

TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN XỬ LÝ ẢNH

**Nhận diện biển số xe bằng thuật toán Otsu
và sử dụng hệ số tương quan để nhận diện
ký tự**

Lớp 64CS3 – Nhóm 04		
Thành viên nhóm	Nguyễn Xuân Cường	15507764
	Dương Gia Khánh	1526864
	Vương Trung Thành	186864
	Ngô Đức Thịnh (TN)	189464
Giảng viên hướng dẫn	ThS. Thái Thị Nguyệt	

Lời cảm ơn

Nhóm 04 xin gửi lời cảm ơn đến cô giáo, **ThS. Thái Thị Nguyệt** – giảng viên bộ môn Khoa học máy tính - khoa Công nghệ thông tin đã hướng dẫn chúng em trong suốt quá trình học hỏi và hướng dẫn làm đồ án. Cô đã dành nhiều thời gian của mình để tận tình chỉ bảo, hướng dẫn, định hướng cho nhóm thực hiện các bài thực hành, giúp cho chúng em hiểu biết thêm về môn học cũng như học hỏi thêm kinh nghiệm và kỹ năng để hoàn thành bài tập.

Mặc dù đã cố gắng hoàn thành tốt công việc được phân công trong phạm vi và khả năng cho phép song trong quá trình làm đồ án do kiến thức chuyên ngành của chúng em còn hạn chế nên không thể tránh khỏi một vài sai sót nhất định khi trình bày và đánh giá vấn đề. Rất mong nhận được sự góp ý, đánh giá của cô để đồ án của chúng em hoàn thiện hơn, đồng thời bổ sung vốn kinh nghiệm của chúng em trên con đường sắp tới.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU	5
1.1 Thực trạng	5
1.2 Mô tả bài toán.....	5
1.3 Quy trình nhận diện biển số xe	5
CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT.....	6
2.1 Xử lý ảnh và các vấn đề cơ bản trong xử lý ảnh.....	6
2.1.1 Xử lý ảnh là gì ?.....	6
2.1.2 Định nghĩa về xử lý ảnh.....	6
2.1.3 Các bước chính trong xử lý ảnh số	7
2.1.4 Ứng dụng của xử lý ảnh.....	8
2.2 Phân vùng ảnh theo ngưỡng.....	9
2.2.1 Phân đoạn ảnh dựa vào kỹ thuật cắt ngưỡng	9
2.3 Phân vùng ảnh theo biên	13
2.3.1 Khái niệm cơ bản	13
2.3.2 Khái quát về biên và phân loại các kỹ thuật dò biên	13
2.4 Lược đồ xám Histogram	26
CHƯƠNG 3. CÁC PHƯƠNG PHÁP XỬ LÝ ẢNH TRONG NHẬN DIỆN BIỂN SỐ XE	27
3.1 Giảm nhiễu	27
3.1.1 Bộ lọc Gauss	27
3.1.2 Bộ lọc Bilateral	28
3.2 Xử lý nhị phân.....	29
3.2.1 Ảnh nhị phân.....	29
3.2.2 Thuật toán Otsu	29
3.2.3 Thuật toán Canny	31
3.3 Lọc biên với Contour	32
3.4 Mô hình OCR.....	Lỗi! Thẻ đánh dấu không được xác định.
3.4.1 Giới thiệu	Lỗi! Thẻ đánh dấu không được xác định.
3.4.2 Cơ chế OCR	Lỗi! Thẻ đánh dấu không được xác định.
3.4.3 Mã nguồn mở OCR. Lỗi! Thẻ đánh dấu không được xác định.	
CHƯƠNG 4. THỰC NGHIỆM	40
4.1 Khai báo các thư viện trong python	40
4.2 Đọc file ảnh và chuyển xám ảnh	40

4.3	Dùng thuật toán chọn ngưỡng hoặc nhận diện biên xử lý ảnh.....	41
4.4	Tìm contour và ghép vào ma trận zero	42
4.5	Áp dụng easyOCR để đi nhận diện chữ trong ảnh biển số xe.	43
CHƯƠNG 5. GIAO DIỆN TKINTER		45
5.1	RGB sang thang độ xám	45
5.2	Làm mịn hình ảnh – Không tuyến tính (Bộ lọc trung vị có trọng số 3x3) 45	
5.3	Phát hiện cạnh Canny	46
5.4	Tách biển số xe.....	46
5.5	Nhận dạng ký tự biển số xe.....	46
TÀI LIỆU THAM KHẢO		47

CHƯƠNG 1. GIỚI THIỆU

1.1 Thực trạng

Hiện nay, số lượng xe cộ tham gia giao thông trên đường là rất lớn dẫn đến tiêu tốn rất nhiều nhân lực và vật lực cho việc quản lý phương tiện cá nhân trong bãi gửi xe. Nếu không có một công cụ thuận tiện thì việc quản lý phương tiện cá nhân rất mất thời gian, dễ gây nhầm lẫn, thiệt hại cho người sử dụng dịch vụ tại các bãi đỗ xe.

Để giảm tải cho các công việc như thu tiền, bảo hiểm xe, tìm xe cộ trong bãi đỗ xe, trên thế giới đã phát triển công nghệ giám sát tự động đối với các phương tiện giao thông, chính nhờ tính cá nhân của biển số xe mà nó đã trở thành đối tượng chính được sử dụng để nghiên cứu, phát triển trong công nghệ này.

Do đó nhóm 04 muốn chọn đề tài này như bước căn bản trong việc tìm hiểu các công cụ giám sát mạnh hơn như kiểm soát xe lưu thông trên đường hay nhận dạng khuôn mặt ... đang được thế giới rất chú trọng lúc này.

1.2 Mô tả bài toán

Hệ thống nhận dạng biển số xe là hệ thống có khả năng phân tích hình ảnh và xác định biển số trên xe, thông qua video, thiết bị ghi hình và hình ảnh.

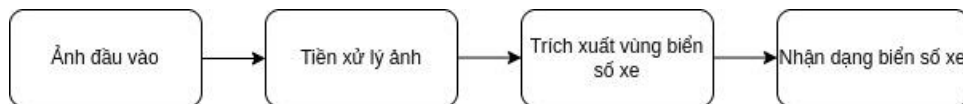
Có nhiều cách thức khác nhau để phân loại ứng dụng nhận diện biển số xe. Trong đó cách phân loại dựa trên mục đích sử dụng là thông dụng nhất và có thể chia ứng dụng nhận dạng biển số xe thành hai loại sau:

- Loại 1: Giới hạn vùng nhìn
- Loại 2: Không giới hạn vùng nhìn

Để giải quyết được bài toán nhận dạng biển số xe, yêu cầu đặt ra là phải giải quyết được 3 bài toán con:

- Bài toán thứ nhất: xác định vùng chứa biển số xe và tách biển số.
- Bài toán thứ hai: phân đoạn từng ký tự.
- Bài toán thứ ba: nhận dạng ký tự.

1.3 Quy trình nhận diện biển số xe



Ảnh đầu vào: sẽ là các hình ảnh chứa các biển số xe theo tiêu chí hình chữ nhật.

Tiền xử lý ảnh: thực hiện nhiệm vụ chuyển đổi không gian màu, khử nhiễu, chỉnh mức xám, cân bằng ánh sáng... nhằm mục đích thu được dữ liệu theo yêu cầu của bài toán tốt nhất.

Trích xuất vùng biển số xe: Áp dụng các thuật toán phân vùng theo ngưỡng hay nhận diện biên để đi xác định vùng chứa biển số xe và tách ra khỏi nền.

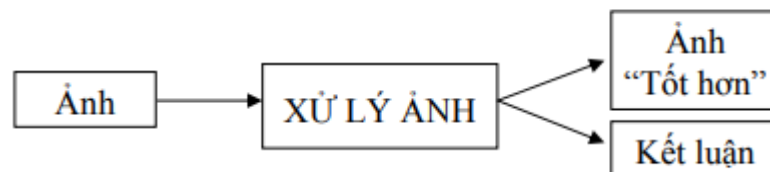
Nhận dạng biển số xe: Bước này ta sẽ đi nhận diện từng ký tự của biển và nhận diện chúng đưa ra kết quả dự đoán.

CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT

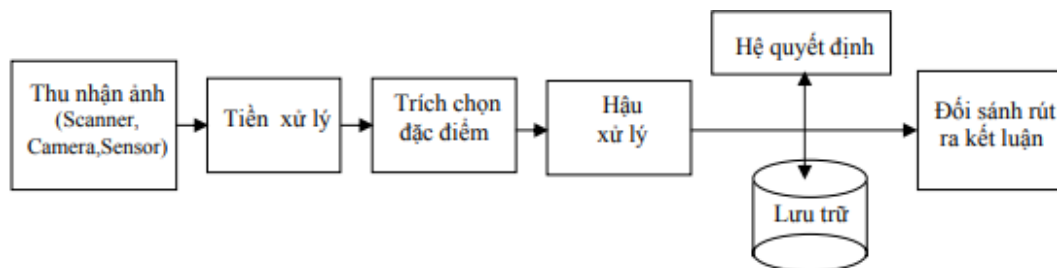
2.1 Xử lý ảnh và các vấn đề cơ bản trong xử lý ảnh

2.1.1 Xử lý ảnh là gì ?

Con người thu nhận thông tin qua các giác quan, trong đó thị giác đóng vai trò quan trọng nhất. Những năm trở lại đây với sự phát triển của phần cứng máy tính, xử lý ảnh và đồ họa đã phát triển một cách mạnh mẽ và có nhiều ứng dụng trong cuộc sống. Xử lý ảnh và đồ họa đóng một vai trò quan trọng trong tương tác người máy. Quá trình xử lý ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.



Ảnh có thể xem là tập hợp các điểm ảnh và mỗi điểm ảnh được xem như là đặc trưng cường độ sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian và nó có thể xem như một hàm n biến $P(c_1, c_2, \dots, c_n)$. Do đó, ảnh trong xử lý ảnh có thể xem như ảnh n chiều. Sơ đồ tổng quát của một hệ thống xử lý ảnh:



2.1.2 Định nghĩa về xử lý ảnh

Xử lý ảnh (XLA) là đối tượng nghiên cứu của lĩnh vực thị giác máy, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới với các đặc tính và tuân theo ý muốn của người sử dụng. Xử lý ảnh có thể gồm quá trình phân tích, phân lớp các đối tượng, làm tăng chất lượng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các thông tin hình ảnh của ảnh.

Cũng như xử lý dữ liệu bằng đồ họa, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo bởi các chương trình. Xử lý ảnh số bao gồm các phương pháp và kỹ thuật biến đổi, để truyền tải hoặc mã hoá các ảnh tự nhiên. Mục đích của xử lý ảnh gồm:

- Biến đổi ảnh làm tăng chất lượng ảnh
- Tự động nhận dạng ảnh, đoán nhận ảnh, đánh giá các nội dung của ảnh.

Nhận biết và đánh giá các nội dung của ảnh là sự phân tích một hình ảnh thành những phần có ý nghĩa để phân biệt đối tượng này với đối tượng khác, dựa vào đó ta có thể mô tả cấu trúc của hình ảnh ban đầu. Có thể liệt kê một số phương pháp nhận dạng cơ bản như nhận dạng ảnh của các đối tượng trên ảnh, tách cạnh, phân đoạn hình ảnh,... Kỹ thuật này được dùng nhiều trong y học (xử lý tế bào, nhiễm sắc thể), nhận dạng chữ trong văn bản.

2.1.3 Các bước chính trong xử lý ảnh số

2.1.3.1. Thu nhận ảnh

- Biến đổi thông tin hình ảnh về các cấu trúc được lưu trữ trong máy tính, có thể hiển thị qua các thiết bị ngoại vi như máy in, màn hình,...
- Ảnh được thu từ nhiều người khác nhau : Camera chụp ảnh, video, máy scan,...

2.1.3.2. Tiền xử lý ảnh

- Là quá trình sử dụng các kỹ thuật xử lý ảnh để làm ảnh tốt lên theo mục đích sử dụng
- Mục đích : Lọc nhiễu, nâng cao độ tương phản

2.1.3.3. Phân đoạn ảnh

- Là quá trình phân chia nội dung các đối tượng cần khảo sát ra khỏi ảnh.
- Phân chia các đối tượng tiếp giáp nhau.
- Phân tách các đối tượng riêng biệt thành các đối tượng con.

2.1.3.4. Biểu diễn và mô tả

- Tìm các vùng đặc trưng của điểm ảnh và biểu diễn lại thông qua các điểm đặc trưng

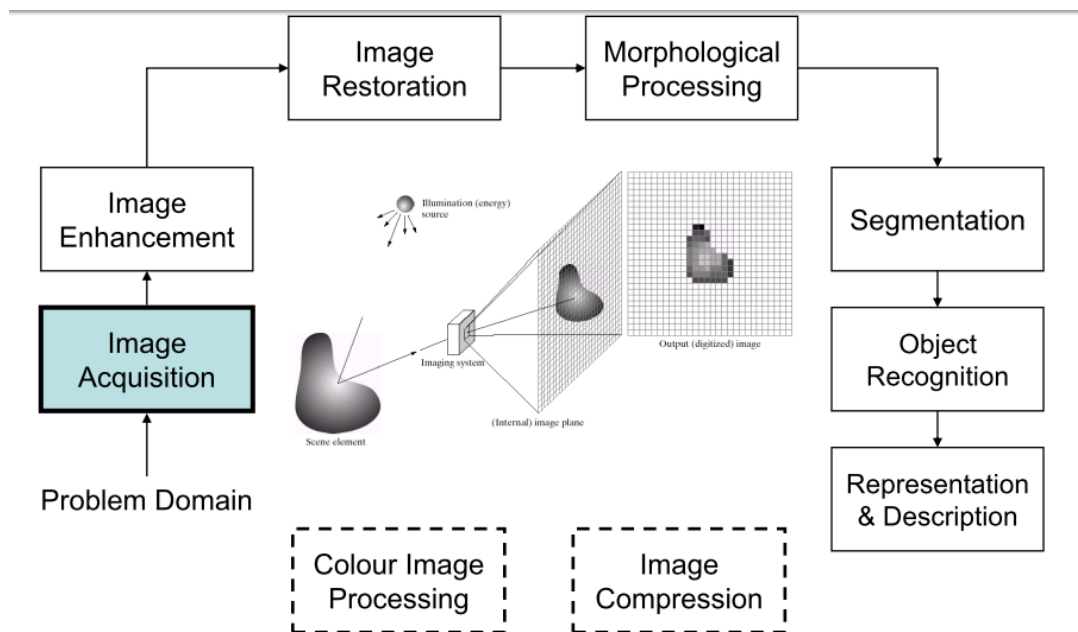
2.1.3.5. Nhận dạng và nội suy

- Nhận dạng là quá trình xác định ảnh bằng cách so sánh với mẫu chuẩn đã được lưu trước.
- Nội suy là phán đoán dựa trên cơ sở nhận dạng

2.1.3.6. Mô hình toán học

Theo lý thuyết về nhận dạng, các mô hình toán học về ảnh được phân theo hai loại nhận dạng ảnh cơ bản:

- Nhận dạng theo tham số
- Nhận dạng theo cấu trúc



2.1.4 Ứng dụng của xử lý ảnh

- Khôi phục hình ảnh, chỉnh sửa, điều chỉnh độ phân giải: Ứng dụng này tương tự như photoshop: từ một hình ảnh được chụp từ máy ảnh, ta có thể chỉnh sửa, xử lý để làm ảnh đẹp hơn hoặc phù hợp nhu cầu người dùng như: làm mờ, lấy biên, chỉnh độ nét, chỉnh độ phân giải, phục hồi và nhận dạng ảnh....
- Trong y học: Các thuật toán xử lý ảnh cho phép biến đổi hình ảnh được tạo ra từ nguồn bức xạ X-ray hay nguồn bức xạ siêu âm thành hình ảnh quang học trên bề mặt film X-quang hoặc trực tiếp trên bề mặt màn hình hiển thị.
- UV imaging: Lĩnh vực này liên quan nhiều đến thám hiểm, do thám. Cách hoạt động như sau: để phân tích thiệt hại của một trận động đất mà con người không thể tới được. Mặt đất nơi đó sẽ được quét bởi vệ tinh hoặc một máy bay sau đó truyền dữ liệu, hình ảnh về máy chủ để phân tích. Sẽ rất nhanh chóng so với việc chờ đợi con người tới đó. Một trận động đất có thể diện tích rất rộng mà con người không thể nào phân tích hết được.
- Truyền và mã hóa: Lĩnh vực này không cần giải thích nhiều chắc các bạn cũng hiểu được. Mình sẽ cho một ví dụ. Ngày nay con người sử dụng internet để truyền nhận các ảnh, video một cách nhanh chóng. Hình ảnh khi ta chụp sẽ được mã hóa và truyền theo internet. Rất nhanh sau vài giây là người bạn có thể nhận được một bức ảnh.
- Thị giác máy tính và robot: Hiện tại công nghệ robot đang phát triển nhanh chóng, và càng ngày càng giống con người hơn. Thị giác của máy tính cũng là một phần quan trọng. Làm thế nào để robot có thể nhìn mọi thứ, tránh vật cản, nhận dạng các vật..? Đó chính là nhờ một hệ thống quá trình xử lý ảnh phức tạp.
- Phát hiện vật cản: Phát hiện vật cản cũng là một lĩnh vực mới và được thực hiện bởi xử lý ảnh: tính toán khoảng cách từ robot tới vật cản bằng

cách xác định được các đối tượng khác nhau trong hình ảnh sau đó xử lý và tính toán chúng.

- Công nghệ nhận dạng: Xử lý ảnh dùng để xác định, nhận dạng các đối tượng, các mối nguy hiểm, nhận dạng vân tay, khuôn mặt, hoặc các loại bệnh trong lĩnh vực y tế.

