

## LỜI NÓI ĐẦU

Cùng với các ngôn ngữ, các thông tin dưới dạng hình ảnh đóng một vai trò rất quan trọng trong công việc trao đổi thông tin. Chính vì vậy những năm gần đây đã có sự kết hợp rất chặt chẽ giữa ảnh và đồ họa trong lĩnh vực xử lý thông tin. Trong công nghệ thông tin, xử lý ảnh chứa một vai trò rất quan trọng, bởi các ứng dụng đa dạng và phong phú của nó trong nhiều lĩnh vực khoa học. Xử lý ảnh là một bộ phận quan trọng trong việc trao đổi thông tin giữa người và máy. Nó góp phần làm cho việc quan sát ảnh trở nên tốt hơn.

Các thao tác Hình thái học (Morphology) nói chung, đặc biệt là Hình thái học số được sử dụng chủ yếu vào việc cải thiện ảnh bằng cách làm rõ (tái hiện) những nét đặc trưng của các hình dạng, do vậy có thể tính toán được hay nhận biết được chúng một cách dễ dàng.

Đồ án này giới thiệu một số khái niệm về các thao tác Hình thái học, sử dụng các thao tác hình thái và ứng dụng của chúng.

**Đồ án bao gồm :**

**Chương 1:**Sơ lược về xử lý ảnh và Morphology.

Giới thiệu sơ bộ về xử lý ảnh và ứng dụng của nó.

**Chương 2 :**Thao tác với Morphology

Chương này là chương chính giới thiệu về các thao tác với ảnh nhị phân, ảnh đa cấp xám. Cụ thể đó là các thao tác như : Phép dẫn, phép co, phép đóng mở ảnh, đánh trống đánh trượt, dẫn theo điều kiện và kỹ thuật đếm vùng.Trong ảnh đa cấp xám, ta còn đề cập đến phép toán làm trơn ảnh, phương pháp gradient, cách phân vùng theo cấu trúc, cách phân loại cỡ đối tượng. Bên cạnh các thao tác có kèm theo ý nghĩa của chúng, có thuật toán và có hình minh họa.

**Chương 3:**Ứng dụng của Morphology

Trong chương này của Đồ án giới thiệu về ý nghĩa của hình thái học trong thực tiễn và các ứng dụng nói chung của thao tác hình thái. Đặc biệt, trong chương này có trình bày khá chi tiết một ứng dụng của phép toán hình thái có tính thiết thực.

#### **Chương 4:** Cài đặt.

Trình bày quá trình cài đặt chi tiết một số thao tác hình thái học.

## **CHƯƠNG I**

### **SƠ LƯỢC VỀ XỬ LÝ ẢNH VÀ PHÉP TOÁN HÌNH THÁI**

#### **1.1 Xử lý ảnh**

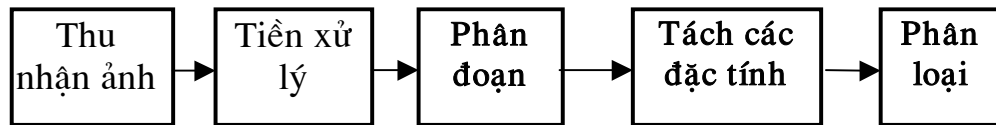
Cũng như xử lý dữ liệu bằng đồ họa, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo ra bởi các chương trình. Xử lý ảnh số bao gồm các phương pháp và kỹ thuật để biến đổi, để truyền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

- ✚ Thứ nhất, biến đổi ảnh và làm đẹp ảnh.
- ✚ Thứ hai, tự động nhận dạng ảnh hay đoán nhận ảnh và đánh giá các nội dung của ảnh.

Nhận biết và đánh giá các nội dung của ảnh (nhận dạng ) là sự phân tích một hình ảnh thành những phần có 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 biên của một đối tượng trên ảnh, tách cạnh, phân đoạn hình ảnh, v.v... Kỹ thuật này được dùng nhiều trong y học (xử lý tế bào, nhiễm sắc thể), nhận dạng chữ trong văn bản.

