ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

BÁO CÁO BÀI TẬP LỚN

MÔN XỬ LÝ ẢNH

ĐỀ TÀI:

HÌNH THÁI HỌC - MORPHOLOGY

GVHD: Võ Tuấn Kiệt

Nhóm 10:

Họ Và Tên	MSSV	Công việc
Nguyễn Văn Thăng	1910543	Phần 2 + Mô phỏng
Cao Văn Lên	1910301	Phần 1 + Mô phỏng
Vương Thành Tín	1811280	Phần 3 + Mô Phỏng

MỤC LỤC

1. Kiến thức nền

1.1 Ảnh số - điểm ảnh

Ảnh số là một dạng dữ liệu được dùng để giao tiếp trên máy tính mà trong đó thông tin được thể hiện dưới dạng màu sắc khác nhau. Bằng cách thuật toán hóa, ảnh số được biểu diễn dưới dạng ma trận cụ thể là bằng cách lấy mẫu đều trong không gian và lượng tử hóa giá trị độ sáng G mà tạo thành. Trong đó, mỗi phần tử trong ma trận được gọi là một điểm ảnh (hay pixel).

Cùng một ảnh nhưng chất lượng ảnh khác nhau tùy thuộc vào độ phân giải không gian - số mẫu mà ta lấy trong quá trình lấy mẫu, cũng như độ phân giải độ sáng - số lượng mức xám G mà ta gán trong quá trình lượng tử hóa, thông thường

$$G = 2^m$$

m là độ sâu ảnh bit - bit depth hay được hiểu là số lượng bit dùng để lưu trữ một mức xám. Với m = 1 ta được ảnh nhị phân. m = 8 bit ta được ảnh xám và m = 24 ta có được ảnh màu thật.

Hình 1: Ảnh màu - Ảnh xám - Ảnh trắng đen

1.2 Hình thái học trong xử lý ảnh - Morphology

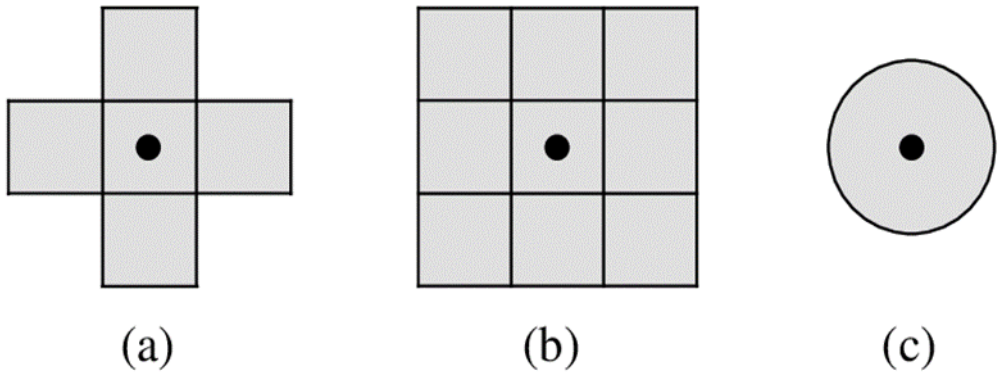
Hình thái học được hiểu là một phương pháp xác định, phân tích và miêu tả cấu trúc hình thái của một đối tượng. Trong xử lý ảnh, xử lý ảnh hình thái học là phương pháp dùng để xử lý ảnh dựa theo hình dáng và cấu trúc của ảnh, giúp làm đẹp câu trúc và

hình thức của ảnh nhị phân, ngoài ra các phép toán này cũng được sử dụng cho cả ảnh xám.

Trong phương pháp xử lý này, mỗi pixel của ảnh đầu ra được xác định qua phép so sánh mỗi pixel tương ứng ở ảnh đầu vào với các pixel lân cận của chúng. Bằng cách lựa chọn kích thước và hình dạng của các pixel lân cận, ta có thể xây dựng các phép toán xử lý hình thái - Morphology Operations (MO) - chuyên biệt khác nhau cho các mục đích khác nhau.

Cụ thể, các phép xử lý hình thái học này phụ thuộc vào một phần tử được gọi là phần tử cấu trúc - Structuring elements (SE), hay còn gọi là Kernel.

Phần tử cấu trúc SE được định nghĩa là một ảnh nhị phân dưới dạng một ma trận có kích nhỏ mang các giá trị 1 và 0, và luôn có một pixel được chọn làm mốc.