2.2 Phân vùng ảnh theo ngưỡng

2.2.1 Phân đoạn ảnh dựa vào kỹ thuật cắt ngưỡng

2.2.1.1. Khái niệm

Ảnh đa cấp xám (ảnh xám) là ảnh mà các pixel có giá trị xám từ 0 đến 255 (thông thường). Ảnh màu (RGB) thì mỗi pixel có 3 thành phần đại diện cho 3 màu (Red, Green, Blue). Trong đó: giá trị Red, Green, Blue từ 0 đến 255. Cắt ngưỡng là kỹ thuật chuyển giá trị mức xám của ảnh (pixel) về một giá trị xám chung nào đó. Ví dụ: cắt ngưỡng nhị phân là chuyển ảnh về 2 giá trị: 0 hoặc 1

Các loại phân đoạn ảnh dựa trên kỹ thuật cắt ngưỡng:

- Cắt ngưỡng toàn cục (giá trị ngưỡng đơn lẻ và không đổi)
- Cắt ngưỡng đa ngưỡng (nhiều giá trị ngưỡng)
- Cắt ngưỡng thích nghi (giá trị ngưỡng thay đổi và phụ thuộc giá trị mức xám của pixel cắt ngưỡng và các pixel lân cận)

2.2.1.2. Phân đoạn ảnh dựa vào kỹ thuật cắt ngưỡng toàn cục

a. Cắt ngưỡng toàn cục (đơn ngưỡng)

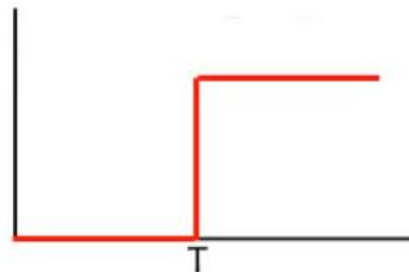
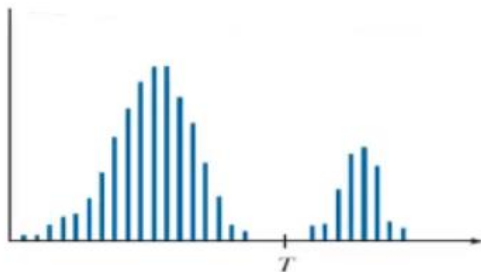
Cắt ngưỡng dựa vào 1 giá trị ngưỡng T cho toàn bộ ảnh

$$g(x, y) = \begin{cases} 1 & \text{nếu } f(x, y) > T \\ 0 & \text{nếu } f(x, y) \leq T \end{cases}$$

Chú thích: $f(x, y)$: biểu diễn ảnh đầu vào

$g(x, y)$: biểu diễn đã được cắt ngưỡng

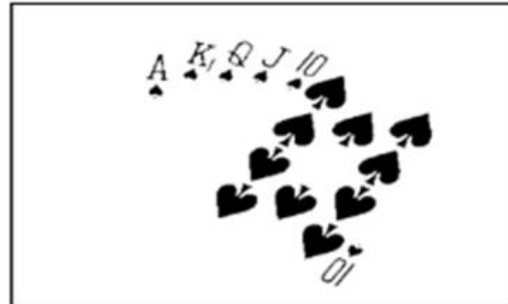
Ví dụ ta sẽ lược đồ xám $f(x, y)$



Phân chia ảnh vào 2 vùng: Vùng có mức xám = 1 và vùng có mức xám = 0



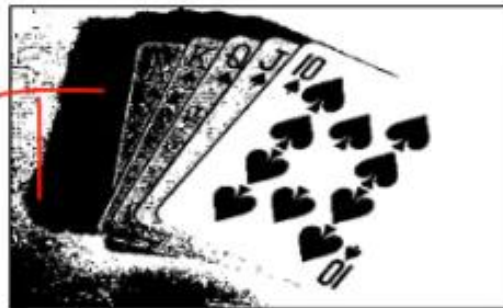
Ảnh gốc



Ảnh sau khi
cắt ngưỡng

Lưu ý: Chất lượng của ảnh cắt ngưỡng là phụ thuộc vào T

Ví dụ khi chọn ngưỡng quá cao và quá thấp:



Ngưỡng
quá cao



Ngưỡng
quá thấp

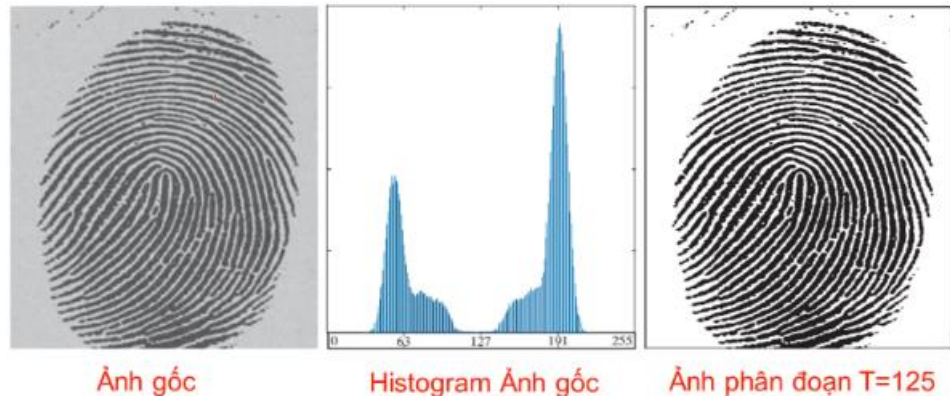
Như vậy:

- Phân đoạn ảnh bằng kỹ thuật cắt ngưỡng thì phải xác định ngưỡng T.
- Ngưỡng T được áp dụng cho toàn bộ ảnh nên gọi là Ngưỡng toàn cục
- Thuật toán tìm ngưỡng: Là những thuật toán tìm ngưỡng để áp dụng phân đoạn ảnh dựa trên kỹ thuật cắt ngưỡng toàn cục.
 - + Thuật toán tìm Ngưỡng toàn cục cơ bản
- Bước 1: Chọn giá trị ngưỡng ban đầu T (thường là trung bình mức xám trong ảnh)
- Bước 2: Sử dụng ngưỡng T để phân đoạn ảnh, tạo ra 2 nhóm điểm ảnh
 - o Nhóm G1 chứa các pixel với mức xám $> T$
 - o Nhóm G2 chứa các pixel với mức xám $\leq T$
- Bước 3: Tính mức xám trung bình nhóm G1 là m_1 và mức xám trung bình nhóm G2 là m_2
- Bước 4: Tính ngưỡng mới:

$$T = \frac{m_1 + m_2}{2}$$

- Bước 5: Lặp lại bước 2 đến 4 cho đến khi giá trị của T trong các lần lặp tiếp theo nhỏ hơn ΔT
- ➔ Kết quả của thuật toán tìm ngưỡng toàn cục cơ bản cho kết quả ngưỡng

$T = 125, 4$. Thực hiện sau 3 lần lặp, bắt đầu cho T là giá trị trung bình cường độ xám của ảnh và $\Delta T = 0$. Ảnh phân đoạn dựa trên kỹ thuật cắt ngưỡng toàn cục với $T = 125$



+ Thuật toán tìm Ngưỡng Otsu

- Bước 1: Gọi n_i là số pixel có mức xám $i \Rightarrow$ tính xác suất pixel có mức xám i $p_i = \frac{n_i}{MN}$
- Bước 2: Chọn giá trị ngưỡng ban đầu k (bất kỳ lớn hơn hoặc bằng 0)
- Bước 3: Tạo ra 2 nhóm điểm ảnh:

o Nhóm A chứa các pixel với mức xám $\leq k$ ($[0 \dots k]$)

o Nhóm B chứa các pixel với mức xám $> k$ ($[k+1 \dots L-1]$)

- Bước 4: Tính xác suất tích lũy đến mức xám k của nhóm A là $P_1(k)$ và xác suất tích lũy đến mức xám $L-1$ của nhóm B là $P_2(k)$

$$P_1(k) = \sum_{i=0}^k p_i, \quad P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k)$$

- Bước 5: Tính giá trị trung bình xác suất trong nhóm A là $m_1(k)$ và nhóm B là $m_2(k)$

$$m_1(k) = \frac{1}{P_1(k)} \sum_{i=0}^k i p_i, \quad m_2(k) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} i p_i$$

- Bước 6: Tính giá trị trung bình toàn cục của mức xám ảnh: $m_G = \sum_{i=0}^{L-1} p_i$
- Bước 7: Tính giá trị phương sai giữa 2 nhóm A và B:

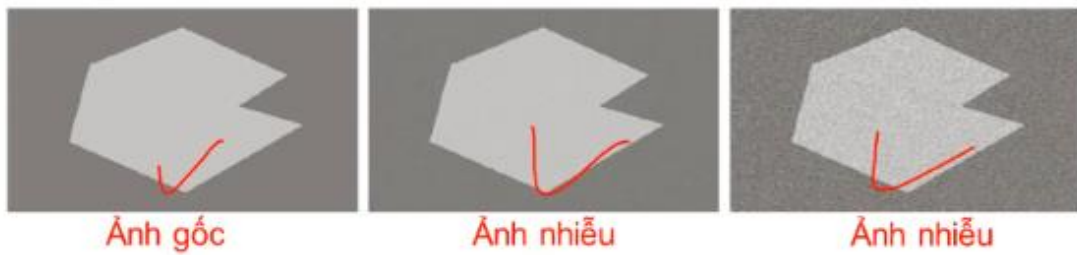
$$\sigma_B^2(k) = P_1(k)[m_1(k) - m_G]^2 + P_2(k)[m_2(k) - m_G]^2$$

- Bước 8: Giá trị ngưỡng được tính: $k^* = \max \{\sigma_B^2(k)\}$

$$0 \leq k \leq L-1$$

Ảnh hưởng nhiễu và hiệu ứng chiếu sáng trong phân đoạn ảnh bằng kỹ thuật cắt ngưỡng toàn cục:

- Khó khăn trong trường hợp ảnh bị nhiễu:



- Khó khăn trong trường hợp ảnh có hiệu ứng chiếu sáng

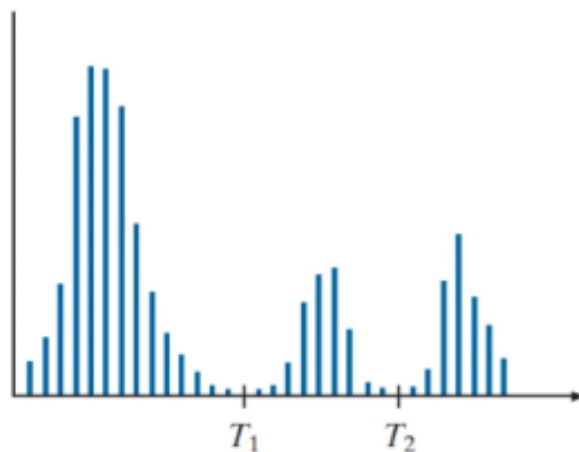


➔ Phải lọc nhiễu trước khi phân đoạn bằng kỹ thuật cắt ngưỡng

b, Phân đoạn ảnh dựa vào kỹ thuật cắt ngưỡng đa ngưỡng

Dựa vào nhiều giá trị ngưỡng T (T có nhiều giá trị)

$$g(x, y) = \begin{cases} a & \text{nếu } f(x, y) > T_2 \\ b & \text{nếu } T_1 < f(x, y) \\ c & \text{nếu } f(x, y) \leq T_1 \end{cases}$$



➔ Phân chia ảnh vào 3 vùng: Vùng mức xám = a, vùng mức xám mức = b và vùng mức xám = c

c, Phân đoạn ảnh dựa vào kỹ thuật cắt ngưỡng thích nghi

Ý tưởng: Tính giá trị ngưỡng cho mỗi pixel trong ảnh dựa trên vùng lân cận của nó

- m_G là giá trị trung bình mức xám của các pixel trong ảnh đầu vào

- S_{xy} là vùng ảnh con hình vuông với kích thước mỗi cạnh k , với tọa độ của pixel tại tâm của vùng S_{xy}
- σ_{xy} là độ lệch chuẩn của pixel tổng vùng S_{xy}

Thuật toán:

- Bước 1: Tính m_G và σ_{xy}
- Bước 2: Tính ngưỡng của pixel(x,y) trong vùng S_{xy} :

$$T_{xy} = a\sigma_{xy} + bm_G$$

Trong đó: a, b là số không âm

- Bước 3: Ảnh phân đoạn được tính bằng:

$$g(x, y) = \begin{cases} 1 & \text{nếu } f(x, y) > T_{xy} \\ 0 & \text{nếu } f(x, y) \leq T_{xy} \end{cases}$$

Trong đó: $f(x, y)$ là ảnh đầu vào

2.3 Phân vùng ảnh theo biên

2.3.1 Khái niệm cơ bản

Biên là một trong những đặc trưng quan trọng của ảnh. Cũng vì thế mà trong nhiều ứng dụng, người ta sử dụng cách phân đoạn dựa theo biên. Việc phân đoạn ảnh dựa vào biên được tiến hành qua một số bước:

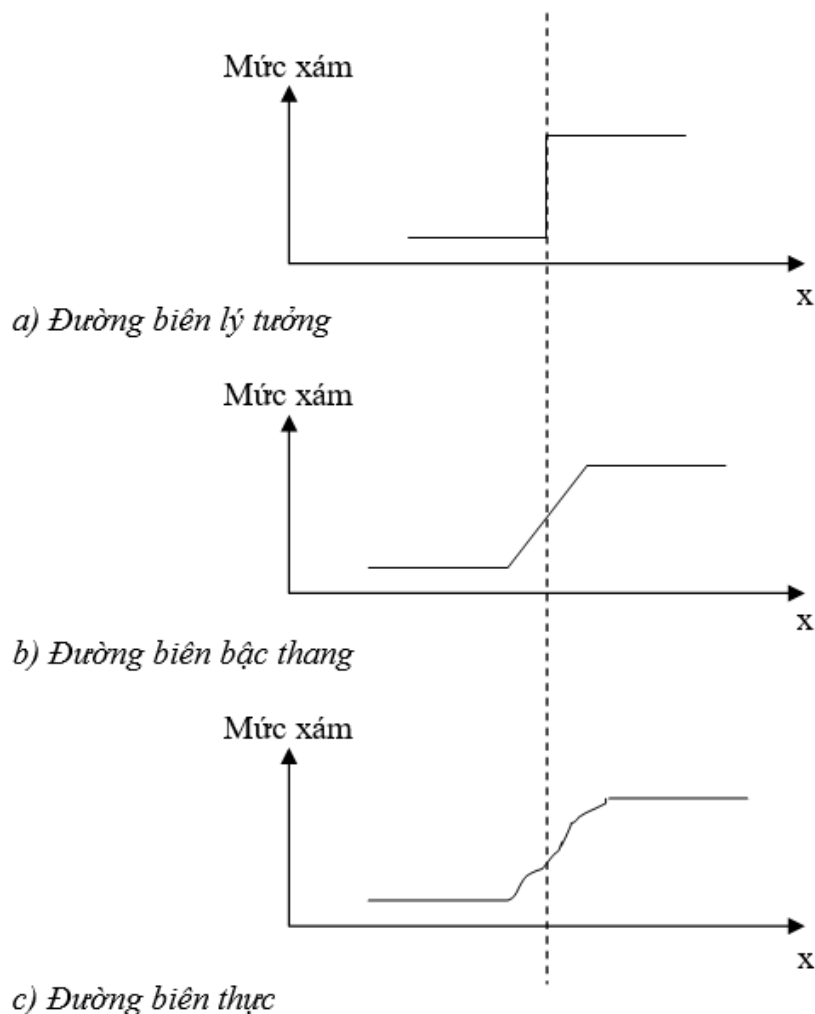
- Phát hiện và làm nổi biên.
- Làm mảnh biên.
- Nhị phân hóa đường biên.
- Mô tả biên.

2.3.2 Khái quát về biên và phân loại các kỹ thuật dò biên

2.3.2.1. Khái niệm về biên

Biên là một vấn đề chủ yếu trong phân tích ảnh vì các kỹ thuật phân đoạn ảnh chủ yếu dựa vào biên. Một điểm ảnh có thể coi là điểm biên nếu ở đó có sự thay đổi đột ngột về mức xám. Tập hợp các điểm biên tạo thành biên hay đường bao của ảnh (boundary). Thí dụ, trong một ảnh nhị phân, một điểm có thể gọi là biên nếu đó là điểm đen và có ít nhất một điểm trắng lân cận.

Để hình dung tầm quan trọng của biên ta xét thí dụ sau: Khi người họa sĩ vẽ một cái bàn gỗ, chỉ cần vài nét phác thảo về hình dáng như mặt bàn, chân bàn mà không cần thêm các chi tiết khác, người xem đã có thể nhận ra đó là cái bàn. Nếu ứng dụng của ta là phân lớp nhận diện đối tượng, thì coi như nhiệm vụ đã hoàn thành. Tuy nhiên nếu đòi hỏi thêm về các chi tiết khác như vân gỗ hay màu sắc, ... thì với chừng ấy thông tin là chưa đủ. Nhìn chung về mặt toán học, người ta coi điểm biên của ảnh là điểm có sự biến đổi đột ngột về độ xám như chỉ ra trong hình dưới đây:



Như vậy, phát hiện biên một cách lý tưởng là xác định được tất cả các đường bao trong các đối tượng. Định nghĩa toán học của biên ở trên là cơ sở cho các kỹ thuật phát hiện đường biên. Điều quan trọng là sự biến thiên mức xám giữa các điểm ảnh trong một vùng thường là nhỏ, trong khi đó biến thiên mức xám của điểm ở vùng giáp ranh (khi qua biên) lại khá lớn.

2.3.2.2. Phân loại các kỹ thuật phát hiện biên

Xuất phát từ định nghĩa toán học của biên người ta thường sử dụng 2 phương pháp phát hiện biên sau:

- Phương pháp phát hiện biên trực tiếp: phương pháp này nhằm làm nổi biên dựa vào sự biến thiên về giá trị độ sáng của điểm ảnh. Kỹ thuật chủ yếu dùng phát hiện biên ở đây là kỹ thuật đạo hàm. Nếu lấy đạo hàm bậc nhất của ảnh ta có phương pháp gradient; nếu lấy đạo hàm bậc hai ta có kỹ thuật Laplace. Hai phương pháp trên được gọi là phương pháp dò biên cục bộ. Ngoài ra, người ta cũng sử dụng phương pháp "đi theo đường hao": dựa vào nguyên lý qui hoạch động và được gọi là phương pháp đồ biên tổng thể.
- Phương pháp gián tiếp: nếu bằng cách nào đấy, ta phân được ảnh thành các vùng thì đường phân ranh giữa các vùng đó chính là biên. Việc phân vùng ảnh thường dựa vào kết cấu (texture) bề mặt của ảnh.

Cũng cần lưu ý rằng, kỹ thuật dò biên và phân vùng ảnh là hai bài toán đối ngẫu của nhau. Thực vậy, dò biên để thực hiện phân lớp đối tượng và một khi đã phân lớp xong có nghĩa là đã phân vùng được ảnh. Và ngược lại, khi đã phân vùng, ảnh đã phân lập được thành các đối tượng, ta có thể phát hiện được biên. Phương pháp dò biên trực tiếp tỏ ra khá hiệu quả vì ít chịu ảnh hưởng của nhiễu. Song nếu sự biến thiên độ sáng không đột ngột, phương pháp này lại tỏ ra kém hiệu quả. Phương pháp dò biên gián tiếp tuy khó cài đặt, song lại áp dụng khá tốt khi sự biến thiên độ sáng nhỏ.