#### **1.2. Các quá trình của xử lý ảnh**

Các quá trình của xử lý ảnh được tiến hành theo sơ đồ sau:



Hình 1: Sơ đồ quá trình xử lý ảnh

**Thu nhận ảnh:** Đây là công đoạn đầu tiên mang tính quyết định đối với quá trình XLA. Ảnh đầu vào sẽ được thu nhận qua các thiết bị như camera, sensor, máy scanner, vv ... và sau đó các tín hiệu này sẽ được số hoá. Việc lựa chọn thiết bị thu nhận ảnh sẽ phụ thuộc vào đặc tính của các đối tượng cần xử lý. Các thông số quan trọng ở bước này là độ phân giải, chất lượng màu, dung lượng bộ nhớ và tốc độ thu nhận ảnh của các thiết bị.

**Tiền xử lý:** Ở bước này, ảnh sẽ được cải thiện về độ tương phản, khử nhiễu, khử bóng, khử độ lệch, v.v.. với mục đích làm cho chất lượng ảnh trở nên tốt hơn nữa, chuẩn bị cho các bước xử lý phức tạp hơn về sau trong quá trình XLA. Quá trình này thường được thực hiện bởi các bộ lọc.

**Phân đoạn ảnh:** Phân đoạn ảnh là *bước then chốt* trong XLA. Giai đoạn này nhằm phân tích ảnh thành những thành phần có cùng tính chất nào đó dựa theo biên hay các vùng liên thông. Tiêu chuẩn để xác định các vùng liên thông có thể là cùng màu, cùng mức xám hay cùng độ nhám vv ... Mục đích của phân đoạn ảnh là để có một miêu tả tổng hợp về nhiều phần tử khác nhau cấu tạo nên ảnh thô. Vì lượng thông tin chứa trong ảnh rất lớn – trong khi trong đa số các ứng dụng chúng ta chỉ cần trích chọn một vài đặc trưng nào đó, do vậy cần có một quá trình để giảm lượng thông tin không lờ ấy. Quá trình này bao gồm ***phân vùng ảnh*** và ***trích chọn đặc tính*** chủ yếu.

**Tách các đặc tính:** Kết quả của bước phân đoạn ảnh thường được cho dưới dạng dữ liệu điểm ảnh thô, trong đó hàm chứa biên của một vùng ảnh, hoặc tập hợp tất cả các điểm ảnh thuộc về chính vùng ảnh đó. Trong cả hai trường hợp, sự chuyển đổi dữ liệu thô này thành một dạng thích hợp hơn cho việc xử lý trong máy tính là rất cần thiết. Để chuyển đổi chúng, câu hỏi đầu tiên cần phải trả lời là nên biểu diễn một vùng ảnh *dưới dạng biên* hay *dưới dạng một vùng hoàn chỉnh* gồm tất cả những điểm

ảnh thuộc về nó. Biểu diễn dạng biên cho một vùng phù hợp với những ứng dụng chỉ quan tâm chủ yếu đến các đặc trưng hình dạng bên ngoài của đối tượng, ví dụ như các góc cạnh và điểm uốn trên biên chẳng hạn. Biểu diễn dạng vùng lại thích hợp cho những ứng dụng khai thác các tính chất bên trong của đối tượng, ví dụ như vân ảnh hoặc cấu trúc xương của nó. Sự chọn lựa cách biểu diễn thích hợp cho một vùng ảnh chỉ mới là một phần trong việc chuyển đổi dữ liệu ảnh thô sang một dạng thích hợp hơn cho các xử lý về sau. Chúng ta còn phải đưa ra một phương pháp mô tả dữ liệu đã được chuyển đổi đó sao cho những tính chất cần quan tâm đến sẽ được làm nổi bật lên, thuận tiện cho việc xử lý chúng.

**Phân loại :** Đây là bước cuối cùng trong quá trình XLA. Nhận dạng ảnh (image recognition) có thể được nhìn nhận một cách đơn giản là việc *gán nhãn* cho các đối tượng trong ảnh. Ví dụ đối với nhận dạng chữ viết, các đối tượng trong ảnh cần nhận dạng là các mẫu chữ, ta cần tách riêng các mẫu chữ đó ra và tìm cách gán đúng các ký tự của bảng chữ cái tương ứng cho các mẫu chữ thu được trong ảnh. Giải thích là công đoạn *gán nghĩa* cho một tập các đối tượng đã được nhận biết.

