 e Fig.9.

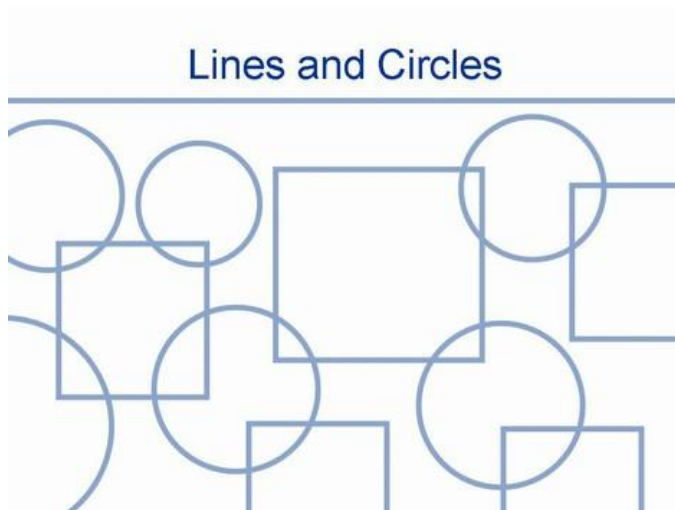


Fig. 9. Imagem utilizada no experimento.

A realização do experimento se deu em duas etapas. Na primeira, as duas imagens acima foram carregadas pelo programa a partir do HD. Na segunda, as imagens foram impressas, com o uso de uma impressora laser, e colocadas em frente à câmera. Então, a imagem da câmera foi capturada e analisada. Para cada imagem, a análise foi feita 10 vezes e os resultados aqui apresentados são a média dos valores obtidos.

IV. RESULTADOS

Primeiramente, as imagens foram lidas a partir do HD. Nesta etapa, foram utilizados os valores $param_1 = 100$ e $param_2 = 44$. Depois, as imagens foram impressas e capturadas por meio da câmera acoplada ao computador. Neste caso, para a segunda figura, foi necessário alterar os parâmetros para os valores $param_1 = 116$ e $param_2 = 10$.

A. Primeira imagem (Fig. 8)

1) Imagem do HD: foram detectados 22 linhas e 0 círculos (Fig. 10).

2) Imagem da câmera: foram detectados 27 linhas e 0 círculos (Fig. 11).

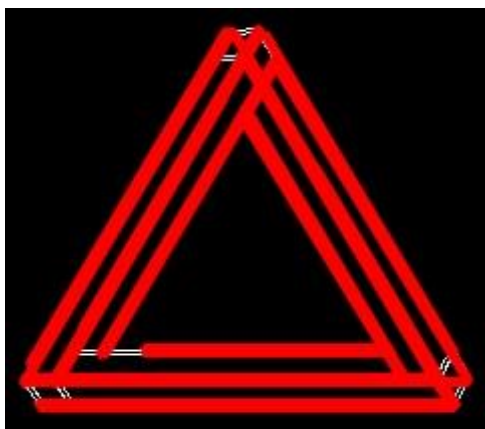


Fig. 10. Linhas detectadas pelo algoritmo de Hough.

B. Segunda imagem (Fig.9)

1) Imagem do HD: foram detectados 50 linhas (Fig. 12) e 6 círculos (Fig. 13).

2) Imagem da câmera: foram detectados 48,7 linhas e 6,2 círculos (Fig. 14).

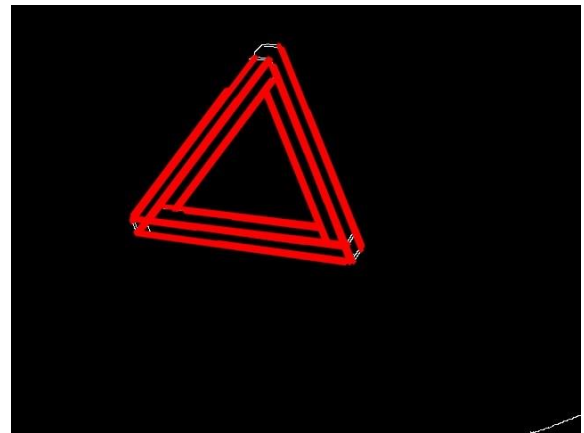


Fig. 11. Linhas detectadas a partir de imagem capturada pela câmera.