2.3.2.3. Quy trình phát hiện biên

- Bước 1: Vì ảnh thu nhận thường nhiễu, nên bước đầu tiên là phải khử nhiễu.
- Bước 2: Tiến hành làm nổi biên bởi các toán tử đạo hàm.
- Bước 3: Định vị điểm biên. Vì các kỹ thuật làm nổi biên có hiệu ứng phụ là tăng nhiễu, do vậy sẽ có một số điểm biên giả cần phải loại bỏ.
- Bước 4: Liên kết và trích chọn biên.

2.3.2.4. Làm mảnh biên

Làm giảm hay mảnh biên thực chất là làm nổi biên với độ rộng chỉ 1 điểm ảnh (pixel). Như trong chương tăng cường ảnh, kỹ thuật Laplace dùng trong phát hiện biên cho kết quả trực tiếp biên ảnh với độ rộng 1 điểm ảnh (pixel). Song với nhiều kỹ thuật khác thì không hoàn toàn như vậy.

Khi thực hiện đạo hàm một ảnh, ta thu được những điểm cực trị cục bộ. Theo kỹ thuật gradient, những điểm cực trị cục bộ có thể coi như biên. Do vậy cần tách biệt những điểm cực trị đó để xác định chính xác biên ảnh và để giảm độ rộng biên ảnh. Một phương pháp hay dùng trong kỹ thuật làm mảnh biên chữ là phương pháp "Loại bỏ các điểm không cực đại".

Giả sử ảnh $I(x,y)$ gồm gradient hướng và gradient biên độ (còn gọi là bản đồ biên độ và bản đồ hướng). Với mỗi điểm ảnh $I(x,y)$ ta xác định các điểm lân cận của nó theo hướng gradient. Gọi các điểm đó là $I(x1,y1)$ và $I(x2,y2)$. Các điểm này được minh họa trong hình 6.7.

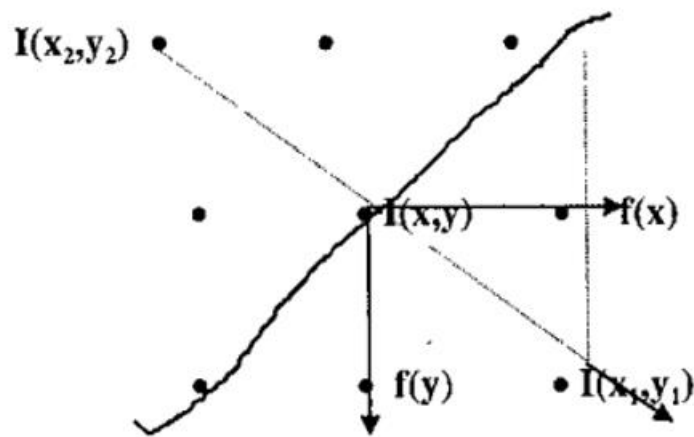
Nếu $I(x,y)$ lớn hơn cả $I(x1,y1)$ và $I(x2,y2)$, giá trị của $I(x,y)$ sẽ được bảo toàn. Nếu không nó sẽ được đặt là 0 và coi như bị loại bỏ.

Ngoài thuật toán kể trên còn nhiều kỹ thuật làm mảnh biên để xác định khung của đối tượng (ảnh hay ký tự) như kỹ thuật mảnh biên chữ do Sherman đề xuất sau đó được Fraser cải tiến cho ảnh nhị phân. Kỹ thuật này có thể mô tả tóm tắt như sau:

Tại mỗi vị trí của số, phần tử trung tâm sẽ được xóa (đổi thành trắng) nếu nó thỏa

mãn một trong hai điều kiện sau:

- Nó là điểm đen duy nhất kết nối với 2 điểm đen không kề nhau.
- Nó là điểm đen có duy nhất một lân cận cũng là điểm đen ngoại trừ không tồn tại một chuyển đổi nào tại phần tử trước nó.



Cụ thể là điểm đó phải đồng thời thoả mãn các điều kiện sau

$$\begin{cases} 2 \leq NZ(P1) \leq 6 \\ Z0(P1) = 1 \\ P2 * P4 * P8 = 0 \text{ hay } Z0(P2) \neq 1 \end{cases}$$

Với:

+ $Z0(P1)$ là số phần tử 0 chuyển sang phần tử khác 0 trong tập có thứ tự $P2, P4, P8, P1$

+ $NZ(P1)$ là số hàng xóm (lân cận) khác 0 của $P1$

Hình dưới đây mô tả một số tình huống áp dụng và không áp dụng được điều kiện này và thí dụ về ảnh sau khi đã làm mảnh.

P_3	P_2	P_9
P_4	P_1	P_8
P_5	P_6	P_7

1	1	0
0	P_1	0
0	0	0

1	1	0
0	P_1	1
0	0	0

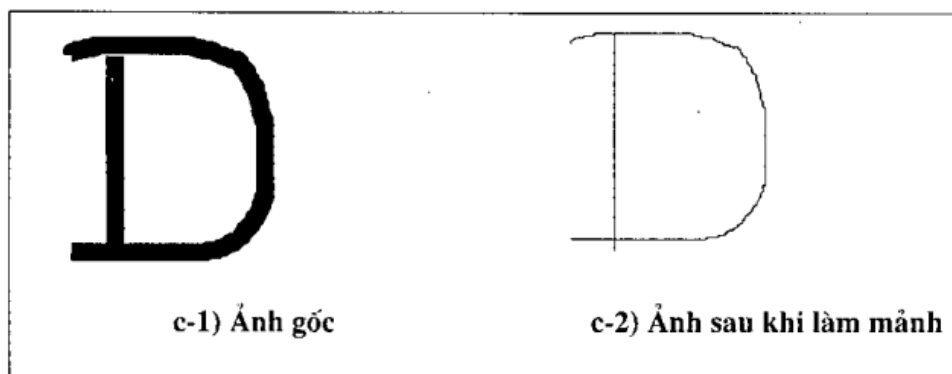
1	0	0
1	P_1	0
0	0	0

a) Điểm ảnh P_1
và 8 lân cận

b-1) xoá P_1 sẽ tạo
thành 2 miền

b-2) xoá P_1 sẽ gây
ngăn đoạn

b-3) $2 < NZ(P1) < 6$
xong không xóa được



2.3.2.5. Nhị phân hoá đường biên

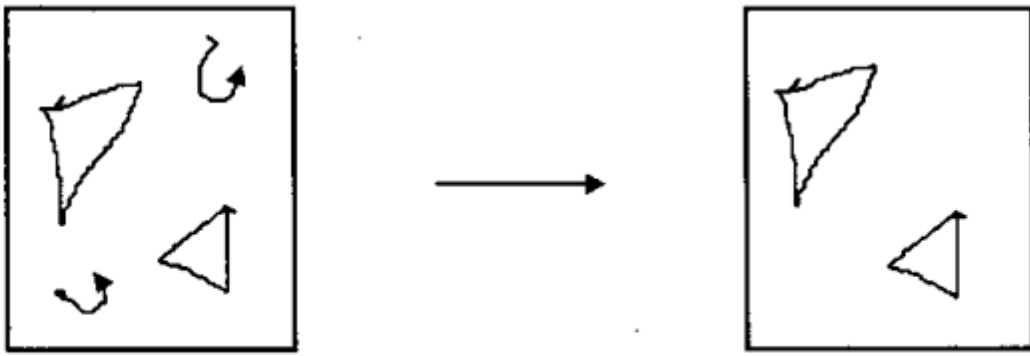
Nhị phân hóa đường biên là giai đoạn then chốt trong quá trình trích chọn vì nó xác định đường bao nào thực sự cần và đường bao nào có thể loại bỏ. Nói chung, người ta thường nhị phân hóa đường biên theo cách thức làm giảm nhiễu hoặc tránh hiện tượng kéo sợi trên ảnh. Điều này cũng giải thích tại sao phân

đoạn dựa theo biên có hiệu quả khi ảnh có độ tương phản tốt. Trong trường hợp ngược lại, có thể sẽ bị mất một phần đường bao hay đường bao có chân, không khéo kín, do đó sẽ bất lợi cho biểu diễn sau này. Một phương pháp hay được dùng là chọn ngưỡng thích ứng. Với cách chọn này, ngưỡng sẽ phụ thuộc vào hướng của gradient nhằm làm giảm sự xoắn của biên độ. Lúc đầu người ta định ra một ngưỡng nào đó và sau đó sử dụng hệ số thích ứng thông qua lời giải toán tử đạo hàm theo hướng tìm được để tinh chỉnh.

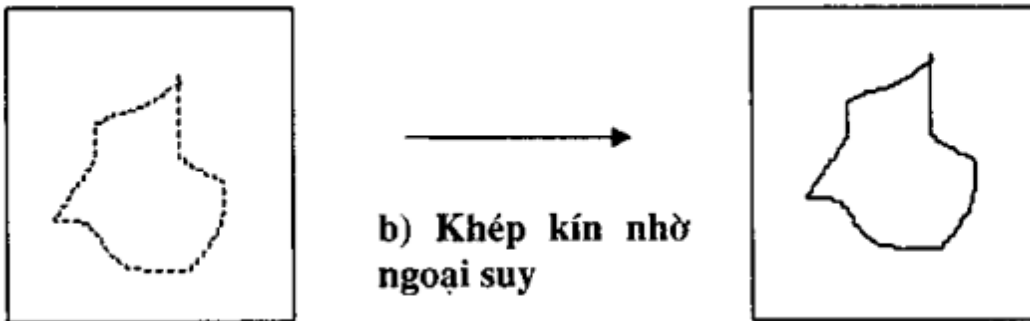
2.3.2.6. *Mô tả đường biên*

Khi đã có bản đồ biên của ảnh, ta cần phải biểu diễn nó dưới dạng thích hợp phục vụ cho việc phân tích và làm giảm lượng thông tin dùng để miêu tả đối tượng. Nguyên tắc chủ yếu là tách biệt mỗi biên và gán cho nó một mã. Bắt đầu từ thời điểm này, chúng ta không quan tâm tới cách biểu diễn quen thuộc của ảnh bởi ma trận số mà ta quan tâm đến cách miêu tả bởi một cấu trúc thích hợp và cô đọng.

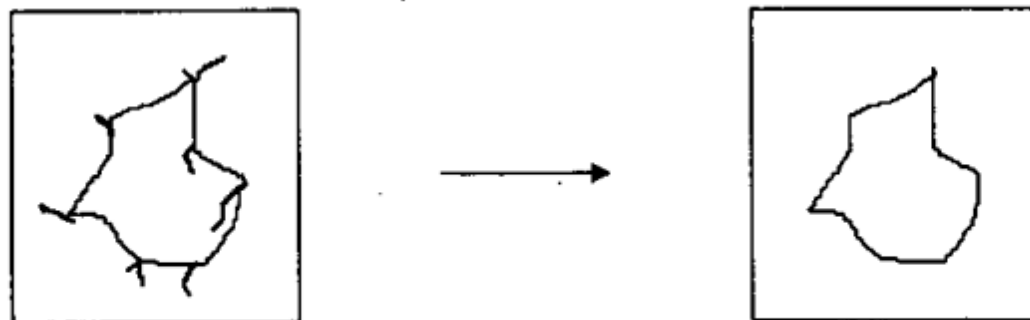
Quá trình miêu tả biên đối tượng là khá rộng vì có nhiều phương án khác nhau và mỗi phương án có liên quan mật thiết với các đặc thù của từng ứng dụng. Việc tách biệt đường bao có thể được tăng cường thêm các điều kiện nhằm loại bỏ các đường bao không khép kín hoặc khép kín nhờ phương pháp ngoại suy, hay bỏ đi các chân rết bám theo các đường bao kín.



a) Loại bỏ đường bao hở



b) Khép kín nhờ ngoại suy



c) Loại bỏ các chân rết

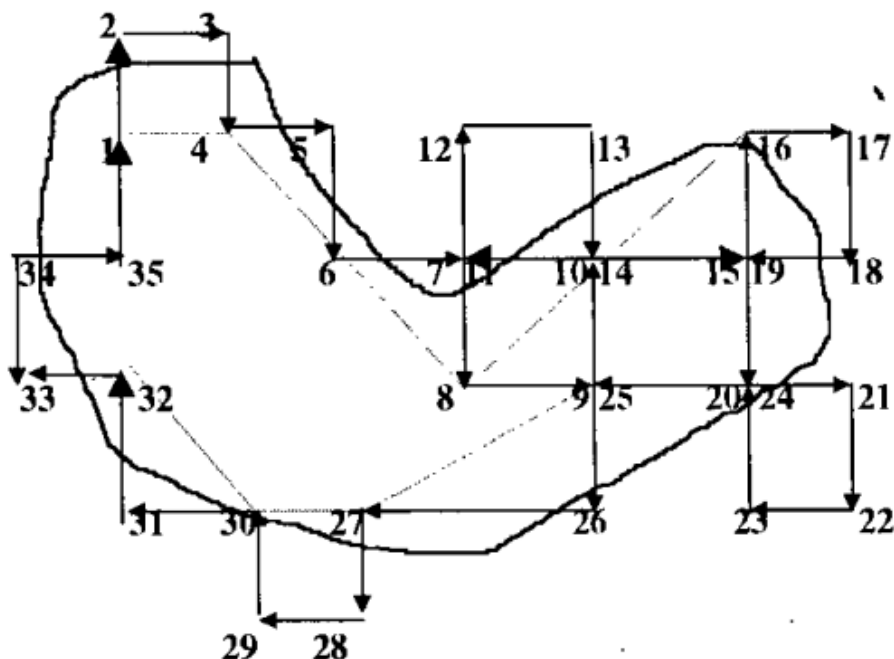
Việc mã hoá đường bao có thể thực hiện theo nhiều cách khác nhau. Có thể dùng biểu diễn chính xác đường bao, hay xấp xỉ nhờ nội suy. Thông thường các cấu trúc cơ sở mã hoá đường biên bao gồm 4 loại: điểm, đoạn thẳng, cung và đường cong.

Tuy nhiên, cũng cần lưu ý là luôn có xung đột giữa độ phức tạp tính toán và khả năng biểu diễn ảnh của cách mã hoá thông tin. Biểu diễn đường bao bởi các điểm nói chung không phức tạp song lại rất nghèo nàn về cấu trúc và không cô đọng (vì có bao nhiêu điểm tạo nên biên ta phải mô tả bấy nhiêu). Trong khi đó, biểu diễn bằng đường cong đa thức bậc cao làm tăng độ phức tạp tính toán, song bù lại cấu trúc dữ liệu lại rất cô đọng. Một số phương pháp mã hoá đường bao hay dùng sẽ được trình bày dưới đây:

a, Mã hoá theo toạ độ Đề các (Descartes)

Kiểu mã hoá này khá đơn giản. Đường bao của ảnh được biểu diễn bởi một danh sách các điểm ảnh tạo nên đường bao. Gọi C là đường bao của ảnh và $C(i,j)$ là các điểm. Rõ ràng cách mã hoá này bộc lộ nhược điểm là không giảm được lượng thông tin. Tuy nhiên, việc tính toán lại khá nhanh và có thể cung cấp những phương tiện phục vụ cho việc trích chọn các đặc trưng hình học của ảnh

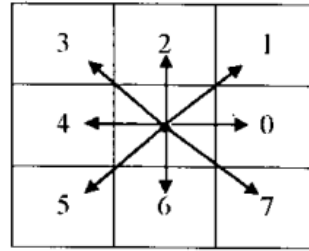
Việc mã hoá được thực hiện theo phương pháp “Đi theo đường bao”, sử dụng kỹ thuật tìm kiếm thông tin theo chiều sâu trên cây. Nếu áp dụng một cách đơn thuần kỹ thuật này, có thể ta sẽ thu được đường bao thô theo nghĩa có điểm có thể xuất hiện 2 lần. Người ta thường phối hợp với việc kiểm tra 8 liên thông để làm mịn biên.



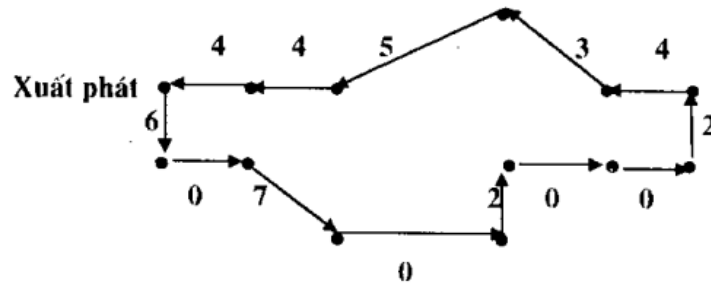
Hình trên cho ta danh sách các điểm ảnh trên đường bao được duyệt theo thứ tự từ 1 đến 35.

b, Mã hoá Freeman

Phương pháp mã hoá này cũng là một cách biểu diễn chính xác các điểm đường bao bằng việc sử dụng vị trí tương đối của điểm trên đường bao với điểm trước. Nguyên tắc của việc mã hoá này thể hiện trên hình 6-12a. Người ta dùng một mặt nạ để xác định mã của mỗi điểm trong j8 liên thông so với điểm ở tâm. Từ một điểm đã cho trên đường bao, người ta mã hoá đường bao bằng cách đi theo nó (hình 6-12b). Tuy nhiên, người ta thích mã hoá theo góc giữa các cung (6-12c). Trong trường hợp này, mã góc sẽ được tính bởi: $goc_i = \text{MOD}8(|\text{arci} - \text{arci-1} + 11| - 3|$. Hình 6-12d là mã của 6-12b theo góc và gọi là mã đường bao.

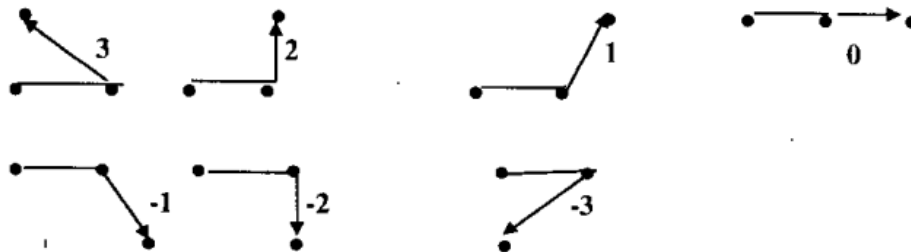


a) 8 liên thông và mã hướng tương ứng.