### 1.3. Khái niệm về phép toán hình thái MORPHOLOGY

Hiểu một cách đầy đủ thì ” Morphology ” là hình thái và cấu trúc của đối tượng, hay nó diễn tả những phạm vi và các mối quan hệ giữa các phần của một đối tượng. Hình thái học quá quen thuộc trong các lĩnh vực ngôn ngữ học và sinh học. Trong ngôn ngữ học, hình thái học là sự nghiên cứu về cấu trúc của từ, tập hợp từ, câu... và đó cũng là một lĩnh vực nghiên cứu từ nhiều năm nay. Còn trong sinh học, Hình thái học lại chú trọng tới hình dạng của một cá thể hơn, chẳng hạn có thể phân tích hình dạng của một chiếc lá để từ đó có thể nhận dạng được loại cây đó là cây gì; nghiên cứu hình dạng của một nhóm vi khuẩn, dựa trên các đặc điểm nhận dạng để phân biệt chúng thuộc nhóm vi khuẩn nào, v.v... Tùy theo trường hợp cụ thể mà có một cách phân lớp phù hợp với nó: Có thể phân lớp dựa trên những hình dạng bao quanh như (elip, tròn,...), kiểu và mức độ của những hình dạng bất quy tắc (lồi, lõm,...), những cấu trúc trong (lỗ, đường thẳng, đường cong,...) mà đã được tích lũy qua nhiều năm quan sát.

Tính khoa học của Hình thái học số chỉ mới thực sự phát huy khả năng của nó kể từ khi máy tính điện tử số ra đời và đã làm cho Hình thái học trở nên thông dụng, có

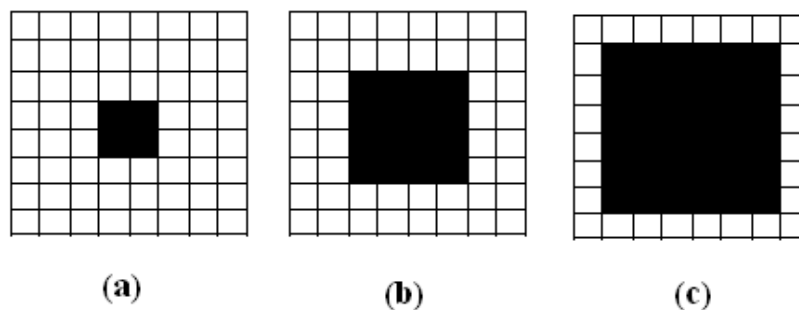
nhiều tính năng mới. Những đối tượng ảnh trong Hình thái học hầu như, ta có thể coi hầu như là tập hợp của các điểm ảnh, nhóm lại theo cấu trúc 2 chiều. Những thao tác toán học cụ thể trên tập hợp điểm đó được sử dụng để làm rõ (tái hiện) những nét đặc trưng của những hình dạng, do vậy mà có thể tính toán được hay nhận biết được chúng một cách dễ dàng.