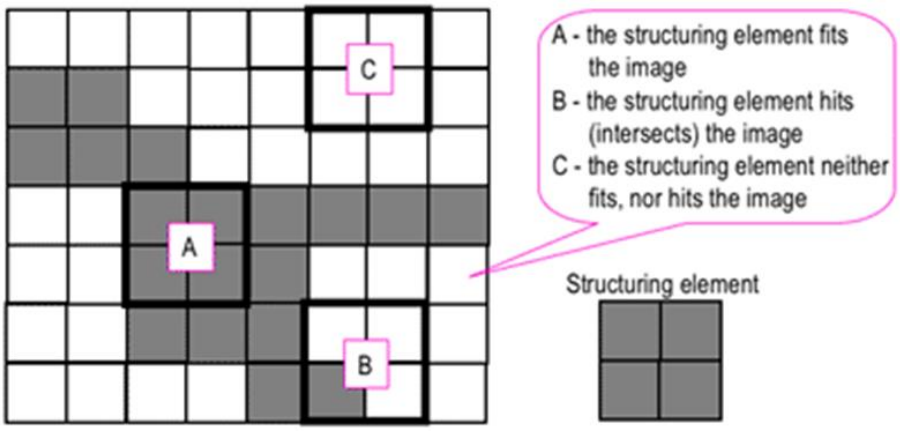
Hình 2: *Một số phần tử cấu trúc Kernel thường được sử dụng trong các phép toán xử lý hình thái học; các điểm bôi đen trên hình được gọi là mốc của SE*

Trong quá trình thực hiện các phép toán hình thái học, SE sẽ lần lượt quét qua toàn bộ các pixel trong ảnh đầu vào, tiến hành so sánh với các pixel lân cận rồi từ đó cho ra kết quả là giá trị mức xám tương ứng cho từng pixel ảnh đầu ra.

Trong quá trình SE quét lần lượt qua hết tất cả các pixel trên ảnh, sẽ có ba trường hợp xảy ra:

- **Fit**: trong trường hợp này các phần tử trong SE trùng hoàn toàn với các phần tử ảnh.
- **Hit**: Fit: trong trường hợp này có ít nhất một phần tử trong SE trùng với các phần tử ảnh.

- **Miss**: trong trường hợp này sẽ không có bất kỳ sự tương đồng nào giữa phần tử



trong SE và các phần tử ảnh.

Hình 3: *(A) Fit - (B) Hit - (C) Miss*

1.3 Tập hợp - Các phép toán tập hợp

Trong toán học, tập hợp được hiểu là một tập gồm 2 hay nhiều đối tượng nào đó, mà các đối tượng đó được gọi là các phần tử và có thể được biểu diễn dưới dạng liệt kê hoặc ký hiệu.

Ví dụ: trong xử lý ảnh giả xử trong một bức ảnh nhị phân, gọi A là tập hợp các điểm ảnh với tọa độ (x; y), sao cho các giá trị tại điểm ảnh đó bằng 1. Khi đó, tập hợp A sẽ được biểu diễn như sau:

$$A = \{ (x; y) \mid I_A (x; y) = 1 \}$$

Các phép toán tập hợp cơ bản:

Phép hợp: là phép toán giữa hai hay nhiều tập hợp với nhau, mà kết quả nhận được là tập hợp có phần tử thuộc một hay nhiều các tập hợp thành phần. kí hiệu ∪ và được biểu diễn dưới dạng tập hợp là:

$$A \cup B = \{ x \mid x \in A \text{ hoặc } x \in B \}$$

Phép giao: là phép toán giữa hai hay nhiều tập hợp với nhau, mà kết quả nhận được là tập hợp có phần tử đều thuộc các tập hợp thành phần. kí hiệu ∩ và được biểu diễn dưới dạng tập hợp là:

$$A \cap B = \{ x \mid x \in A \text{ và } x \in B \}$$

Phép bù: là phép toán cho biết các phần tử không thuộc tập hợp và được biểu diễn dưới dạng:

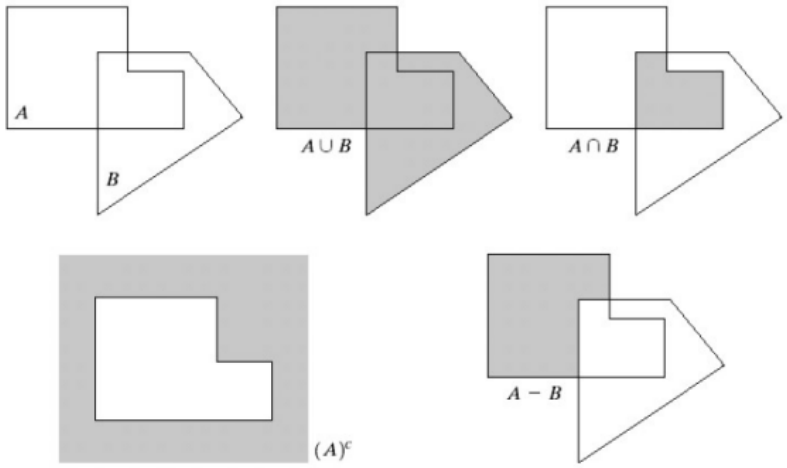
$$A^C = \{x \mid x \notin A\}$$

Phép hiệu: là phép toán giữa các tập hợp, ví dụ ta lấy hiệu giữa hai tập hợp A và B thì kết quả sẽ được một tập hợp chỉ chứa các phần tử thuộc A mà không thuộc B, ký hiệu và được biểu diễn dưới dạng:

$$A - B = \{ x \mid x \in A \text{ và } x \notin B \}$$

Tập con: A được gọi là tập con của B nếu mỗi phần tử trong tập hợp A đều thuộc tập hợp B, ký hiệu $\subset$

Tập rỗng: là tập hợp không chứa bất kỳ phần tử nào, ký hiệu Φ



Hình 4: Hình ảnh minh họa cho các phép toán tập hợp

2. Các phép toán hình thái học căn bản trong xử lý ảnh

Những hàm hình thái học áp dụng các phần tử cấu trúc lên trên ảnh đầu vào, và cho ảnh đầu ra với cùng kích thước. Giá trị của ảnh đầu ra được quyết định dựa trên sự so sánh giữa giá trị pixel tương ứng trên ảnh với những pixel lân cận.

2.1 Xói mòn (erosion) và giãn nở (dilation)

2.1.1 Giãn nở (Dilation)

Sự giãn nở của tập hợp A bởi phần tử cấu trúc B được biểu diễn :

$$A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \emptyset \}$$

Cách làm : Lấy đối xứng B, dịch phần tử cấu trúc B theo z, nếu B có trùng lấp với A thì đánh số 1 tại vị trí tương ứng trọng tâm của B

Ảnh gốc *Ảnh được giản nở*

Tính chất:

- Giao hoán: $A \oplus B = B \oplus A$
- Kết hợp: $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
- Tính đối ngẫu giữa xói mòn và giản nở: $(A \ominus B)^c = A^c \oplus \hat{B}$

2.1.2 Xói mòn (erosion)

Ngược lại với phép giãn nở là xói mòn.

Sự xói mòn của tập hợp A bởi phần tử cấu trúc B được biểu diễn :

$A \ominus B = \{ z \mid (B)_z \subseteq A \}$

$$A \ominus B \neq B \ominus A$$

Cách làm: Dịch phần tử cấu trúc B theo z, nếu B nằm hoàn toàn trong A thì đánh số 1 vị trí tương ứng trọng tâm của B.

Ảnh gốc Ảnh được xói mòn

2.1.3 Ứng dụng phát hiện biên.

Thức hiện các phép toàn giãn nở và xói mòn cho phép ta thay đổi kích thước của vật thể trong ảnh nhị phân. Thực hiện phép trừ giữa A, $A \oplus B, v\text{à } A \ominus B$ cho ta đường bao của vật thể. Ta có thể sử dụng các phép trừ sau:

$A - (A \ominus B)$ $(A \oplus B) - A$ $(A \oplus B) - (A \ominus B)$

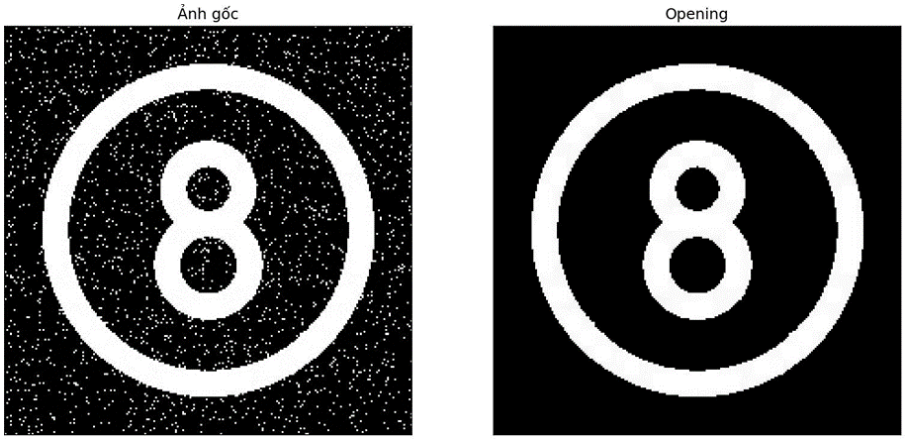
ví dụ : $A - (A \ominus B)$

2.2 Opening and closing

2.2.1 Opening

Là thực hiện phép co rồi bắt đầu thực hiện phép giãn với một cấu trúc. Giúp làm mượt các đường viền, phá vỡ các khe nhỏ, loại bỏ các đối tượng nhỏ, làm mượt các đỉnh lồi.

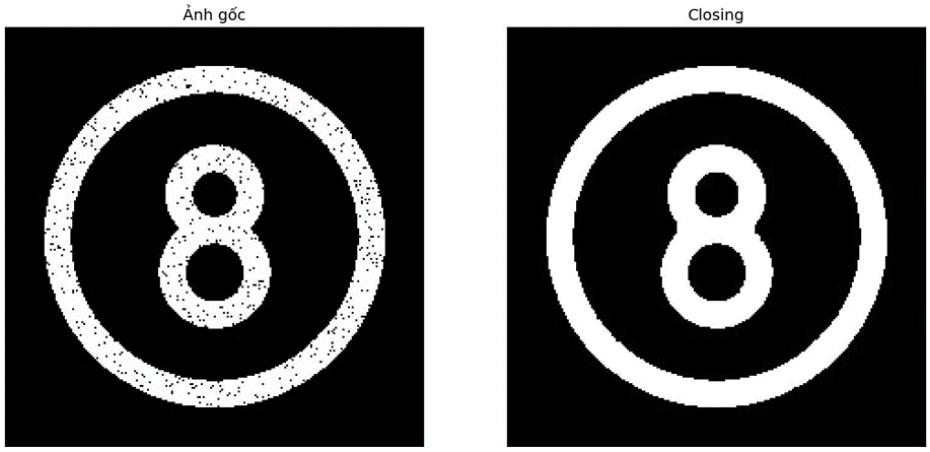
Công thức: $A \circ B = (A \ominus B) \oplus B$



2.2.2 Closing

Là thực hiện phép giãn rồi bắt đầu thực hiện phép co với một cấu trúc. Giúp làm mượt các đường viền, loại bỏ các lỗ nhỏ, làm mượt các đỉnh khe hẹp.

Công thức: $A \bullet B = (A \oplus B) \ominus B$



2.2.3 Ứng dụng khử nhiễu

Sử dụng Opening và Closing để khử nhiễu như sau. Giả sử có 1 ảnh nhị phân bị nhiễu, một vài điểm ảnh trắng bị đổi thành đen và ngược lại. Để khử nhiễu, ta thực hiện Closing và Openning như sau, với mặt nạ B cho trước:

$$(A \circ B) \bullet B = \Big(\big((A \ominus B) \oplus B\big) \oplus B\Big) \ominus B$$

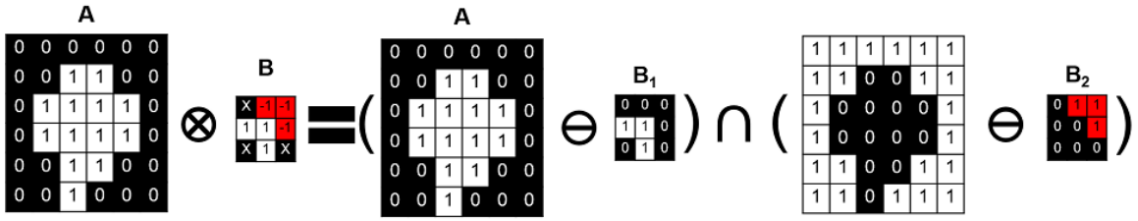
Cách này gọi là Morphological Filtering. Đầu tiên, $(A \ominus B)$ sẽ loại bỏ các điểm ảnh đen riêng biệt, tuy nhiên nó sẽ tạo ra các lỗ. Ta sẽ lấp các lỗ đó bằng cách giãn nở 2 lần. Lúc này, kích thước vật thể bị phình to ra nên ta cần xói mòn thêm 1 lần.

