

# Canny Edge Detection and Hough Transform

## Report

### 1.Canny Edge Detection

Phương pháp này do John Canny ở phòng thí nghiệm MIT khởi xướng vào năm 1986. Canny đã đưa ra một tập hợp các ràng buộc mà một phương pháp phát hiện biên phải đạt được. Ông đã trình bày một phương pháp tối ưu nhất để thực hiện được các ràng buộc đó. Và phương pháp này được gọi là phương pháp Canny. Bây giờ chúng ta sẽ tìm hiểu hoạt động và triển khai của thuật toán này

Các bước cơ bản liên quan đến thuật toán này:

- Giảm nhiễu bằng bộ lọc Gaussian
- Tính toán độ dốc dọc theo trục ngang và trục dọc
- Non-Maximum Suppresion
- Định ngưỡng độ trễ

#### Giảm nhiễu :

Bước này là cực kỳ quan trọng trong canny edge detection. Nó sử dụng bộ lọc Gaussian để loại bỏ nhiễu khỏi hình ảnh, đó là bởi vì nhiễu này có thể được coi là các cạnh do sự thay đổi cường độ đột ngột của máy dò cạnh. Tổng các phần tử trong hạt nhân Gaussian là 1, do đó hạt nhân phải được chuẩn hóa trước khi áp dụng dưới dạng tích chập cho hình ảnh.

$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Trong đoạn code này, chúng ta sử dụng một nhân có kích thước 5 X 5 và sigma=1.4, sẽ làm mờ hình ảnh và loại bỏ nhiễu khỏi nó.

```
# noise reduction step  
img = cv2.GaussianBlur(img, (5, 5), 1.4)
```

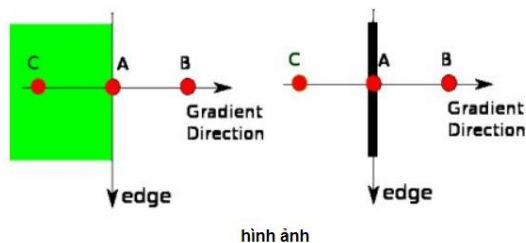
**Tính gradient:** Khi hình ảnh được làm mịn, các đạo hàm  $I_x$  và  $I_y$  được tính theo trục  $x$  và  $y$ . Nó có thể được thực hiện bằng cách sử dụng tích chập các hạt nhân Sobel-Feldman với hình ảnh như đã cho:

$$K_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

Hạt nhân Sobel

Sử dụng 2 hạt nhân: 1 hạt nhân để thay đổi theo chiều ngang và một hạt nhân để thay đổi theo chiều dọc.

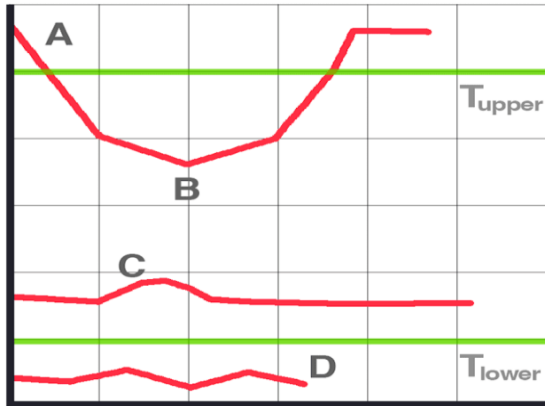
**Non-Maximum Suppression:**



Điểm A nằm trên cạnh (theo phương thẳng đứng). Gradient hướng bình thường đối với cạnh. Điểm B và C theo hướng gradient. Vì vậy, điểm A được kiểm tra với điểm B và C để xem nó có tạo thành cực đại cục bộ hay không. Nếu là cực đại cục bộ nó được xem xét cho giai đoạn tiếp theo, nếu không nó sẽ bị loại bỏ (đưa về 0).

Tóm lại, kết quả ta nhận được là một ảnh nhị phân với “các cạnh mỏng”.

**Định ngưỡng độ trễ:** Ta sẽ xét các pixel dương trên mặt nạ nhị phân kết quả của bước trước. Nếu giá trị gradient vượt ngưỡng  $max\_val$  thì pixel đó chắc chắn là cạnh. Các pixel có độ lớn gradient nhỏ hơn ngưỡng  $min\_val$  sẽ bị loại bỏ. Còn các pixel nằm trong khoảng 2 ngưỡng trên sẽ được xem xét rằng nó có nằm liên kề với những pixel được cho là "chắc chắn là cạnh" hay không. Nếu liên kề thì ta giữ, còn không liên kề bất cứ pixel cạnh nào thì ta loại. Sau bước này ta có thể áp dụng thêm bước hậu xử lý loại bỏ nhiễu (tức những pixel cạnh rời rạc hay cạnh ngắn) nếu muốn. Ảnh minh họa về ngưỡng lọc