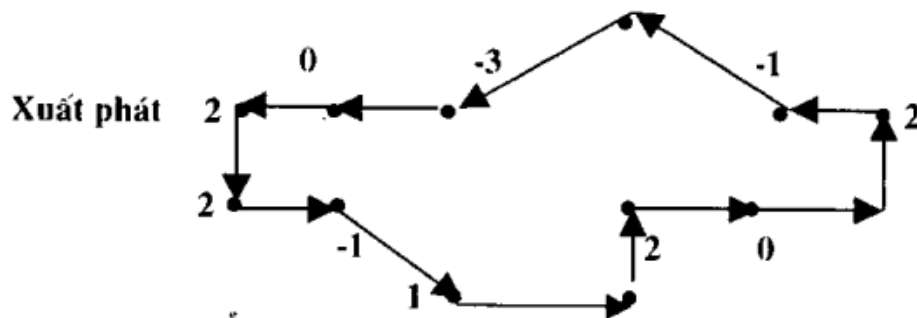


Cung = { 6 0 7 0 2 0 0 2 4 3 5 4 4 }

b) Đồ thị biểu diễn đường bao có trọng số.



c) Mã hoá góc.



Góc = { 2 2 -1 -1 2 -2 0 2 2 -1 2 -1 0 }

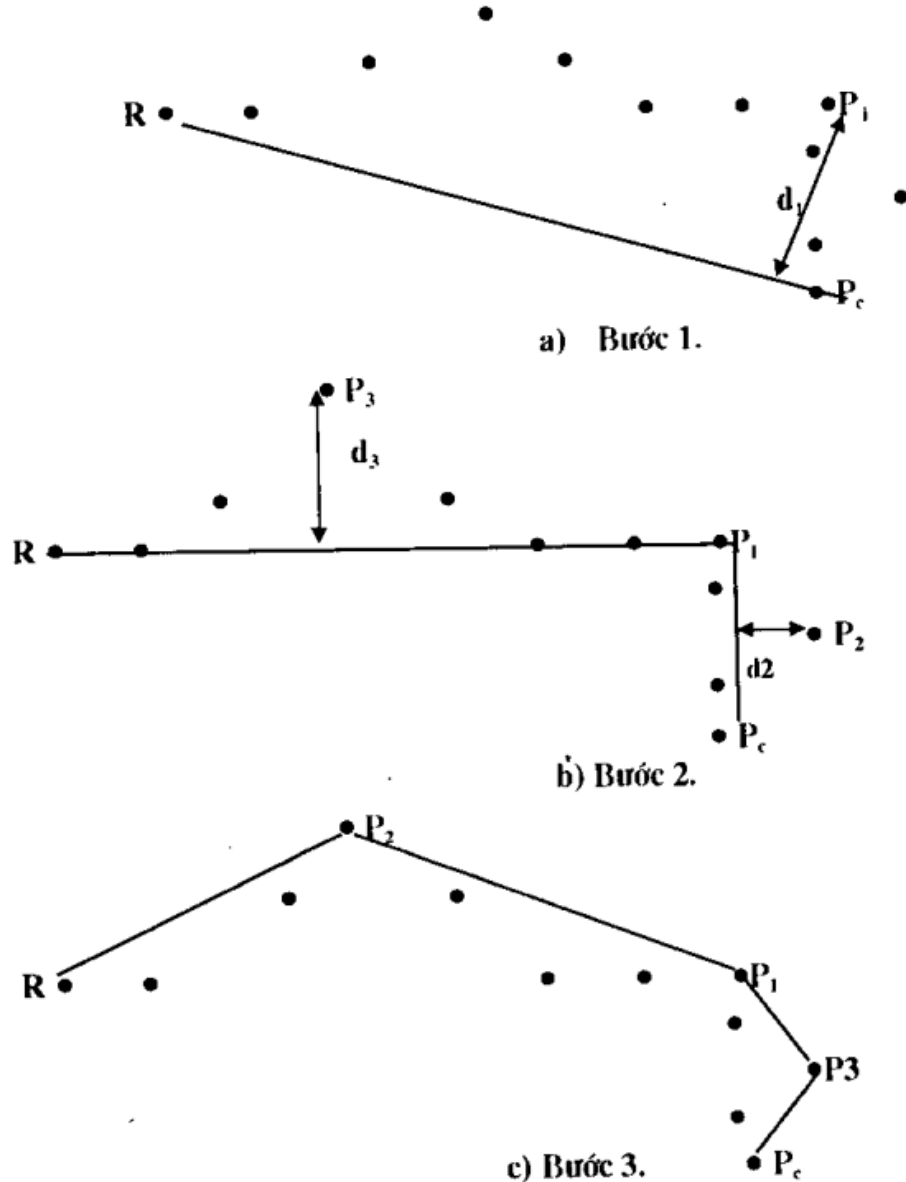
d) Mã hóa theo góc.

2.3.2.7. Xấp xỉ bởi đường thẳng

Ngược với hai cách mã hoá ở trên, kỹ thuật mã hoá bởi đoạn thẳng không cho phép khôi phục tất cả các thông tin chứa đựng trong đường bao nhưng lại có thể xấp xỉ nó bởi đoạn thẳng với độ chính xác phụ thuộc vào người dùng. Nếu độ chính xác càng thấp, thông tin mô tả càng cô đọng.

Có thể mô tả tóm tắt như sau: tiến hành lấy mẫu đường bao theo nhiều giai đoạn với bước nhảy là P. Nối điểm xuất phát R với điểm đang xét P_c bởi một

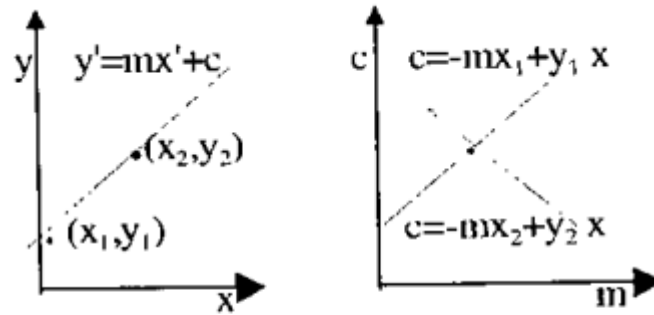
đoạn thẳng. Sau đó tính tọa độ của P_i , một điểm nằm giữa R và P_c sao cho khoảng cách từ P_i đến đoạn thẳng là cực đại. Gọi khoảng cách này là d_1 . Nếu d_1 lớn hơn một ngưỡng cho trước (độ chính xác của xấp xỉ) người ta phân đoạn RP_c thành hai đoạn RP_i và P_iP_c và tiếp tục thực hiện lấy mẫu với từng đoạn cho tới khi đoạn thẳng tìm được là "rất gần" với đường bao.



Cũng trong phương pháp xấp xỉ bởi đoạn thẳng, có một cách tiếp cận hoàn toàn khác với phương pháp trên, đó là phép biến đổi Hough. Cách tiếp cận đó như sau: Giả sử (x', y') là một điểm trong ảnh. Mọi điểm qua (x', y') phải thỏa phương trình:

$$y' = mx' + c \text{ với } m, c \text{ là các tham số (*)}$$

Viết lại phương trình trên ta có: $c = -mx' + y'$



a) đường thẳng qua (x_1, y_1) và qua (x_2, y_2) b) giao điểm của 2 đường

Như vậy, mọi đường thẳng qua (x', y') tương ứng với một điểm trong không gian tham (c, m) . Xét hai điểm (x_1, y_1) và (x_2, y_2) nằm trên m cùng một đường thẳng:

Với mỗi điểm ảnh, mọi đường thẳng qua m được biểu diễn bởi một điểm trong (c, m) . Thí dụ: $P(x_1, y_1): c = -mx_1 + y_1$

$$P(x_2, y_2): c = -mx_2 + y_2$$

Nhưng một đường thẳng duy nhất trong không gian (x, y) qua 2 điểm (x_1, y_1) và (x_2, y_2) được biểu diễn bởi giao của 2 đường trong không gian tham số (c, m) . Điểm giao cho giá trị của c và m trong phương trình $y = mx + c$ (hình 6.14a, b).

Để áp dụng kỹ thuật này, không gian tham số (c, m) cần phải được lượng hoá và như vậy ta cần dùng một ma trận tham số $P(c, m)$ với:

$c_1 \leq c \leq c_2$ và $m_1 \leq m \leq m_2$; K là số điểm chia của C và N là số điểm ảnh

Cách tiến hành như sau:

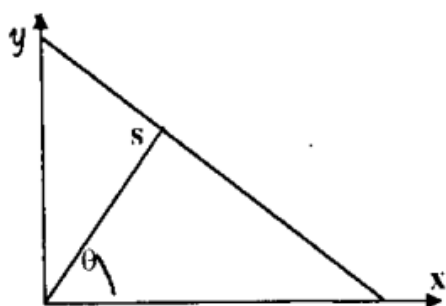
- Khởi tạo bảng tham số $P(c, m)$: các phần tử của bảng này được gán là 0.
- Với mỗi điểm $(x_i, y_i) \in$ biên ảnh, $P(c, m) = P(c, m) + 1$ cho các điểm thoả phương trình (*).
- Lặp lại quá trình trên cho đến khi toàn bộ ảnh được quét.

Kết thúc các bước này, mỗi phần tử của ma trận $P(c, m)$ chứa số điểm biên thoả (*). Nếu số điểm này vượt quá một ngưỡng T nào đấy thì một đường thẳng dạng $y' = mx' + c$ được khởi tạo. Cần chú rằng, trong biến đổi Hough, các điểm nằm trên cùng một đường thẳng không nhất thiết là liên tục. Đây là một tính chất quan trọng.

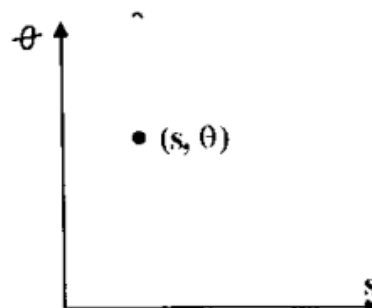
Tuy nhiên, cần xem xét phương pháp này có áp dụng trong thực tế được không? Điều đó có nghĩa là cần tính độ phức tạp tính toán của phương pháp. Dễ dàng thấy rằng độ phức tạp tính toán là $O(KL)$ với K và L có nghĩa như trên. Độ phức tạp này là chấp nhận được. Song cũng dễ thấy, nếu biểu diễn bởi đường thẳng thì khi biểu diễn các đường đứng (verticale straight line) thì c có xu hướng tiến ra vô cùng. Một cách khắc phục là dùng hệ toạ độ cực (s, θ) . Trong biến đổi

Hough, một đường thẳng trong một mặt phẳng với khoảng cách s và hướng θ có thể biểu diễn bởi: $s = x \cos \theta + y \sin \theta$ (hình 6.14 c và d).

Đường thẳng này có thể coi như một điểm trong mặt phẳng (s, θ) như trong hình . Kỹ thuật này sử dụng tất cả các điểm của đường bao và bản đồ gradient hướng. Biến đổi Hough tạo nên một ánh xạ các điểm của đường thẳng với một điểm trong mặt phẳng.



c) đường thẳng.



d) biến đổi Hough.

Giả sử các điểm của đường bao là (x_i, y_i) , $i=1, 2, \dots, N$. Với một số giá trị đã chọn của s và θ , ảnh xạ cặp (x_i, y_i) thành điểm (s, θ) có nghĩa là với 1 điểm ta đi tìm phương trình đường thẳng đi qua điểm này và có hướng vuông góc với gradient hướng. Ta tính được $C(s, \theta)$. Ta thu được tập biểu diễn bởi quan hệ sau:

$$C(s_k, \theta_i) = C(s_k, \theta_i) + 1 \text{ nếu } x_i \cos \theta + y_i \sin \theta = s_k \text{ với } \theta_i = \theta_i (**)$$

Cực trị cục bộ $C(s, \theta)$ cho các đường thẳng khác nhau đi qua điểm biên này. Lưu ý rằng việc tìm kiếm trong không gian 2 chiều có thể biến đổi sang không gian một chiều nếu gradient hướng θ_i ở mỗi điểm biên là đã biết.

Miền biến thiên của các tham số:

$$-\sqrt{N1^2 + N2^2} \leq s \leq \sqrt{M1^2 + M2^2}$$

$$\frac{-\pi}{2} \leq \theta \leq \frac{\pi}{2}$$

2.3.2.8. Xấp xỉ đa thức

Nếu ta dùng phương pháp xấp xỉ đường cong (biên thường cong) bằng đường thẳng như trên, sai số sẽ lớn. Để giảm sai số người ta nghĩ đến dùng xấp xỉ bởi đường cong đa thức. Cũng giống như xấp xỉ đường thẳng, ta cũng phải lấy mẫu đường bao để tính các hệ số của đa thức. Các đa thức thường được dùng là loại spline.

Giả sử đường bao được biểu diễn theo tham số t . Các tọa độ tương ứng của một điểm trên đường bao là $x(t)$, $y(t)$. Việc biểu diễn bởi đường cong đa thức có thể viết:

$$x(t) = \sum_{i=0}^n p_i B_{i,k}(t)$$

Với p_i là các điểm kiểm tra và $B_{i,k}$, $i = 1, 2, \dots, n$; $k = 1, 2, \dots$ là các đường cong chuẩn bậc k . Các đa thức $B_{i,k}$ được tính 1 cách hồi quy như sau:

$$\begin{cases} B_{i,k}(t) = \frac{(t-t_i)B_{i,k} - i(t)}{t_{i+k-1} - t_i} + \frac{(t_{i,k} - t)B_{i+1,k} - i(t)}{t_{i+k-1} - t_i} \text{ với } k = 2, 3, \dots \\ \text{và } B_{i,1}(t) = 1 \text{ nếu } t_i \leq t \leq t_{i+1} \text{ và } B_{i,1}(t) = 0 \text{ nếu không.} \end{cases}$$

Các t_i , $i = 0, 1, \dots$ gọi là các mẫu. Các đỉnh liên kết với các mẫu kí hiệu là s_i và được định nghĩa như trung bình cục bộ của chuỗi $k-1$ mẫu:

$$s_i = (t_{i+1} + t_{i+2} + \dots + t_{i+k}) / (k-1) \quad k > 2 \text{ và } 0 \leq i \leq n$$

Các t_i , s_i là các giá trị đặc biệt của tham số đỉnh. Thí dụ với $k = 3$ ta có đường bậc 2, $k = 4$ ta có dạng lập phương.

Nếu việc phân đoạn là đều, có nghĩa là $t_{i+1} - t_i = \Delta t$ với mọi i thì $B_{i,k}(t)$ là đường spline đều và ta có:

$$B_{i,k}(t) = B_{0,k}(t-i), \quad i = k-1, \dots, n-k+1$$

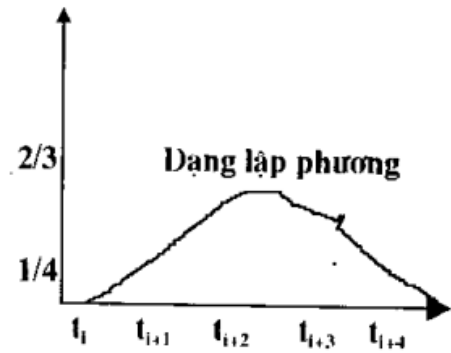
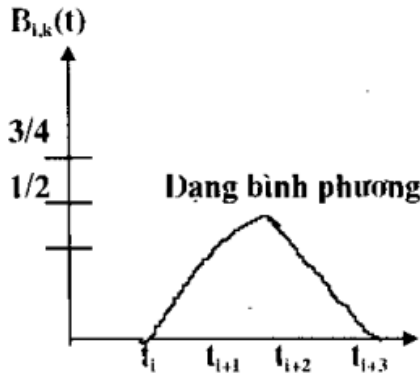
Nếu $\Delta t = 1$ và đường spline là mở và đều, các t_i sẽ được chọn bởi:

$$t_i = \begin{cases} 0 & i < k \\ i - k + 1 & k \leq i \leq n \\ n - k + 2 & i > n \end{cases}$$

Nếu đường spline có chu kỳ đều, các t_i sẽ được chọn bởi:

$$t_i = i \bmod (n+1)$$

$$B_{i,k}(t) = B_{0,k}(t-i \bmod (n+1))$$



2.3.2.9. Nói lóng thuật toán phân vùng

Các phương pháp trên gọi là phương pháp tất định (deterministic) theo nghĩa mỗi điểm ảnh chỉ thuộc một vùng. Tuy nhiên, thực tế có điểm nằm trên đường giáp ranh và như vậy nó thuộc nhiều vùng. Sẽ có ích nếu ta xây dựng một véc-tơ P_k cho mỗi điểm x_k . Véc-tơ này chứa xác suất $p_k(i)$ để thuộc R_i , $i=1, 2, \dots, N$:

$$P_k = [p_k(1), p_k(2), \dots, p_k(N)]$$

Véc-tơ này có thể được sử dụng để tạo nên sự phân vùng tất định: điểm ảnh x_k sẽ được gán cho vùng R_i có xác suất $p_k(1)$ lớn nhất. Xác suất $p_k(1)$ cũng được gọi là trọng số tin cậy (confidence weight) và phải thoả mãn:

$$p_k^{(n)}(i) = \frac{1/|f(n,l) - m_i|}{\sum_{i=1}^N 1/|f(n,l) - m_i|} \quad i=1, \dots, N$$

Xét $d_i = |f(n,l) - m_i|$ là khoảng cách từ điểm có cường độ sáng $f(n,l)$ với miền R_i có trung bình m_i , $i = 1, \dots, N$

Trên thực tế, rất có thể có 2 điểm ảnh kề cận thuộc về 2 miền khác nhau R_i , R_j .

Các miền như vậy gọi là miền tương hợp. Các miền không tương hợp là các miền không hy vọng tìm thấy các vị trí ảnh lân cận.

Để miêu tả sự tương hợp giữa 2 miền R_i và R_j , người ta đưa ra các khái niệm hàm tương hợp: $r(i,j)$ với $-1 \leq r(i,j) \leq 1$ và được định nghĩa:

$$r(i, j): \begin{cases} < 0 & R_i, R_j \text{ tương hợp} \\ = 0 & R_i, R_j \text{ độc lập} \\ > 0 & R_i, R_j \text{ tương hợp} \end{cases}$$

Các hàm tương hợp có thể xác định trước hoặc được ước lượng nhờ phân vùng ban đầu. Khi đã có hàm tương hợp, ta có thể sử dụng để cập nhật lại vectơ tin cậy p_k .