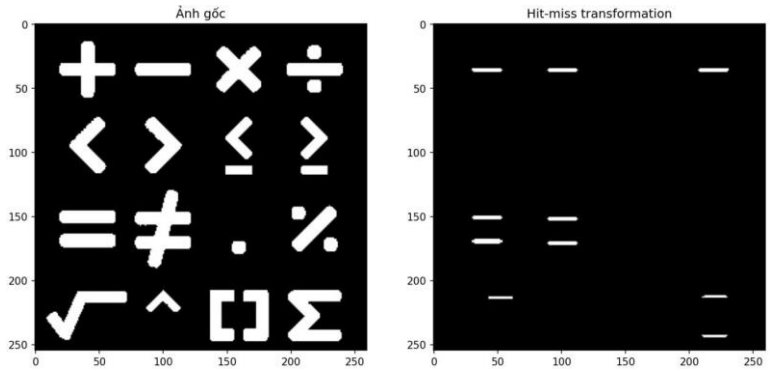
Morphological Filtering

2.2.4 The hit-or-miss transform

Phép hit-or-miss transform được thực hiện giống như những phép toán hình thái ở trên, bằng cách dịch chuyển trọng tâm của phần tử cấu trúc đến tất cả các điểm pixels của ảnh, và so sánh phần tử cấu trúc với các pixels nằm dưới của ảnh. Nếu giá trị những pixels của phần tử cấu trúc khớp chính xác các giá trị pixels nằm dưới trong hình, pixel trên ảnh nằm ngay dưới trọng tâm của phần tử cấu trúc sẽ được gán bằng 1, ngược lại sẽ gán bằng 0.

Công thức: $$A \otimes B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

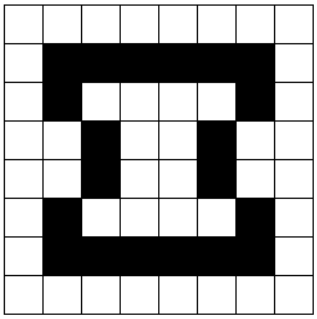




3. Các thuật toán hình thái học phổ biến

3.1 Làm đầy vùng (Region filling/Hole filling)

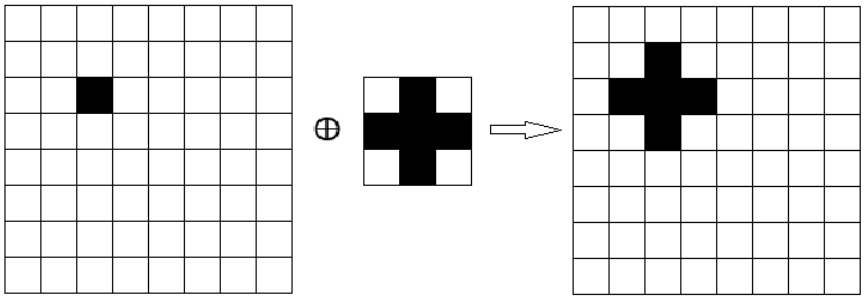
Làm đầy vùng của một đối tượng từ biên là bổ sung giá trị 1 vào toàn bộ vùng ở phía bên trong biên của đối tượng. Cho một ảnh nhị phân A, các điểm ảnh ở biên có giá trị là 1 và các điểm ảnh khác có giá trị là 0.



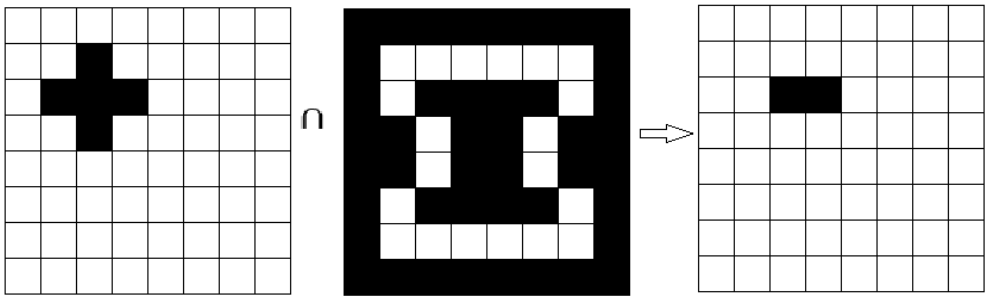
Ảnh nhị phân và vùng biên cho trước

Ta bắt đầu thuật toán bằng việc gán giá trị 1 vào điểm ảnh p ở bên trong biên, sau đó lặp lại phép toán giãn nở với phần tử cấu trúc B thường là ma trận 3x3 dạng chữ thập.