## CHƯƠNG II

### THAO TÁC VỚI HÌNH THÁI HỌC

#### 2.1. Thao tác trên ảnh nhị phân

Những thao tác hình thái nhị phân được xây dựng trên ảnh có 2 cấp xám bao gồm chỉ những điểm ảnh, ta kí hiệu đen (1) hoặc trắng (0). Trước hết, để bắt đầu, ta hãy xem hình 2.1a. Tập hợp các điểm ảnh đen tạo nên đối tượng ảnh hình vuông và trong 2.1b, đối tượng ảnh cũng là hình vuông nhưng là hình vuông lớn hơn so với 2.1a 1 điểm ảnh về mọi phía, nghĩa là thay mọi lân cận trắng của các điểm ảnh trong 2.1a thành các điểm ảnh đen. Đối tượng trong 2.1c cũng được thao tác tương tự, tức là 2.1b được tăng thêm 1 điểm ảnh về mọi phía. Thao tác đó có thể coi như một phép dẫn đơn giản, phép dẫn một điểm ảnh về mọi phía. Việc dẫn đó có thể được thực hiện cho đến khi toàn bộ ảnh được thay bằng các điểm ảnh đen. Tuy nhiên trong thực tế, đối tượng ảnh được xem như là một tập hợp toán học của các điểm ảnh đen, mỗi điểm ảnh đen được coi như là một điểm trong không gian hai chiều và nó được xác định bởi số hàng và số cột. Do vậy, đối tượng ảnh trong 2.1a có thể được viết lại là  $\{ (3, 3) (3, 4) (4, 3) (4, 4) \}$ , với điểm ảnh phía trên bên trái là  $(0, 0)$ . Tuy nhiên, việc viết như vậy sẽ rất dài dòng và bất tiện nên ta gọi đơn giản đối tượng ảnh là  $A$ , và các phần tử trong đó là các điểm ảnh.



Hình 2.1: Hiệu quả của thao tác nhị phân đơn giản trên một ảnh nhỏ. (a) ảnh ban đầu (b) ảnh dẫn 1 điểm ảnh (c) ảnh dẫn 2 điểm ảnh (so với ảnh ban đầu).

### 2.1.1. Phép dẫn nhị phân(Dilation)

Bây giờ ta sẽ chỉ ra một số thao tác tập hợp đơn giản nhằm mục đích định nghĩa phép dẫn nhị phân qua chúng. Phép dịch A bởi điểm x(hàng, cột), được định nghĩa là một tập

$$(\mathbf{A})_x = \{c \mid c = a + x, a \in A\}$$

Chẳng hạn nếu x có tọa độ (1, 2), khi đó điểm ảnh đầu tiên phía trên bên trái của A sẽ dịch đến vị trí:  $(3, 3) + (1, 2) = (4, 5)$ . Các điểm ảnh khác trong A sẽ dịch chuyển một cách tương ứng, tức ảnh được dịch sang phải (cột) điểm ảnh và xuống phía dưới (hàng) điểm ảnh.

- Phép đối của tập A được định nghĩa như sau:

$$\hat{\mathbf{A}} = \{c \mid c = -a, a \in A\}$$

đó chính là phép quay A một góc  $180^\circ$  so với ban đầu.

- Phần bù của tập A là tập các điểm ảnh không thuộc đối tượng A, ở đây chính là các điểm ảnh trắng. Theo lý thuyết tập hợp thì:

$$\mathbf{A}^c = \{c \mid c \notin A\}$$

- Giao của hai tập hợp A và B là tập các phần tử thuộc về cả A lẫn B. Kí hiệu:

$$\mathbf{A} \cap \mathbf{B} = \{c \mid (c \in A) \wedge (c \in B)\}$$

- Hợp của hai tập hợp A và B là tập các phần tử thuộc A hoặc / và B. Kí hiệu:

$$\mathbf{A} \cup \mathbf{B} = \{c \mid (c \in A) \vee (c \in B)\}$$

- Hiệu của hai tập hợp A và B là tập:

$$\mathbf{A} - \mathbf{B} = \{c \mid (c \in A) \wedge (c \notin B)\}$$

nó là tập các các phần tử thuộc A nhưng không thuộc B

hay  $\mathbf{A} - \mathbf{B} = \mathbf{A} \cap \mathbf{B}^c$

Bây giờ ta có thể định nghĩa phép dẫn (dilation) qua lý thuyết tập hợp như sau:  
Phép dẫn tập A bởi tập B, đó là tập :

$$\mathbf{A \oplus B = \{c \mid c = a + b, a \in A, b \in B\}} \quad (1)$$

Dễ thấy trong toán học, đây là phép tổng trực tiếp A và B. A là đối tượng ảnh được thao tác và B được gọi là phần tử cấu trúc (viết tắt là cấu trúc). Để hiểu kĩ hơn về điều này, ta hãy coi A là đối tượng 2.1a và  $B = \{(0, 0) (0, 1)\}$