Thường các miền không tương hợp có xu hướng cạnh tranh các điểm lân cận, trong khi đó các miền tương hợp lại có xu hướng hợp lại. Sự cạnh tranh và hợp nhất diễn ra liên tục cho đến khi đạt được một trạng thái ổn định. Mỗi điểm x_i nhận được một độ tin cậy thêm từ các điểm x_i trong lân cận và trọng số $p_k(i)$ của điểm x_k , tại bước n được tính:

$$\Delta P_k^n(i) = \sum_l d_{kl} \sum_{j=1}^N r_{kl}(i,j) P_l^n(j)$$

Việc lấy tổng được thực hiện trên tất cả các trọng số tin cậy $p_i(j)$ của các điểm x_i nằm trên lân cận với x_k .

$$P_k^{(n+1)}(i) = \frac{P_k^{(n)}(i)[1 + \Delta P_k^{(n)}(i)]}{\sum_{i=1}^N P_k^{(n)}(i)[1 + \Delta P_k^{(n)}(i)]}$$

Quá trình lặp sẽ dừng khi tiêu chuẩn hội tụ đạt được. Khi thuật toán kết thúc sẽ tạo nên một vùng đồng nhất liên thông khá lớn nhờ loại bỏ được các vùng nhiễu giả tạo bên trong. Cách phân vùng tắt định có thể dễ dàng thực hiện bằng cách chọn $P_k(l)$ lớn nhất: x_k thuộc vùng R_l . Mức độ hiệu quả của kỹ thuật này phụ thuộc vào việc chọn hàm tương hợp. S.Peleg và A.Roselfield đã nêu ra một cách chọn sau:

$$r_{kl}(i,j) = \ln \frac{N^2 \sum_{k=1}^{N^2} P_k^{(n)}(i) P_l^{(o)}(i)}{\sum_{k=1}^{N^2} P_k^{(n)}(i) \sum_{k=1}^{N^2} P_l^{(o)}(i)}$$

với N^2 là số điểm ảnh của ảnh gốc và $P_k^{(0)}(i)$ là xác suất khởi tạo ban đầu. Cần chú ý rằng giá trị $r_k(i,j)$ phải thuộc khoảng đóng $[-1, 1]$

2.4 Lược đồ xám Histogram

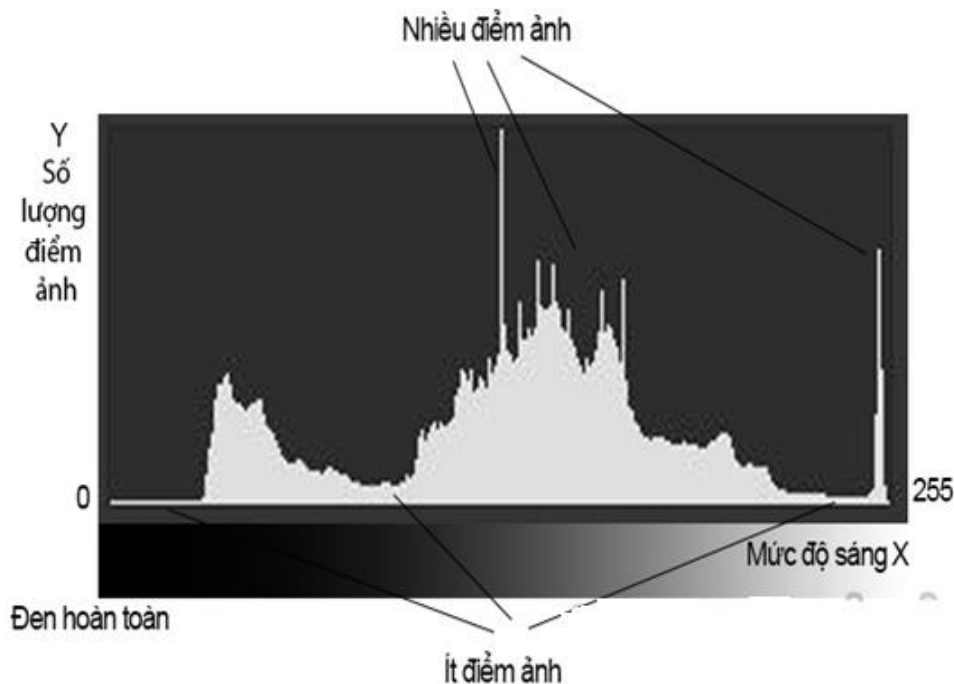
Là một đồ thị dạng thanh biểu diễn tần suất xuất hiện các mức xám của ảnh. Trong đó trục hoành biểu diễn giá trị mức xám của ảnh có giá trị từ 0 đến 255, trục tung biểu diễn tần suất xuất hiện mức xám của ảnh.

Công thức tổng quát:

$$p(r_k) = \frac{n_k}{MN}$$

Chú thích:

- $p(r_k)$ là tần suất xuất hiện mức xám.
- n_k là giá trị điểm ảnh tại vị trí k.



CHƯƠNG 3. CÁC PHƯƠNG PHÁP XỬ LÝ ẢNH TRONG NHẬN DIỆN BIẾN SỐ XE

3.1 Giảm nhiễu

Noise được hiểu cơ bản là các dạng chấm hạt nhỏ phân bố trên hình ảnh. Noise có thể làm biến dạng các chi tiết trong ảnh khiến cho chất lượng ảnh thấp.

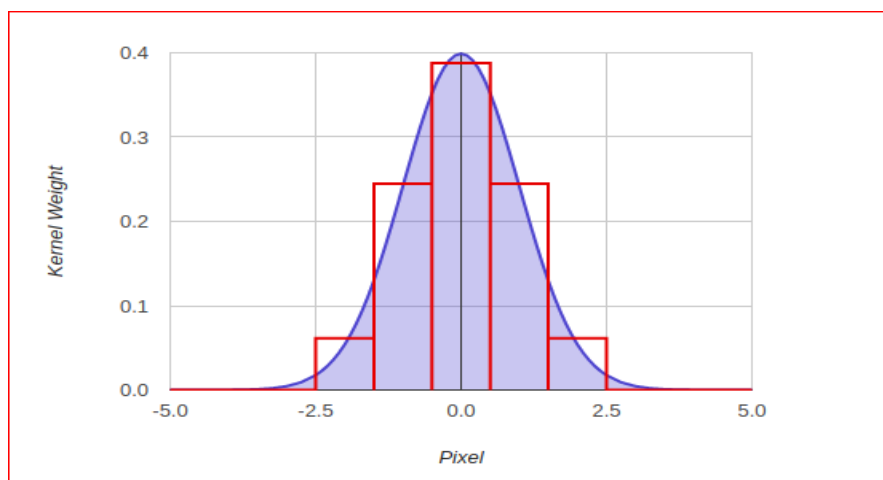
Trên thực tế có nhiều loại nhiễu, nhưng người ta thường chia làm ba loại: nhiễu cộng, nhiễu nhân và nhiễu xung. Bản chất của nhiễu thường tương ứng với tần số cao và cơ sở lý thuyết của bộ lọc là chỉ cho những tín hiệu có tần số nhất định đi qua, nên người ta thường sử dụng bộ lọc thông thấp hay trung bình.



3.1.1 Bộ lọc Gauss

Bộ lọc Gauss được cho là bộ lọc hữu ích nhất, được thực hiện bằng cách nhân chập ảnh đầu vào với một ma trận lọc Gauss sau đó cộng chúng lại để tạo thành ảnh đầu ra.

Ý tưởng chung là giá trị mỗi điểm ảnh sẽ phụ thuộc nhiều vào các điểm ảnh ở gần hơn là các điểm ảnh ở xa. Trọng số của sự phụ thuộc được lấy theo hàm Gauss (cũng được sử dụng trong quy luật phân phối chuẩn).



Hình 3.1 Ma trận lọc Gauss

Giả sử ảnh là một chiều. Điểm ảnh ở trung tâm sẽ có trọng số lớn nhất. Các điểm ảnh ở càng xa trung tâm sẽ có trọng số giảm dần khi khoảng cách từ chúng tới điểm trung tâm tăng lên. Như vậy điểm càng gần trung tâm sẽ càng đóng góp nhiều hơn vào giá trị điểm trung tâm.



Ảnh gốc



Ảnh sau khi làm mờ, giảm nhiễu

Hình 3.2 Kết quả sau khi lọc Gauss

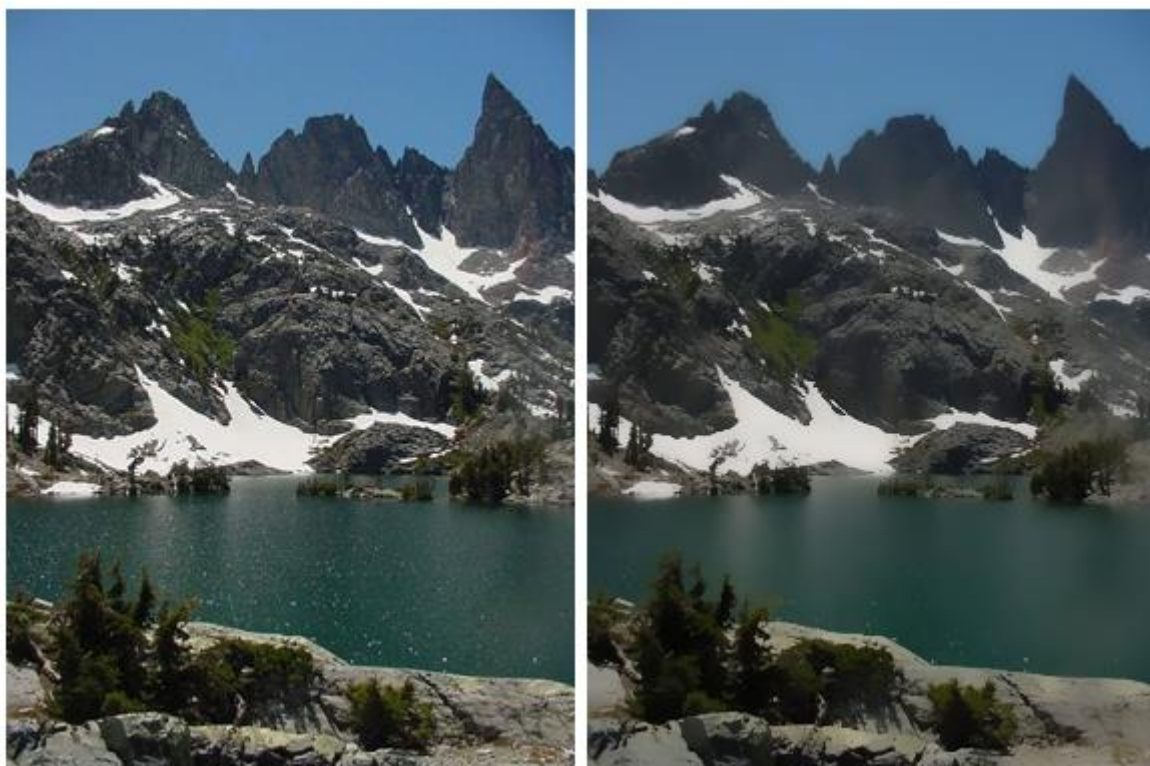
3.1.2 Bô lọc Bilateral

Bộ lọc Bilateral hay còn gọi là bộ lọc hai chiều, nó là một bộ lọc hiệu quả cao trong việc loại bỏ nhiễu mà vẫn giữ lại được các đường viền (cạnh) trong ảnh. Do bản chất là bộ lọc làm mịn không tuyến tính, bảo toàn cạnh và giảm nhiễu. Giá trị cường độ của mỗi pixel được thay thế bằng giá trị cường độ trung bình có trọng số từ các pixel lân cận.

- Phương pháp này lọc một cách có lựa chọn để làm mờ các pixel có cường độ tương tự trong một vùng lân cận. Các cạnh được giữ nguyên.
- Phương pháp này cho phép kiểm soát không chỉ kích thước không gian của bộ lọc mà còn cả mức độ bao gồm các pixel lân cận trong đầu ra được lọc. Điều này được thực hiện, dựa trên sự thay đổi về cường độ màu của chúng và cũng như khoảng cách từ pixel được lọc.

Lọc song phương về cơ bản áp dụng phương pháp làm mờ Gaussian 2D (có trọng số) cho hình ảnh, đồng thời xem xét sự thay đổi về cường độ của các pixel lân cận để giảm thiểu việc làm mờ gần các cạnh (mà muốn giữ nguyên). Điều này có nghĩa là dạng của mặt nạ phụ thuộc vào nội dung hình ảnh cục bộ, tại mọi vị trí pixel.

Ví dụ chúng ta đang cần lọc một vùng trong hình ảnh, gần một cạnh. Bộ lọc làm mờ Gaussian đơn giản sẽ làm mờ cạnh vì nó nằm gần vùng được lọc (gần tâm của bộ lọc Gaussian). Nhưng bộ lọc song phương có thể phân biệt được cạnh, bởi vì nó cũng xem xét sự khác biệt về cường độ pixel. Vì vậy, nó sẽ tính toán trọng lượng thấp hơn nhiều cho các pixel nằm dọc theo cạnh, do đó giảm ảnh hưởng của chúng lên vùng được lọc. Các vùng có cường độ đồng đều hơn sẽ bị mờ nặng hơn, vì chúng không liên kết mạnh với các cạnh.



Hình 3.3 Kết quả sau khi lọc Bilateral (ảnh trái)

3.2 Xử lý nhị phân

3.2.1 Ảnh nhị phân

Là ảnh mà giá trị của các điểm ảnh chỉ được biểu diễn bằng hai giá trị là 0 (Đen) và 255 (Trắng). Để đi nhị phân hoá bức ảnh, ta có thể làm bằng cách đặt ngưỡng hoặc nhận diện biên. Và nhóm sẽ đi thực hiện nhị phân hoá ảnh theo cách đặt ngưỡng dùng thuật toán Otsu và nhận diện biên bằng thuật toán Canny.

3.2.2 Thuật toán Otsu

Otsu là tên một nhà nghiên cứu người Nhật đã nghĩ ra ý tưởng cho việc tính ngưỡng một cách tự động dựa vào giá trị điểm ảnh của ảnh đầu vào nhằm thay thế cho việc sử dụng ngưỡng cố định.

Kỹ thuật Otsu là một trong những thuật toán hiệu quả nhất để xác định tự động giá trị ngưỡng thích nghi cho các ảnh được thu nhận trong các ngữ cảnh khác nhau. Mặc dù vậy, thuật toán yêu cầu ảnh đầu vào chứa hai thành phần chính: đối tượng (foreground) và nền (background). Ý tưởng cơ bản của thuật toán Otsu đó là tìm giá trị ngưỡng tối ưu mà sau khi phân ngưỡng thì các điểm ảnh bên trong mỗi lớp sẽ gần nhau nhất có thể (mật độ các điểm ảnh trong mỗi lớp khá dày đặc), nói cách khác phương sai trung bình của các điểm ảnh bên trong mỗi lớp sẽ đạt giá trị cực tiểu.

Các bước trong thuật toán:

Bước 1: Đầu tiên sử dụng lược đồ Histogram biểu diễn tần suất xuất hiện mức xám. Tính các giá trị:

$$p_i = \sum_{i=0}^{L-1} \frac{n_i}{MN}$$

Chú thích:

- n_i là số lượng điểm ảnh của giá trị i
- MN : là tổng số điểm ảnh trên ảnh
- L : là tổng số ngưỡng độ xám trên ảnh (0, ..., 256)
- $p_0 + p_1 + \dots + p_{L-1} = 1$

Bước 2: Tính tổng tích lũy $P_1(k)$ với $k = 0, \dots, L-1$ theo công thức:

$$P_1(k) = \sum_{i=0}^k p_i$$

Bước 3: Tính trung bình tích lũy đến mức k là $m(k)$:

$$m(k) = \sum_{i=0}^k i \cdot p_i$$

Bước 4: Tính trung bình cường độ toàn cục m_G :

$$m_G = \sum_{i=0}^{L-1} i \cdot p_i$$

Bước 5: Tính phương sai giữa các nhóm $\sigma_B^2(k)$:

$$\sigma_B^2(k) = \frac{(m_G P_1(k) - m(k))^2}{P_1(k) \cdot (1 - P_1(k))}$$

Bước 6: Ngưỡng Otsu k^* là giá trị k thỏa mãn điều kiện $\sigma_B^2(k)$ đạt cực đại. Nếu cực đại không duy nhất thì k^* được tính là trung bình các giá trị k tương ứng với các cực đại tìm thấy được.

Ví dụ: Cho ảnh sau, tìm ngưỡng tự động theo phương pháp Otsu.

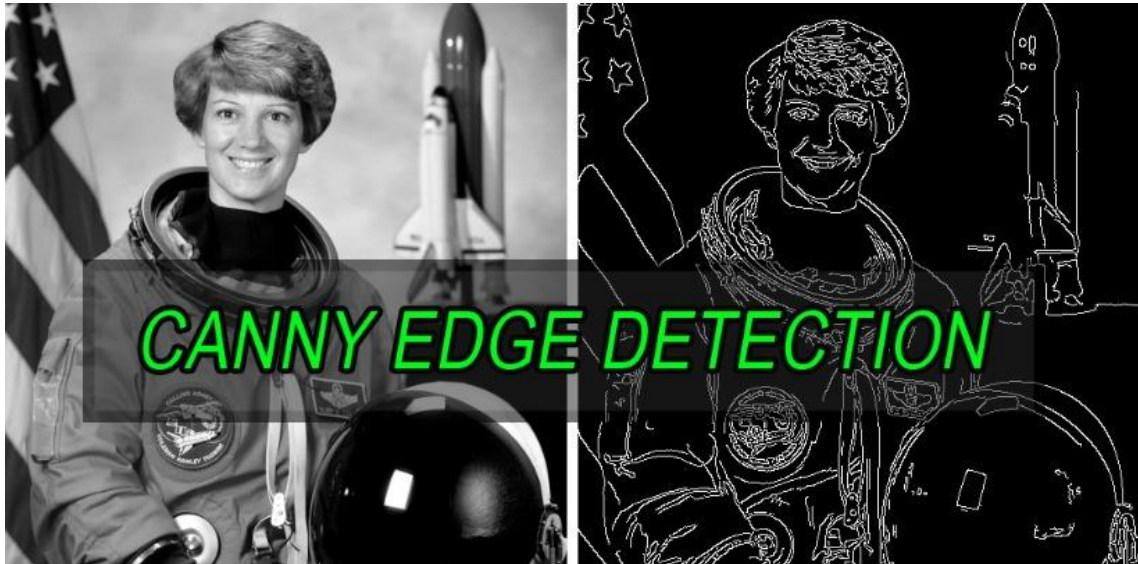
$$I = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

i	p_i	$P_i(k)$	$m(k)$	m_G	σ_B^2
0	15/30	0.5000	0.0000	1.1667	1.36
1	5/30	0.6667	0.1667	1.1667	1.68
2	4/30	0.8000	0.4333	1.1667	1.56
3	3/30	0.9000	0.7333	1.1667	1.11
4	2/30	0.9667	1.0000	1.1667	0.51

5	1/30	1.0000	1.1667	1.1667	0.00
---	------	--------	--------	--------	------

Như vậy ngưỡng $T = 1$ với $\sigma_B^2 = 1.68$.