- Ở trên cùng của biểu đồ, chúng ta có thể thấy rằng **A** là một cạnh chắc chắn, vì  $A > T_{upper}$ .
- **B** cũng là một cạnh, mặc dù  $B < T_{upper}$  kể từ khi nó được kết nối với một lợi thế cạnh mạnh.
- **C** không phải là một cạnh kể từ  $C < T_{upper}$  và không được kết nối với một cạnh mạnh.
- Cuối cùng, **D** không phải là một cạnh vì  $D < T_{lower}$  và tự động bị loại bỏ.

## 2.Hough Transform

Hough Transform là một phương pháp được sử dụng trong xử lý hình ảnh để phát hiện bất kỳ hình dạng nào, nếu hình dạng đó có thể được biểu diễn dưới dạng toán học. Nó có thể phát hiện hình dạng ngay cả khi nó bị hỏng hoặc bị bóp méo một chút.

### 2.1 Abstraction

Ý tưởng chung của việc phát hiện đường thẳng trong thuật toán này là tạo mapping từ không gian ảnh (A) sang một không gian mới (B) mà mỗi đường thẳng trong không gian (A) sẽ ứng với một điểm trong không gian (B).

### 2.2. Phương trình đường thẳng trong không gian ảnh (A)

$$y = ax + b$$

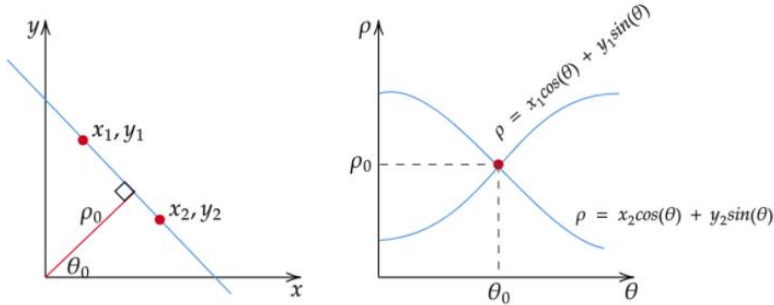
Giá trị của tọa độ gốc  $a$  từ âm vô cùng tới dương vô cùng.

Ex:  $x=0$  thì  $a$  tiến đến  $\infty$

Thuật toán Hough Transform yêu cầu các giá trị  $a, b$  nằm trong một khoảng xác định (hay bị chặn trên dưới), ta phải sử dụng hệ tọa độ cực để biểu diễn phương trình đường thẳng. Cách biểu diễn này cũng nằm trong chương trình toán trung học:

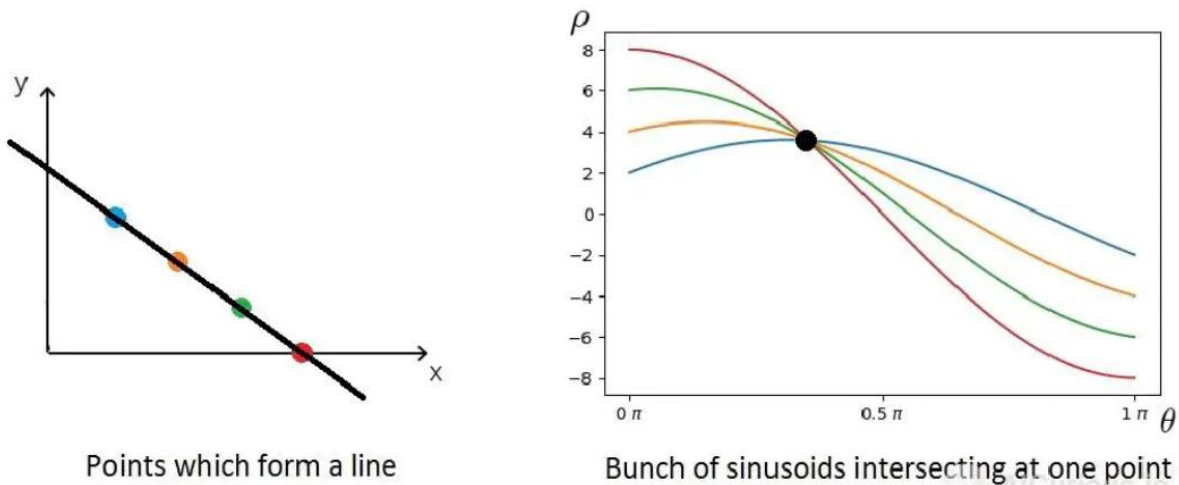
$$\rho = x \cos(\theta) + y \sin(\theta).$$

Từ một đường thẳng trong không gian ảnh (A) với 2 tham số  $a$  và  $b$ , chúng ta sẽ map sang không gian Hough (B) thành một điểm.