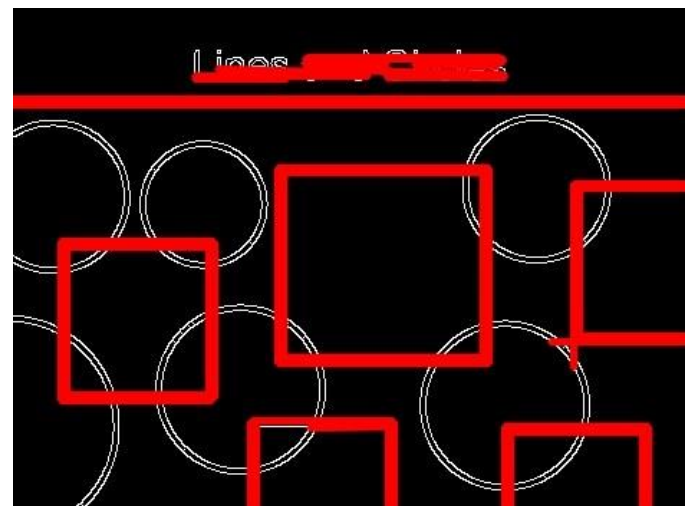


Fig. 12. Linhas detectadas pelo algoritmo de Hough.

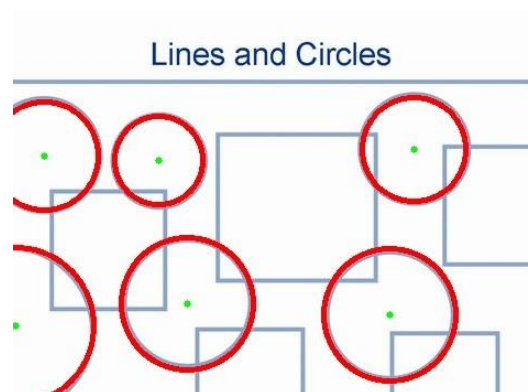


Fig. 13. Círculos detectados com o algoritmo de Hough.

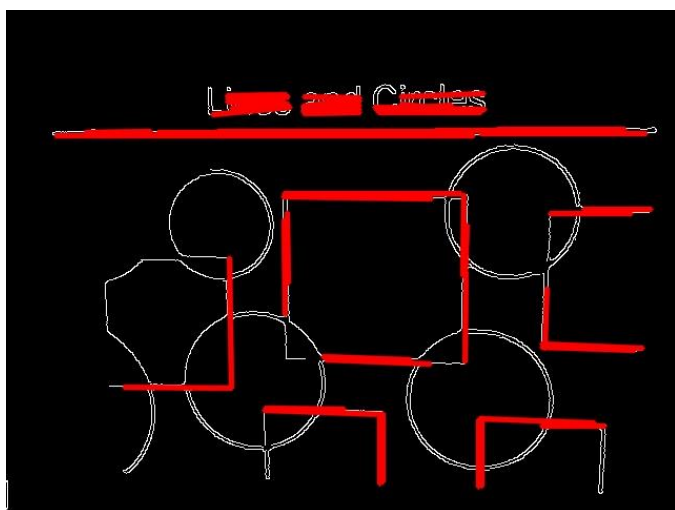


Fig. 14. Linhas detectadas a partir da imagem capturada pela câmera.

V. DISCUSSÃO E CONCLUSÕES

Os resultados obtidos das imagens carregadas diretamente do HD são satisfatórios e estão de acordo com o esperado.

Para as imagens capturadas pela câmera, os resultados obtidos foram discrepantes. Isso se dá, principalmente, por conta da resolução da câmera. Se a câmera não for capaz de captar todos os pontos, o programa não os terá para fazer a análise correta. Entretanto, mesmo com essas discrepâncias, os resultados obtidos não estão muito distantes dos resultados reais.

O uso da Transformada de Hough mostrou ser uma boa solução para o problema da detecção de retas e círculos, uma vez que seus parâmetros estejam devidamente ajustados.

BIBLIOGRAFIA

- [1] Aula_17 – PVC , Prof. Dr. Flávio Vidal
- [2] Open-CV Hough Circles documentation
http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_circle/hough_circle.html
- [3] Open-CV Hough Lines documentation
http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html