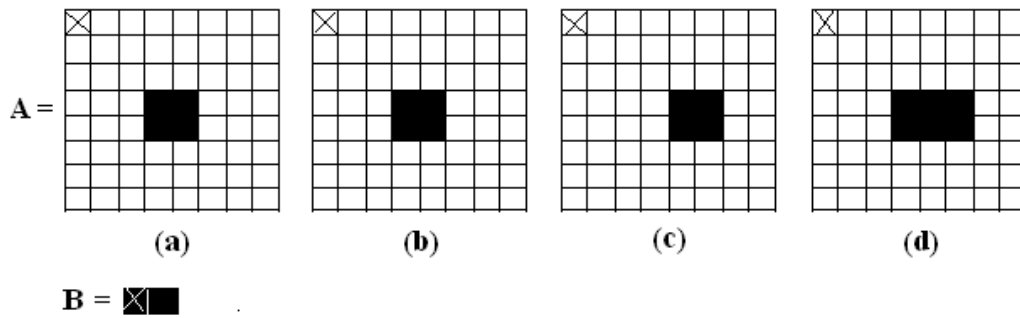
Những phần tử trong tập  $C = A \oplus B$  được tính dựa trên phương trình (1), có thể viết lại như sau:

$$\mathbf{A \oplus B = (A + \{(0, 0)\}) \cup (A + \{(0, 1)\})}$$

Cụ thể:

A		B	=	C
(3, 3)	+	(0, 0)	=	(3, 3)
(3, 4)	+	(0, 0)	=	(3, 4)
(4, 3)	+	(0, 0)	=	(4, 3)
(4, 4)	+	(0, 0)	=	(4, 4)
(3, 3)	+	(0, 1)	=	(3, 4)
(3, 4)	+	(0, 1)	=	(3, 5)
(4, 3)	+	(0, 1)	=	(4, 4)
(4, 4)	+	(0, 1)	=	(4, 5)

Trong đó, tập C gọi là kết quả của phép dẫn A sử dụng phần tử cấu trúc B và gồm các phần tử như được mô tả ở trên, tuy nhiên một vài điểm trong số chúng có thể trùng nhau.



**Hình 2.2:** Dẫn A bởi B. (a) Tập A ban đầu (b) Tập A cộng phân tử (0,0) (c) Tập A cộng phân tử (0,1) (d)hợp của (b) và (c) (kết quả phép dẫn)

Nhìn hình 2.2 trên, ta nhận thấy rằng trong các ảnh có hình 1 dấu thập (x). Những phần tử được đánh dấu (x) hoặc đen, hoặc trắng được coi như gốc (**Ogirin**) của mỗi ảnh. Việc xác định vị trí của gốc cấu trúc là rất quan trọng, nó có thể quyết định hướng co dẫn của ảnh. Nếu gốc ở bên trái, thì ảnh có xu hướng co dẫn về bên phải, gốc ở bên phải thì co dẫn về trái và nếu gốc ở giữa, tất nhiên, ảnh sẽ dẫn đều. Trong thí dụ trên do gốc của cấu trúc B ở bên trái nên ta thấy ảnh được dẫn về bên phải.

Nếu như gốc chứa một điểm ảnh trắng, khi đó ta nói rằng gốc không được bao gồm trong tập B. Thông thường, để dẫn ảnh đều về tất cả các phía, ta thường sử dụng cấu trúc có dạng ma trận  $3 \times 3$  với gốc ở chính giữa. Ta hãy xét thêm một ví dụ nữa, ví dụ này sẽ cho ta thấy sự dẫn về hai phía nếu như ta sử dụng cấu trúc có gốc ở giữa và gốc chứa một điểm ảnh trắng. Trong trường hợp cấu trúc có điểm ảnh trắng ở góc ta nói rằng gốc không được bao gồm trong cấu trúc.