3.2.3 Thuật toán Canny



Bộ lọc Canny là sự kết hợp của nhiều bước khác nhau để tìm và tối ưu đường biên, kết quả là cho ra một đường biên khá mạnh và chính xác. Quá trình tìm biên sử dụng phương pháp Canny có thể được thực hiện qua 4 bước sau:

Bước 1: Loại bớt nhiễu trong ảnh

Người ta loại nhiễu trong ảnh, làm mờ ảnh đi bằng cách nhân chập ảnh với một bộ lọc Gauss, chẳng hạn bộ lọc Gauss 5x5 với hệ số $\delta = 1.4$:

$$M = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Bước 2: Tính toán giá trị Gradient trong ảnh

Vì đường biên trong ảnh là nơi phân cách giữa các đối tượng khác nhau, nên tại đó gradient của nó sẽ có biến đổi mạnh mẽ nhất. Để tính toán gradient trong ảnh, ta có thể sử dụng bộ lọc Sobel, hoặc trực tiếp nhân chập ma trận ảnh với các mặt nạ theo hướng x và y, chẳng hạn:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

Sau đó tính độ lớn gradient trong ảnh:

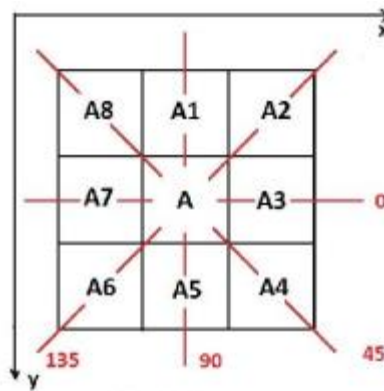
$$G = \sqrt{G_x^2 + G_y^2} \text{ và } \theta = \arctan\left(\frac{G_y}{G_x}\right)$$

Trong đó, G_x và G_y chính là đạo hàm theo hướng x, y của ảnh ta đang xét. Góc θ sẽ được làm tròn theo các hướng thẳng đứng, nằm ngang và theo hướng chéo. Nghĩa là nó sẽ được làm tròn để nhận các giá trị trong 0, 45, 90 và 135 độ.

Bước 3: Loại bỏ các giá trị không phải cực đại

Bước này sẽ tìm ra những điểm ảnh có khả năng là biên ảnh nhất bằng cách loại đi những giá trị không phải là cực đại trong bước tìm gradient ảnh ở trên. Ta thấy rằng, với giá trị của góc θ ở trên thì biên của đối tượng có thể tuân theo bốn hướng, và ta có bốn khả năng sau:

- Nếu $\theta = 0$, khi đó điểm A sẽ được xem xét là một điểm trên biên độ nếu độ lớn gradient tại A lớn hơn gradient của các điểm tại A3, A7.
- Nếu $\theta = 45$, khi đó điểm A được xem là một điểm trên biên độ nếu độ lớn gradient tại A lớn hơn độ lớn gradient của các điểm tại A4, A8.
- Nếu $\theta = 90$, khi đó điểm A sẽ được coi là một điểm nằm trên biên độ nếu độ lớn gradient tại A lớn hơn độ lớn gradient của các điểm tại A1, A5.
- Nếu $\theta = 135$, khi đó điểm A được xem là một điểm nằm trên biên độ nếu độ lớn gradient tại A lớn hơn độ lớn gradient của các điểm tại A2, A6.



Bước 4: Chọn ra biên của đối tượng trong ảnh

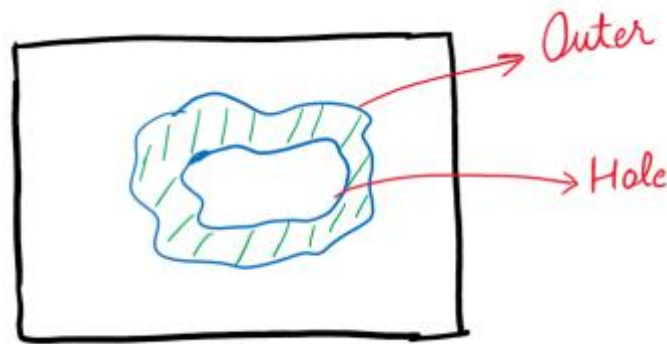
Sau bước trên, ta thu được tập hợp các điểm tương ứng trên đường biên khá mỏng. Vì những điểm có giá trị gradient lớn bao giờ cũng có xác suất biên thật sự hơn những điểm có gradient bé, do đó để xác định chính xác hơn nữa biên của các đối tượng, ta sử dụng các ngưỡng. Theo đó, bộ lọc Canny sẽ sử dụng một ngưỡng trên (upper threshold) và một ngưỡng dưới (lower threshold), nếu gradient tại một điểm trong ảnh có giá trị lớn hơn ngưỡng trên thì ta xác nhận điểm đó là một điểm biên trong ảnh, nếu giá trị này bé hơn ngưỡng dưới thì ta xác nhận điểm đó không phải là điểm biên. Trong trường hợp giá trị gradient nằm giữa ngưỡng trên và ngưỡng dưới thì nó chỉ được tính là điểm trên biên khi các điểm liên kế bên cạnh của nó có giá trị gradient lớn hơn ngưỡng trên.

3.3 Lọc biên với Contour

3.3.1 Giới thiệu

Contour là “tập các điểm-liên-tục tạo thành một đường cong (curve) (boundary), và không có khoảng hở trong đường cong đó, đặc điểm chung trong một contour là các các điểm có cùng /gần xấp xỉ một giá trị màu, hoặc cùng mật độ. Contour là một công cụ hữu ích được dùng để phân tích hình dạng đối tượng, phát hiện đối tượng và nhận dạng đối tượng”. Vậy việc ta đi tìm một contour là đi tìm một đối tượng có màu trắng trên nền đen.

Thuật toán Suzuki tracing: có khả năng phân biệt được đường biên ngoài (Outer) và đường biên trong (Hole) của vật thể.



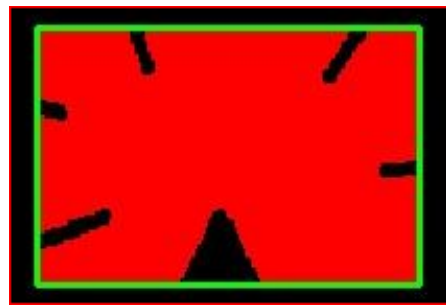
3.3.2 Xấp xỉ đa giác

Xấp xỉ đường viền: tạo ra đối tượng có dữ liệu ít hơn, hình ảnh khờp hơn đối với hình ảnh ban đầu đưa vào do hình ảnh ban đầu chứa nhiều. Do đó xấp xỉ đường viền sẽ làm giảm dữ liệu, phục vụ tốt hơn cho các bước xử lý ảnh tiếp theo. Ví dụ ta có một đường đi với nhiều điểm gấp khúc, xấp xỉ đường viền giúp ta có một đường đi giảm các đường gấp khúc, tuy nhiên vẫn giữ được hình dạng ban đầu với một sai số nhất định chấp nhận được.

Khi ta lọc biên số, đầu tiên ta làm xấp xỉ contour thành một hình đa giác và chỉ lấy những đa giác nào chỉ có 4 cạnh. Nghĩa là lúc xấp xỉ contour bộ nhớ chỉ ghi nhớ vị trí các đỉnh của đa giác đó thành một mảng. Số cạnh của đa giác sẽ bằng số đỉnh và bằng chiều dài của mảng đó.



Hình 3.4 Contour chưa xấp xỉ đa giác



Hình 3.5 Contour đã xấp xỉ đa giác

Tiếp theo ta tính toán tỉ lệ cao/rộng và diện tích của biên số phù hợp, sau đó ta lưu tất cả những biên số có trong hình dưới dạng tọa độ các đỉnh

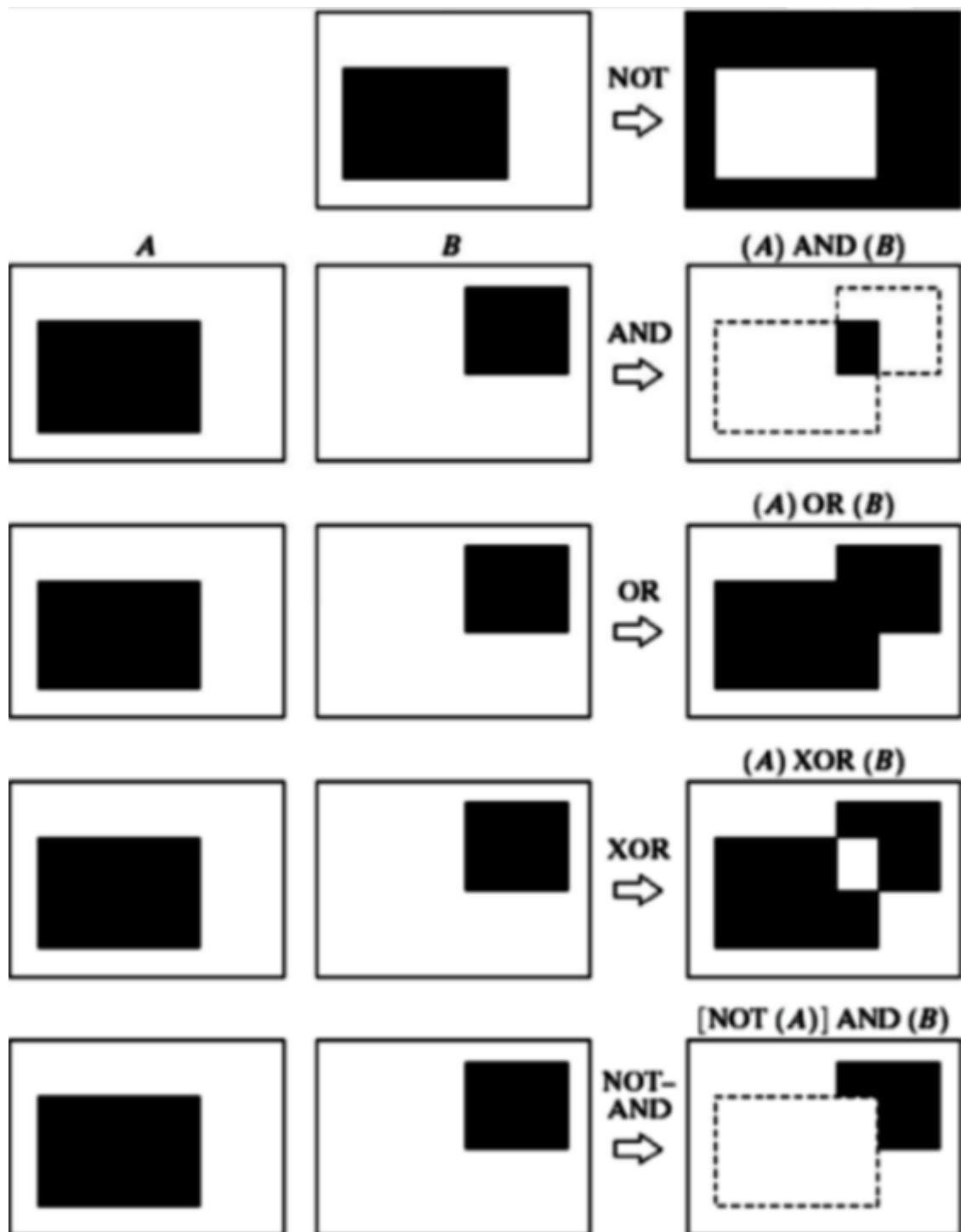
Từ đây, ta cắt hình ảnh biên số từ các tọa độ vị trí đã biết để phục vụ cho mục đích tiếp theo “Tách các kí tự trong biên số”. Lưu ý ở đây ta cắt từ ảnh nhị phân luôn để máy tính xử lý nhanh hơn, tốn ít thời gian hơn.

Trong thư viện OpenCV có hàm `Cv2.approxPolyDP()`: Dựa trên thuật toán Ramer-Douglas-Peucker. Giả sử đường viền có rất nhiều điểm, thay vì phải đi qua hết tất cả các điểm đã có, ta có thể đi thẳng đến 1 điểm xa hơn, đi lướt qua điểm bỏ qua với một khoảng cách chấp nhận được

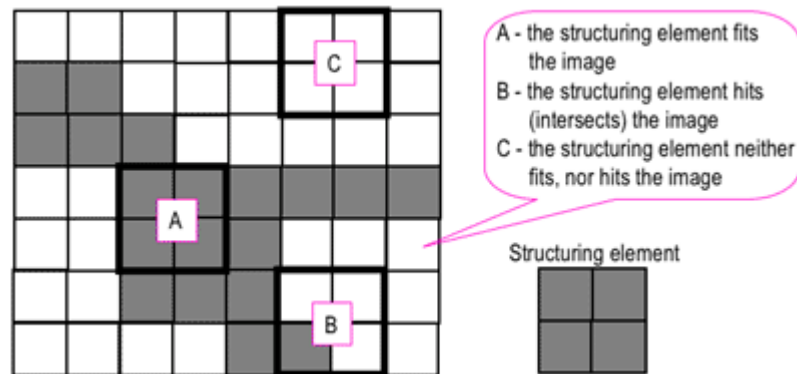
3.3.3 Xử lý ảnh hình thái học

Hình ảnh nhị phân có thể chứa nhiều yếu tố không hoàn hảo. Đặc biệt, các vùng nhị phân được tạo ra bằng các phép phân ngưỡng đơn giản và rất dễ xuất hiện nhiễu và cấu trúc không mong muốn. Do đó cần phải áp dụng kỹ thuật trong xử lý ảnh là xử lý ảnh hình thái học (Morphological Image Processing) giúp loại bỏ nhiễu, làm đẹp cấu trúc và hình thức của ảnh nhị phân. Các phép toán của kỹ thuật này có thể áp dụng được trên ảnh xám.

Morphological Image Processing là tập hợp các phép toán phi tuyến tính (non-linear) tác động đến hình dạng hoặc hình thái của các điểm nhị phân trong ảnh. Dựa trên các phép toán AND OR XOR NOT để biến đổi các điểm nhị phân.

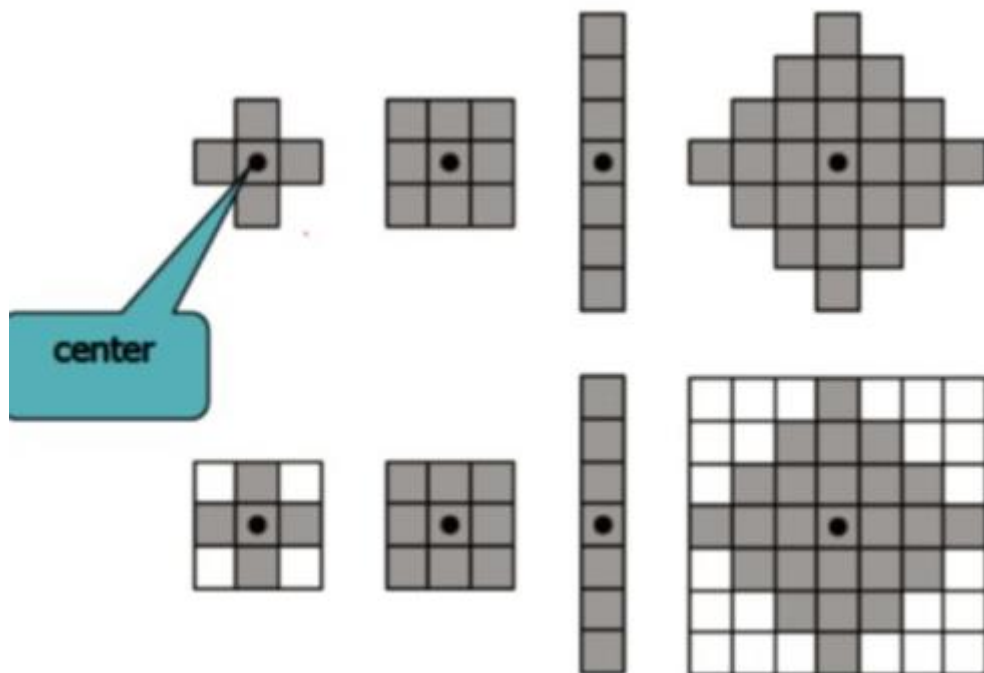


Morphological Operator sẽ sử dụng một cấu trúc nhỏ quét qua và áp dụng toàn bộ ảnh được gọi là Structuring Element (SE). SE sẽ quét qua toàn bộ vị trí trong hình ảnh và so sánh với các pixel lân cận nằm trong vùng sẽ tương ứng.