## 2.2. Mapping giữa không gian ảnh (A) và không gian Hough (B)

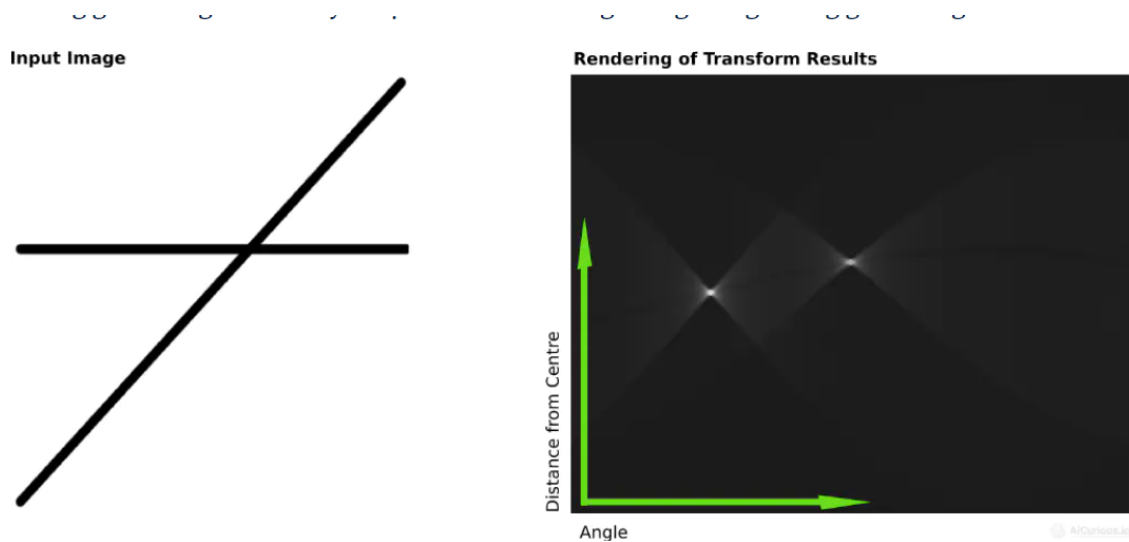
Từ một điểm trong không gian ảnh, chúng ta lại có được một hình sin trong không gian Hough:



Các điểm nằm trên cùng một đường thẳng lại có biểu diễn là các hình sin giao nhau tại một điểm trong không gian Hough. Đây là nơi xuất phát ý tưởng của thuật toán Hough Transform. Chúng ta sẽ dựa vào các điểm giao nhau này để suy ngược lại phương trình đường thẳng trong không gian ảnh.

### Mapping nhiều điểm thẳng hàng từ không gian ảnh sang không gian Hough

Mỗi đường thẳng khác nhau sẽ tạo thành một điểm sáng (nơi giao nhau của nhiều hình sin) trên không gian Hough. Dưới đây là sự biểu diễn 2 đường thẳng trong không gian Hough.



### 3.Line detection

#### Thuật toán

- Hough Transform yêu cầu đầu vào là một ảnh nhị phân. Trên thực tế ảnh sẽ được đưa về dạng ảnh xám, áp dụng các thuật toán lọc biên để xác định các đường biên trong ảnh. Ở đây chúng ta sẽ sử dụng thuật toán Canny để lọc biên.
- Đầu tiên, nó tạo ra một mảng hoặc bộ tích lũy 2D (để giữ các giá trị của hai tham số) và ban đầu nó được đặt thành 0.
- Để các hàng biểu thị  $r$  và các cột biểu thị ( $\theta$ ) theta.
- Kích thước của mảng phụ thuộc vào độ chính xác bạn cần. Giả sử bạn muốn độ chính xác của các góc là 1 độ, bạn cần 180 cột (Độ tối đa cho một đường thẳng là 180).
- Đối với  $r$ , khoảng cách lớn nhất có thể là độ dài đường chéo của hình ảnh. Vì vậy, lấy độ chính xác một pixel, số hàng có thể là chiều dài đường chéo của hình ảnh.

#### Result:

[https://colab.research.google.com/drive/1D\\_pVU4qbus8JnM8HANLVmbAi5ws1GhFB#scrollTo=sdpwLDWUtZFS](https://colab.research.google.com/drive/1D_pVU4qbus8JnM8HANLVmbAi5ws1GhFB#scrollTo=sdpwLDWUtZFS)