Phần tử cấu trúc B



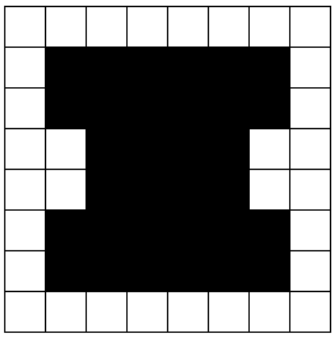
Giới hạn được đặt ra là phép bù với tập hợp A (A^C), vì nếu không có việc hạn chế này thì dẫn tới việc tràn qua các vùng khác trên ảnh, thậm chí toàn bộ ảnh.



Với $X_0 = p$ và B là phần tử cấu trúc, thuật toán làm đầy vùng được minh họa bởi công thức:

$$X_k = (X_{k-1} \oplus B) \cap A^C$$

Với k = 1, 2, 3, …, k-1. Thuật toán kết thúc khi $X_k = X_{k-1}$ hay tất cả điểm ảnh p bên trong biên đối tượng được gán giá trị là 1.

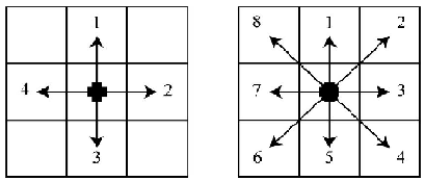


Kết quả thu được sau thuật toán

3.2 Trích chọn các thành phần liên kết (Extraction of Connected Components)

Mục đích của thuật toán là xác định các pixel có thuộc cùng một nhóm có liên kết với nhau.

Path (đường đi): đường đi từ pixel $[i_0, j_0]$ đến pixel $[i_n, j_n]$ là sự liên tiếp của các pixel $[i_1, j_1]$, $[i_2, j_2]$,…với pixel $[i_k, j_k]$ là lân cận của pixel $[i_{k+1}, j_{k+1}]$. Trong đó lân cận được xác định phổ biến thông qua 4-connection và 8 connection.

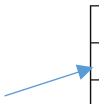


4-connection 8-connection

Nếu tồn tại đường đi như trên, ta nói pixel $[i_0, j_0]$ liên kết với pixel $[i_n, j_n]$ ở phần tiền cảnh (foreground) và chuỗi các pixel lận cận nhau như trên được gọi là các thành phần liền kết (connected components).

Thuật toán được thực hiện như sau:

Quét hàng theo thứ tự từ trái qua phải và từ trên xuống dưới, tìm pixel tiền cảnh đầu tiên có giá trị là 1.



0	0	0	0	0	0	0	0
0	1	1	0	1	1	1	0
0	1	1	0	0	0	1	0
0	0	1	0	0	1	1	0
0	0	1	0	0	1	1	0
0	0	1	0	0	1	1	0
0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0

Ảnh sau lấy ngưỡng

Nếu giá trị pixel lân cận bên trên hoặc bên trái của nó không có nhãn thì đánh dấu nhãn mới, di chuyển đến pixel kế tiếp.

0	**0**	0	0	0	0	0	0
0	**1**	1	0	1	1	1	0
0	1	1	0	0	0	1	0
0	0	1	0	0	1	1	0
0	0	1	0	0	1	1	0
0	0	1	0	0	1	1	0
0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0

Pixel được đánh nhãn 1

Nếu giá trị pixel lân cận bên trên hoặc bên trái của nó có nhãn và các nhãn là giống nhau thì sao chép nhãn đó, di chuyển đến nhãn tiếp theo.