Structuring Element là một ảnh nhị phân nhỏ. Tức là một ma trận nhỏ gồm các pixel mang giá trị 0 và 1:

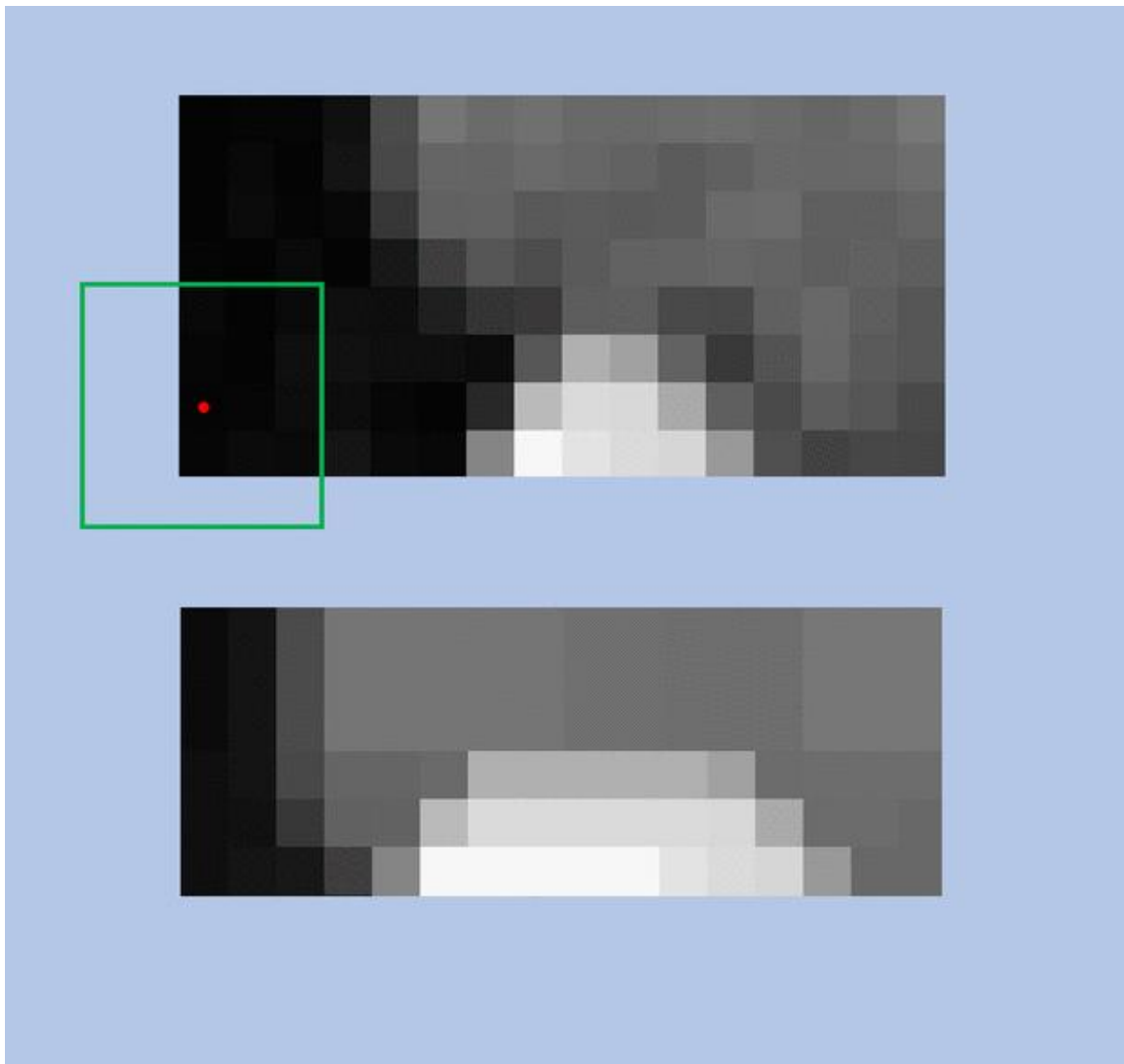
- Kích thước của ma trận xác định kích thước của SE.
- Pixel có giá trị 0 được bỏ qua trong quá trình tính toán.
- Luôn có một pixel làm mốc trong ma trận SE.



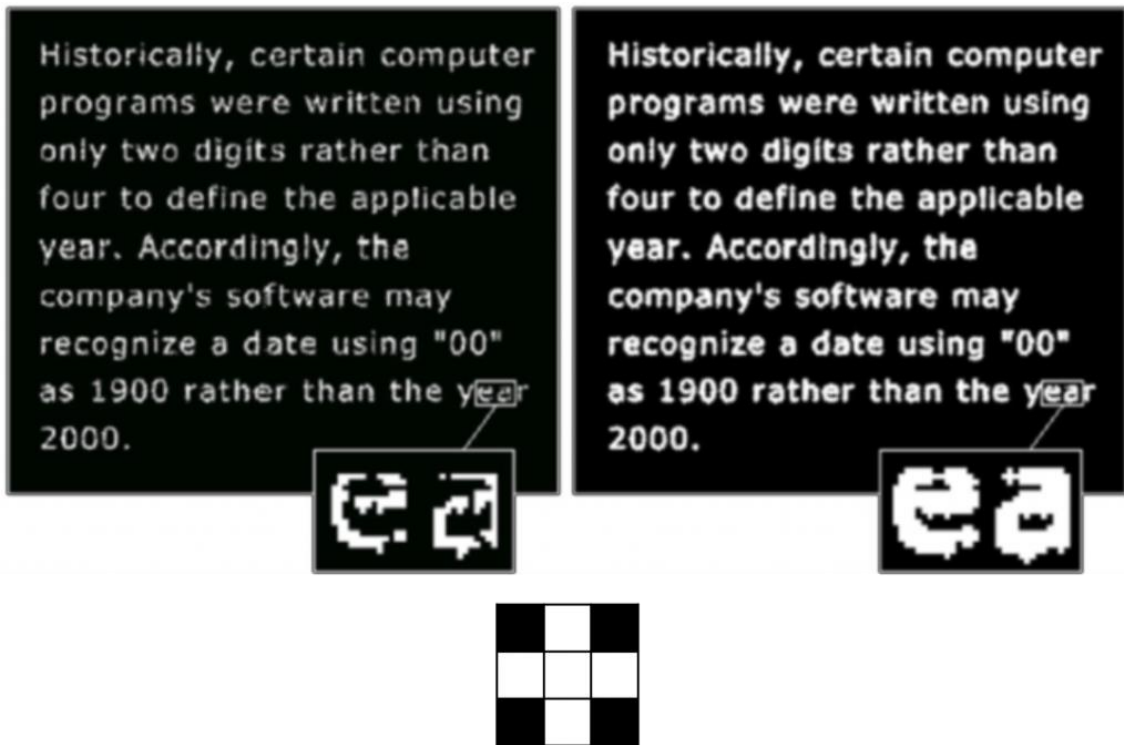
Structuring Element hoạt động giống như ma trận tích chập (convolution kernel) trong lọc ảnh tuyến tính (linear image filtering). SE sẽ dịch chuyển toàn bộ ảnh với điểm mốc.

3.3.4 Phép dẫn (Dilation) và phép co (Erosion)

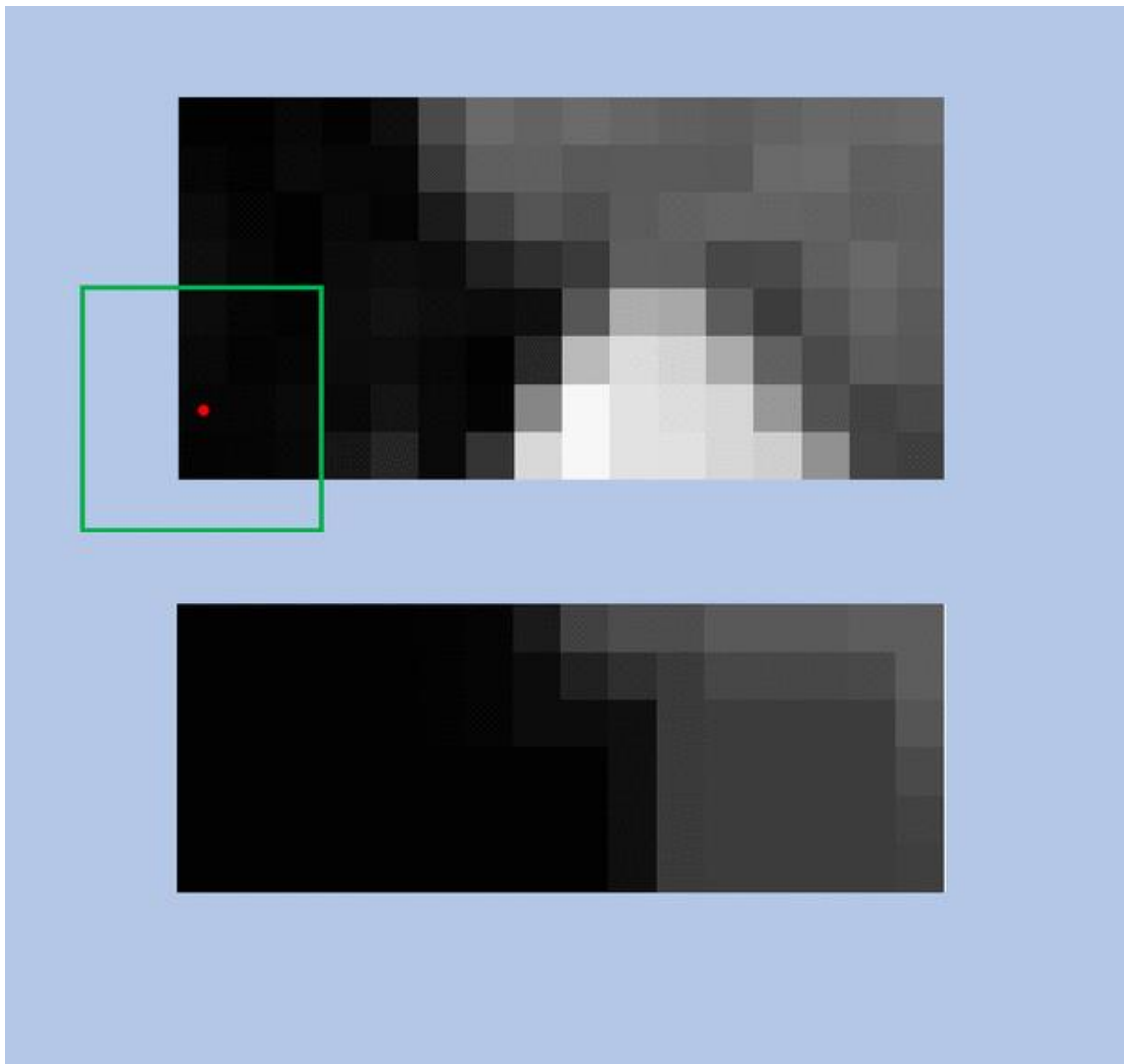
Phép dẫn của ảnh A bởi phần tử cấu trúc B cho ra ảnh G. Mỗi vị trí cấu trúc B quét qua được sẽ chọn giá trị **lớn nhất** và trả về điểm mốc tương ứng trên ảnh G.



Ví dụ có một ảnh nhị phân với các dòng chữ bị đứt nét. Sử dụng phép dẫn với cấu trúc dạng chữ thập sẽ nối liền các điểm đứt nét và các điểm đó sẽ có hơi hướng giống với cấu trúc chữ thập.



Phép co của ảnh A bởi phần tử cấu trúc B cho ra ảnh G. Mỗi vị trí cấu trúc B quét qua được sẽ chọn giá trị **nhỏ nhất** và trả về điểm mốc tương ứng trên ảnh G.



3.4 Hệ số tương quan

3.4.1 Giới thiệu

Hệ số tương quan (Coefficient of correlation) được sử dụng để đo lường độ lớn của mối quan hệ giữa hai biến số. Qua đó có thể thấy được độ mạnh yếu của mối quan hệ. Giá trị của hệ số nằm trong khoảng -1,0 đến 1,0. Nếu giá trị thu được ngoài khoảng nghĩa là phép đo tương quan bị lỗi.

Khái niệm này được nhắc nhiều trong nhiều lĩnh vực. Mỗi loại Correlation Coefficient được định nghĩa riêng, đặc tính và phạm vi sử dụng không giống nhau. Trong trường hợp Correlation Coefficient mang giá trị âm (-) thì mối quan hệ giữa hai biến số được gọi là tương quan âm hay nghịch biến. Nếu giá trị bằng -1 là nghịch biến tuyệt đối.

Hệ số tương quan nếu mang giá trị dương (+) thì 2 biến số có mối quan hệ tương quan dương hay còn gọi là đồng biến. Khi có giá trị bằng 1 đồng biến chỉ ra mối liên hệ thuận tuyệt đối, biến này tăng thì biến kia sẽ tăng và ngược lại. Hệ

số này bằng 0,0 có nghĩa giữa chuyển động của hai biến không có mối quan hệ tuyến tính.

Trong số các loại hệ Correlation Coefficient thì hệ số tương quan Pearson ký hiệu R là loại phổ biến nhất. Nhìn vào hệ số R có thể biết được mối tương quan giữa hai biến số như thế nào, độ mạnh yếu ra sao. Hiện nay tương quan R được nhiều người biết đến, đây hướng pháp tối ưu nhất giúp đo lường mối tương quan giữa các biến số.

Ta có công thức tính hệ số tương quan Pearson như sau:

$$r = \frac{N\sum xy - (\sum x)(\sum y)}{\sqrt{[N\sum x^2 - (\sum x)^2][N\sum y^2 - (\sum y)^2]}}$$

Trong đó:

N: Số cặp điểm

$\sum xy$ = tổng các tích của các điểm được ghép nối

$\sum x$ = tổng điểm x

$\sum y$ = tổng điểm y

$\sum x^2$ = tổng điểm x bình phương

$\sum y^2$ = tổng điểm y bình phương

3.5 Ví dụ

Với bài toán nhận diện kí tự quang học của biển số xe ta sử dụng hệ số tương quan bằng cách so sánh ảnh đang xét với viện các kí tự có sẵn từ 0 tới 1 và A đến Z. Sau đó ta sẽ gán nhãn cho kí tự xét là ký tự có độ tương quan Person lớn nhất.

Số điểm ảnh	Ảnh đang xét (x)	Ảnh mẫu (y)	(xy)	(x ²)	(y ²)
1	0	0	0	0	0
2	1	0	0	1	0
3	1	1	1	1	1
4	0	1	0	0	1
...					
16384(128x128)	1	0	0	1	0

Từ dữ liệu của bảng ví dụ ra sẽ tính được tổng x, y, xy, x², y² và áp dụng công thức tính hệ số độ tương quan Person ở trên để thấy sự tương quan và tìm ảnh có hệ số tương quan lớn nhất với ảnh đang xét.

CHƯƠNG 4. THỰC NGHIỆM

Nhóm sẽ đi thực hiện nhận diện biển số xe với ngôn ngữ lập trình python. Nhóm sẽ đi sử dụng một số thư viện hỗ trợ trong nhận diện biển số xe.

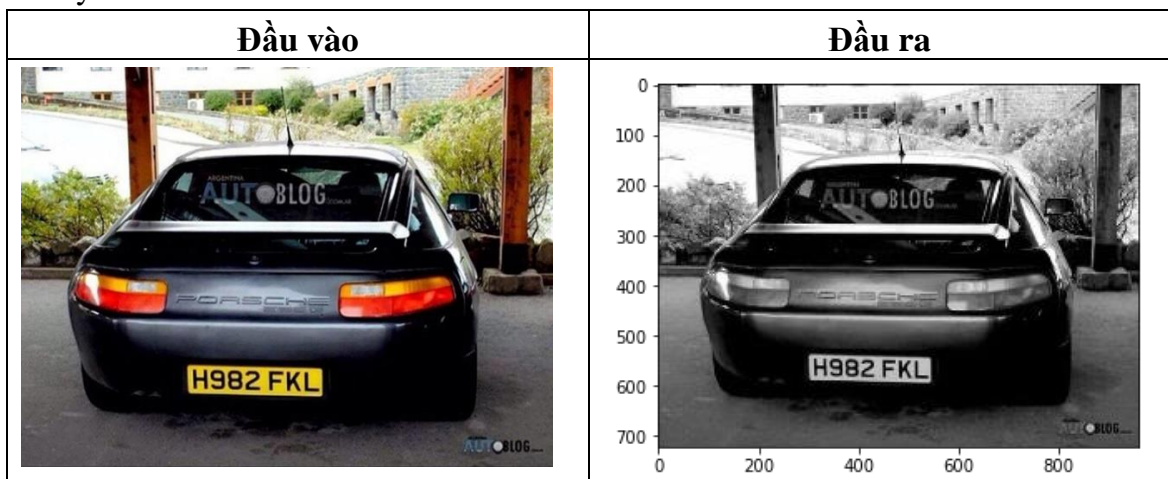
4.1 Khai báo các thư viện trong python

```
import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
```

4.2 Đọc file ảnh và chuyển xám ảnh

```
img = cv2.imread('car3.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_GRAY2RGB))
```

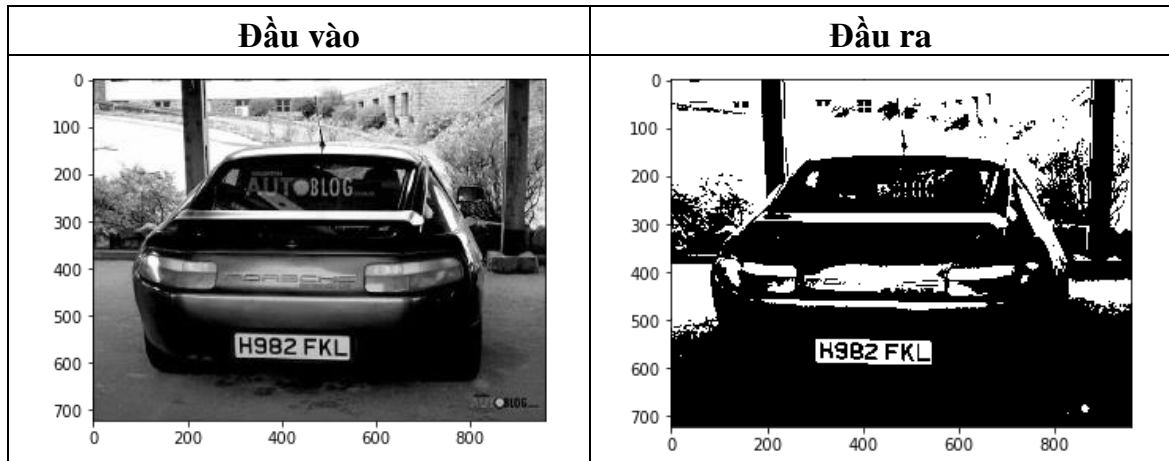
Ta dùng `img = cv2.imread()` để đọc ảnh và lệnh `cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)` để chuyển đổi ảnh thành xám và `plt.imshow(cv2.cvtColor(gray, cv2.COLOR_GRAY2RGB))` để hiển thị ảnh ta vừa xử lý.