0	0	**0**	0	0	0	0	0
0	**1**	1	0	1	1	1	0
0	1	1	0	0	0	1	0
0	0	1	0	0	1	1	0
0	0	1	0	0	1	1	0
0	0	1	0	0	1	1	0
0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0

(1)

0	0	0	0	**0**	0	0	0
0	**1**	**1**	0	1	1	1	0
0	1	1	0	0	0	1	0
0	0	1	0	0	1	1	0
0	0	1	0	0	1	1	0
0	0	1	0	0	1	1	0
0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0

(2)

(1) Pixel đang xét được đánh dấu cùng nhãn bên trái

(2) Lân cận chưa có nhãn, đánh dấu nhãn mới 2

Trường hợp giá trị pixel lân cận bên trên và bên trái của nó có nhãn khác nhau, hãy sao chép nhãn có giá trị nhỏ hơn và nhập các nhãn vào bảng tương đương.

0	0	0	0	0	0	0	0
0	**1**	**1**	0	**2**	**2**	**2**	0
0	**1**	**1**	0	0	**2**	**2**	0
0	0	**1**	0	0	**2**	**2**	0
0	0	**1**	0	0	**2**	**2**	0
0	0	**1**	0	0	**2**	**2**	0
0	0	**1**	**1**	**1**	1	1	0
0	0	0	0	0	0	0	0

(1)

0	0	0	0	0	0	0	0
0	**1**	**1**	0	**2**	**2**	**2**	0
0	**1**	**1**	0	0	**2**	**2**	0
0	0	**1**	0	0	**2**	**2**	0
0	0	**1**	0	0	**2**	**2**	0
0	0	**1**	0	0	**2**	**2**	0
0	0	**1**	**1**	**1**	**1**	**1**	0
0	0	0	0	0	0	0	0

(2)

(1) Pixel đang xét sẽ được gắn nhãn là 1($1 \equiv 2$)

(2) Ảnh sau khi quét lần 1

Quét toàn bộ ảnh và thay thế giá trị nhỏ nhất cho các nhãn trong bảng tương đương.

0	0	0	0	0	0	0	0
0	**1**	**1**	0	**1**	**1**	**1**	0
0	**1**	**1**	0	0	**1**	**1**	0
0	0	**1**	0	0	**1**	**1**	0
0	0	**1**	0	0	**1**	**1**	0
0	0	**1**	0	0	**1**	**1**	0
0	0	**1**	**1**	**1**	**1**	**1**	0
0	0	0	0	0	0	0	0

Kết quả sau thuật toán

3.3 Thuật toán xác định khung xương (Skeletonization)

Mục đích của toán tử rút xương là trích xuất hình dáng đặc trưng đại diện chung cho đối tượng.

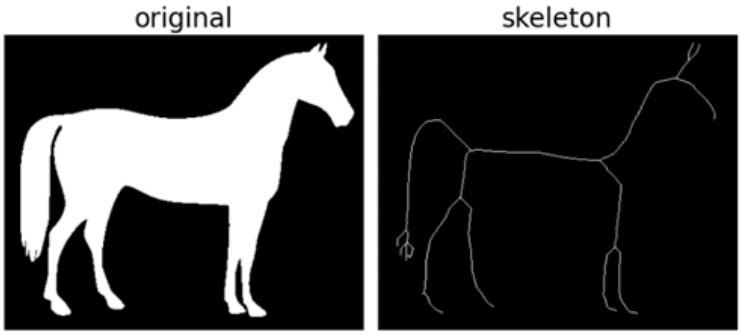
Toán tử rút xương có thể được biểu diễn bằng tập hữu hạn các phép co (erosion) và mở (opening).

$$S(A) = \bigcup_{k=0}^{k} S_k(A)$$

Với $S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$

B là phần tử cấu trúc và $(A \ominus kB)$ là kết quả của phép thực hiện co (erosion) k lần với B của A.

Quá trình sẽ dừng lại cho đến khi $(A \ominus kB) = \emptyset$, khi đó hợp tất cả các tập hợp đó lại sẽ định hình nên khung xương của vật thể.



TÀI LIỆU THAM KHẢO

[1] Rafael C. Gonzalez, Richard E. Wood, " Digital Image Processing - Four Edition", trang 635-674, 2018.

[2] Morphological Image Processing. *Truy cập*: *https://bom.so/uw3LSY*

[3] Skeleton Binary Morphology. *Truy cấp: https://bom.so/7tswWH*

[4] Filling holes in an image using OpenCV. *Truy cập: https://bom.so/mzbBRn*

[5] Tài liệu xử lý ảnh (Image Processing) cơ bản. *Truy cập: https://bom.so/zuZFwr*