4.3 Dùng thuật toán chọn ngưỡng hoặc nhận diện biên xử lý ảnh

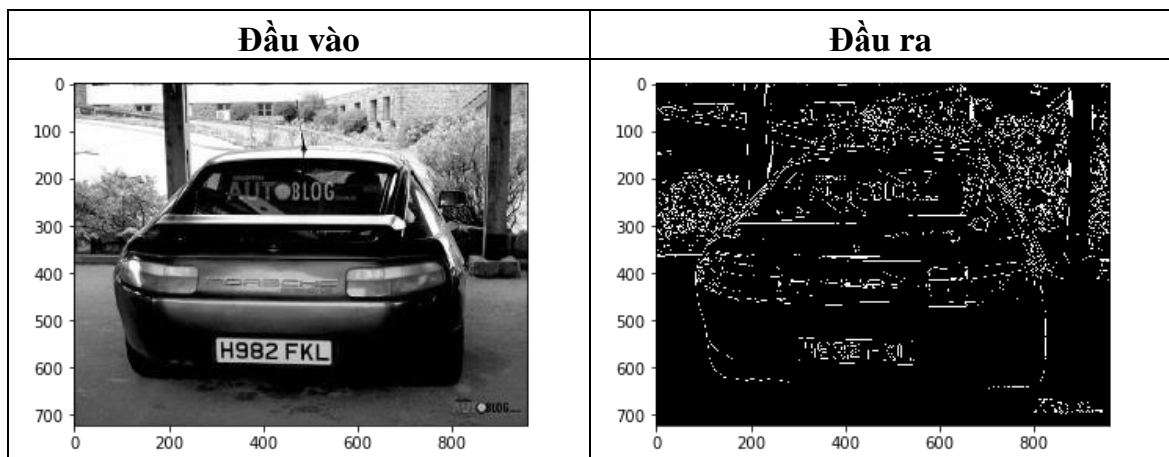
```
ret, threld = cv2.threshold(gray,127,255,cv2.THRESH_OTSU)
plt.imshow(cv2.cvtColor(threld, cv2.COLOR_BGR2RGB))
```

Với lệnh `ret, threld = cv2.threshold(gray,0,255,cv2.THRESH_OTSU)` ta đi áp dụng thuật toán Otsu trong việc phát hiện ngưỡng tự động và nhị phân hóa ảnh để thực hiện cho bước sau. Và hình ảnh cho ra kết quả:



Còn khi ta sử dụng thuật toán Canny để nhận diện biên thì sẽ cho ra các đường biên của ảnh. Nếu như kết hợp với bộ lọc hai chiều BilateralFilter thì ảnh ra sẽ cho ra nhận diện đường biên tốt hơn.

```
bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Noise reduction
edged = cv2.Canny(bfilter, 30, 200) #Edge detection
plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
```



4.4 Tìm contour và ghép vào ma trận zero

```
Keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE,  
cv2.CHAIN_APPROX_SIMPLE)  
contours = imutils.grab_contours(keypoints)  
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]
```

CV_RETR_TREE : khi sử dụng cờ này nó lấy tất cả các đường biên và tạo ra một hệ thống phân cấp đầy đủ của những đường lồng nhau.

CV_CHAIN_APPROX_SIMPLE : Ví dụ : một hình chữ nhật sẽ được mã hoá bằng toạ độ của 4 đỉnh.

Lệnh cv2.findContours() lọc contour theo diện tích các contours chỉ lấy 10 contour có giá trị lớn nhất(tránh lấy nhiều vì sẽ có nhiều)

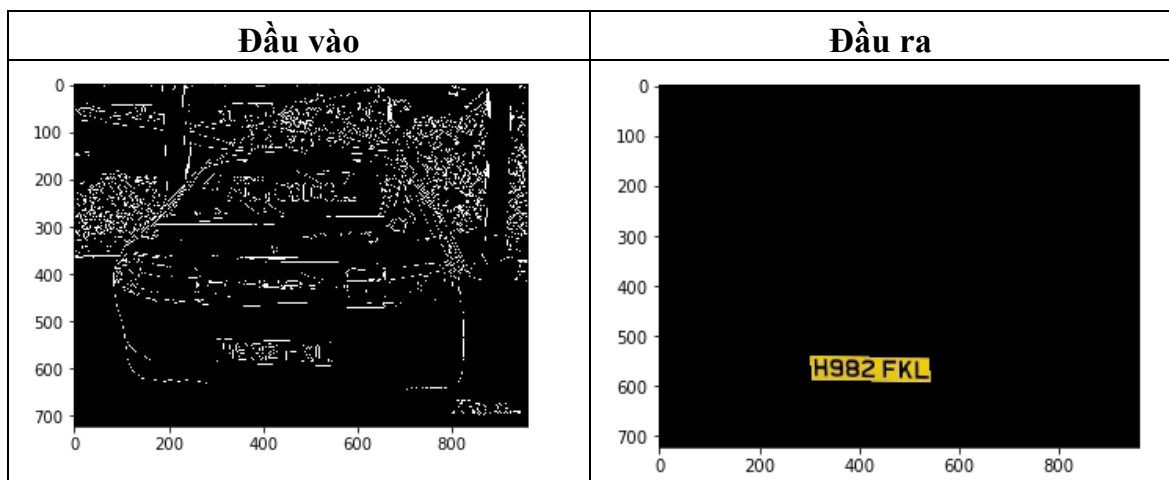
```
location = None  
for contour in contours:  
    approx = cv2.approxPolyDP(contour, 5, True)  
    if len(approx) == 4:  
        location = approx  
        break
```

cv2.approxPolyDP để xấp xỉ đa giác ở đây ta cần tìm là hình chữ nhật nên ta chỉ giữ lại contour nào có 4 cạnh và tách contour ra khỏi ảnh.

```
mask = np.zeros(gray.shape, np.uint8)  
new_image = cv2.drawContours(mask, [location], 0, 255, -1)  
new_image = cv2.bitwise_and(img, img, mask=mask)
```

Ta đi tạo một ma trận zero và thêm contour ta đã giữ lại ở trên vào. Như vậy ta đã có một ảnh mới, ở đây là biển số xe. Ta sẽ xem kết quả với dòng lệnh sau:

```
plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))
```


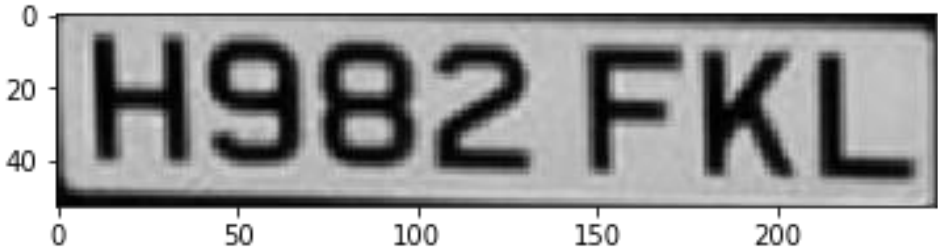


Ta tiếp tục đi loại bỏ nền điểm ảnh đen không cần thiết để còn lại ảnh biển số xe.

```
(x,y) = np.where(mask==255)
(x1, y1) = (np.min(x), np.min(y))
(x2, y2) = (np.max(x), np.max(y))
cropped_image = gray[x1:x2+1, y1:y2+1]
```

Ảnh sau khi được tách ra khỏi nền.

```
plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
```

Đầu vào	
Đầu ra	

Vậy ta đã hoàn thành xong bước tách biển số xe ra khỏi nền.

4.5 Áp dụng công thức hệ số tương quan Person

```
from PIL import Image
import numpy as np
img1 = Image.open("01.png").convert("L")
im1 = np.array(img1)/255

img2 = Image.open("0.png").convert("L")
im2 = np.array(img2)/255


def correlationCoefficient(X, Y):
    n = X.size
```

```

sum_X = X.sum()
sum_Y = Y.sum()
sum_XY = (X*Y).sum()
squareSum_X = (X*X).sum()
squareSum_Y = (Y*Y).sum()
corr = (n * sum_XY - sum_X * sum_Y)/(np.sqrt((n * squareSum_X -
sum_X * sum_X)*(n * squareSum_Y - sum_Y * sum_Y)))
return corr

a = im1.sum()
b = im2.sum()
c = (im1*im2).sum()
d = (im1*im1).sum()
e = (im2*im2).sum()
print(a)
print(b)
print(c)
print(d)
print(e)
print('{0:.6f}'.format(correlationCoefficient(im1, im2)))

```

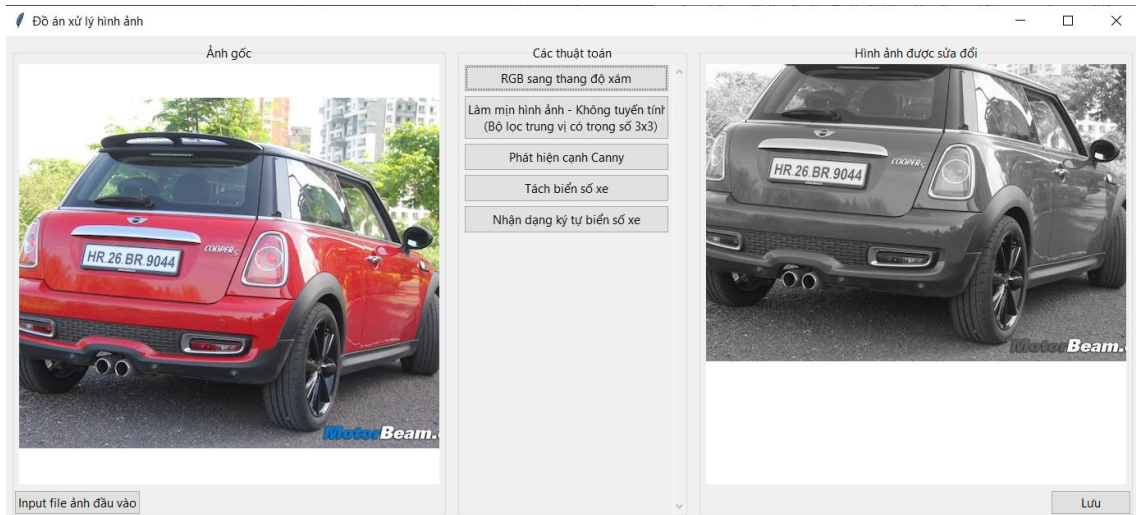
Đầu vào	
Đầu ra	sum_X: 14042.482352941177 sum_Y: 11196.082352941177 sum_XY: 10807.474509803922 squareSum_X: 14009.783175701652 squareSum_Y: 11142.103883121876 r-pearson: 0.461461

Với việc áp dụng thuật toán Otsu hay Canny và chỉ số tương quan Person (nhận dạng ký tự quang học) trong nhận diện biển số xe. Có thể thấy bài toán nhận diện biển số xe ta vẫn cần phải cải tiến thêm về thuật toán, mô hình huấn luyện để có kết quả tốt nhất có thể. Ứng dụng trong cuộc sống giúp ích cho các ngành nghề.

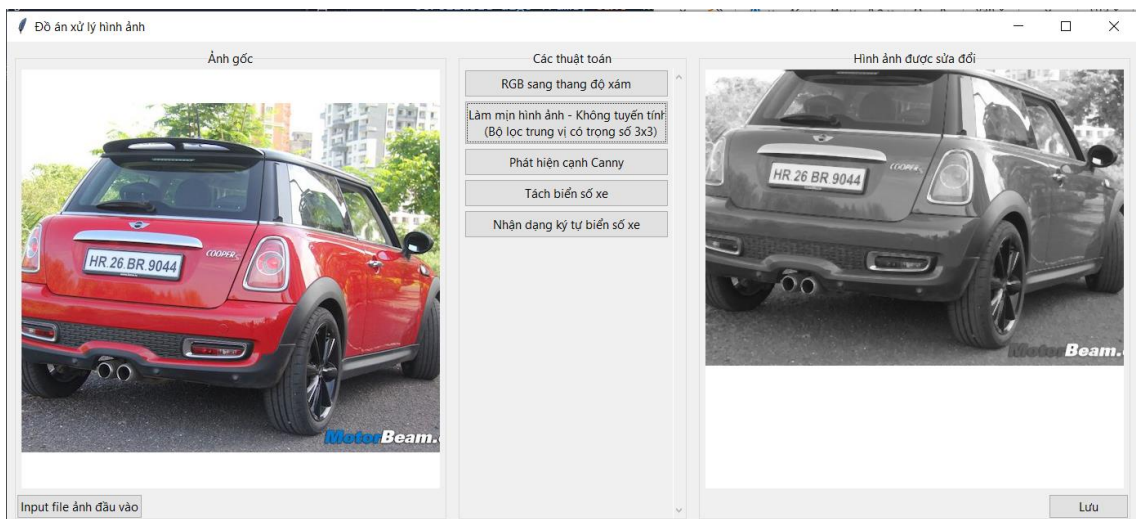
CHƯƠNG 5. GIAO DIỆN TKINTER

Tkinter là thư viện GUI tiêu chuẩn cho Python. Tkinter trong Python cung cấp một cách nhanh chóng và dễ dàng để tạo các ứng dụng GUI. Tkinter cung cấp giao diện hướng đối tượng cho bộ công cụ Tk GUI.

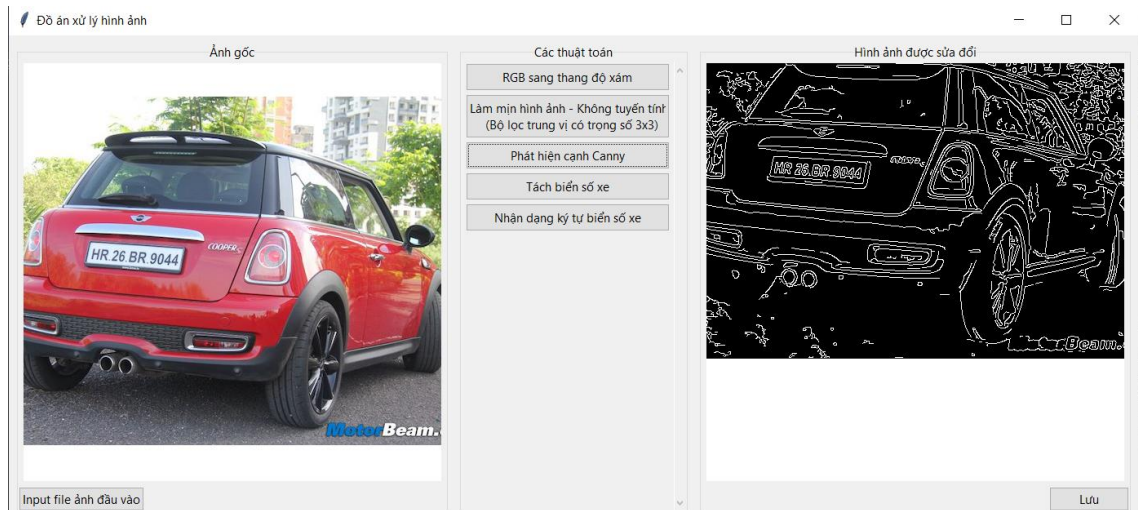
5.1 RGB sang thang độ xám



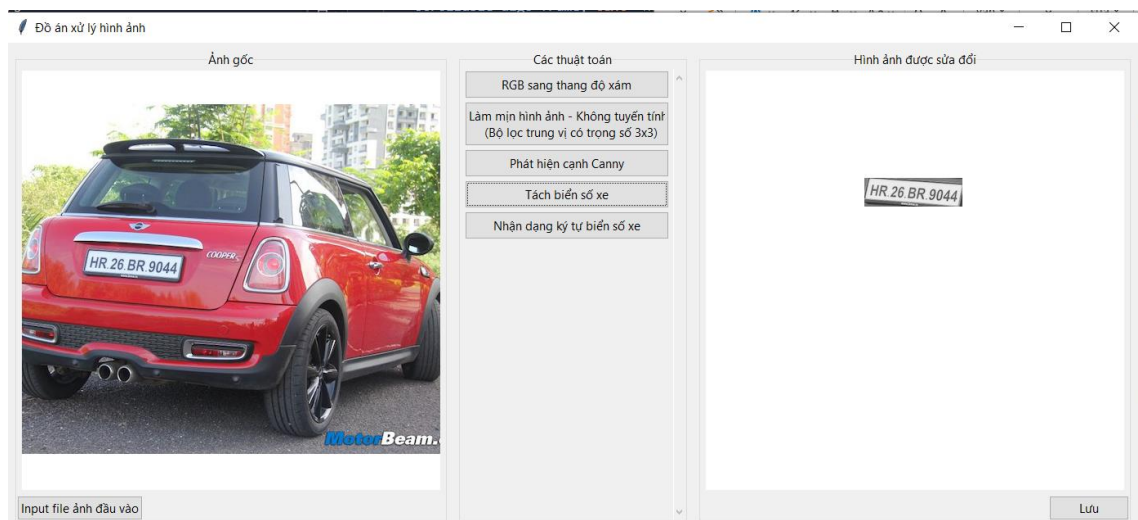
5.2 Làm mịn hình ảnh – Không tuyến tính (Bộ lọc trung vị có trọng số 3x3)



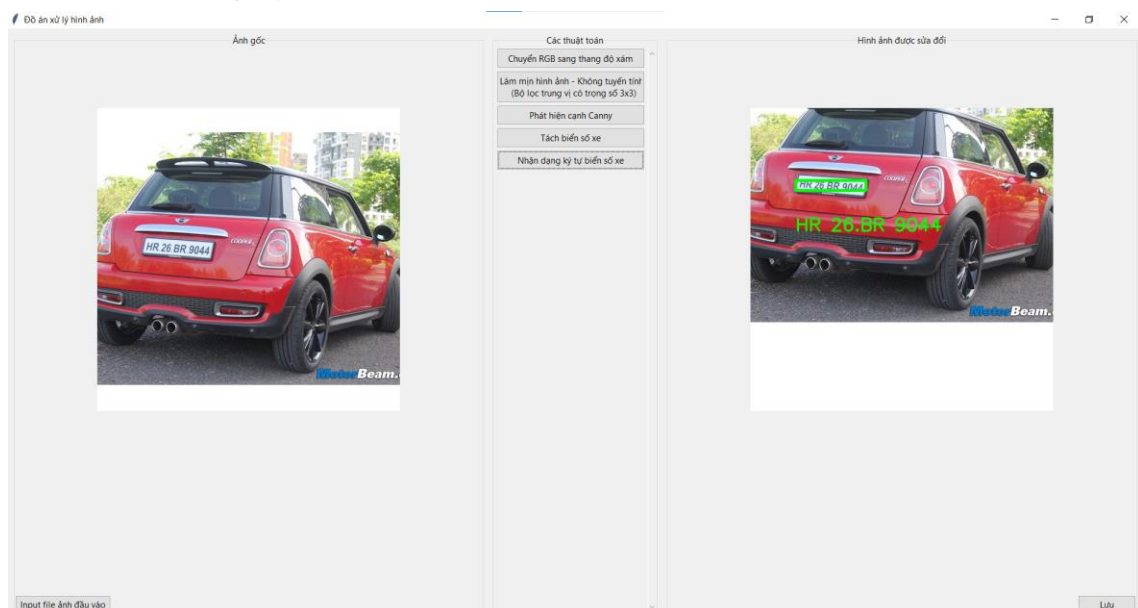
5.3 Phát hiện cạnh Canny



5.4 Tách biển số xe



5.5 Nhận dạng ký tự biển số xe



TÀI LIỆU THAM KHẢO

- [1] TheAILearner, “Optical Character Recognition: Introduction and its Applications”, Available online: <https://theailearner.com/2021/03/10/optical-character-recognition-introduction-and-its-applications/> (Accessed on 04 May 2021).
- [2] TheAILearner, “Optical Character Recognition Pipeline”, Available online: <https://theailearner.com/2019/05/28/optical-character-recognition-pipeline/> (Accessed on 04 May 2021).
- [3] Chris Solomon, Toby Breckon - Fundamentals of Digital Image Processing_ A Practical Approach with Examples in Matlab (2011, Wiley)
- [4] Digital image processing by Gonzalez, Rafael C. Woods, Richard E.