Nhìn vào hình 2.3 dưới đây, ta có:

$$A1 = \{(1, 1)(1, 2)(2, 2)(2, 3)(3, 2)(3, 3)(4, 4)\}$$

và phần tử cấu trúc  $B1 = \{(0, -1)(0, 1)\}$

Dịch A1 bởi (0, -1) cho ta:

$$(A1)_{(0, -1)} = \{(1, 0)(1, 1)(2, 1)(2, 2)(3, 1)(3, 2)(4, 3)\}$$

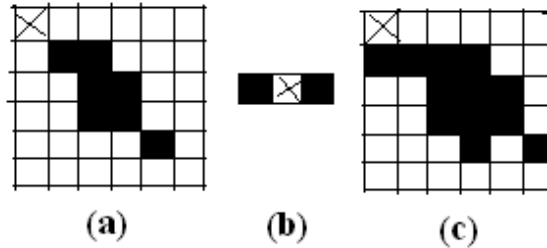
Dịch A1 bởi (0, 1):



$$(A1)_{(0,1)} = \{(1, 2)(1, 3)(2, 3)(2, 4)(3, 3)(3, 4)(4, 5)\}$$

Phép dẫn của A1 bởi B1 là hợp của  $(A1)_{(0,-1)}$  và  $(A1)_{(0,1)}$

Nhận thấy rằng trong hình 2.3, có một số phần tử của đối tượng ban đầu sẽ không có



Hình 2.3: Dẫn mất điểm ảnh .(a) ảnh A1 (b) phần tử cấu trúc B1 (c) A1 được dẫn bởi B1

Mặt trong ảnh kết quả, chẳng hạn (4, 4). Đó chính là do góc của phần tử cấu trúc không phải là một phần tử đối tượng (bởi ta coi phần tử đối tượng là điểm ảnh đen mà ở đây góc lại là một điểm trắng ).

Tổng quát hơn, ta có thể coi phép dẫn (dilation ) là hợp của tất cả các phép dịch bởi các phần tử của cấu trúc, kí hiệu:

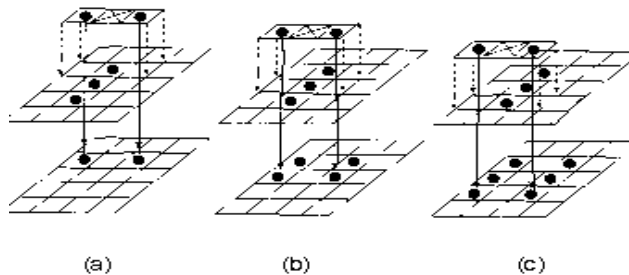
$$A \oplus B = \bigcup_{b \in B} (A)_b$$

Tuy nhiên với vai trò bình đẳng của A và B, ta coi A là cấu trúc và B là ảnh thì khi đó:

$$A \oplus B = \bigcup_{a \in A} (B)_a$$

Từ những điều trên, giúp ta tiếp cận đến một thao tác dẫn ảnh có thể được “ máy tính hóa”. Ta hãy coi những phần tử cấu trúc như là một mẫu và dịch nó trên ảnh. Khi mà góc của phần tử cấu trúc, hay mẫu, khớp với một điểm ảnh đen trên ảnh thì tất cả các điểm ảnh tương ứng với các điểm đen trên cấu trúc sẽ được đánh dấu và thay thế sau. Sau khi toàn bộ ảnh đã được quét qua bởi mẫu, thao tác dẫn ảnh coi như hoàn chỉnh. Thông thường, máy tính sẽ làm như sau: mỗi lần góc cấu trúc trùng điểm đen trên ảnh thì nó sẽ chép các phần tử đen của mẫu và gửi vào một ảnh mới với vị

trí thích hợp, gọi là ảnh kết quả, ảnh này ban đầu chỉ gồm các điểm trắng. Khi đó ảnh kết quả chính là ảnh được dẫn. Điều này được thể hiện khá rõ trong hình 2.4.



Hình 2.4: Dẫn ảnh sử dụng phần tử cấu trúc.(a)Góc cấu trúc định vị trên điểm ảnh đen đầu tiên và những điểm đen cấu trúc được chép sang ảnh kết quả ở những vị trí tương ứng.(b)Quá trình tương tự với điểm đen tiếp theo.(c)Quá trình hình thành

### 2.1.2. Phép co nhị phân (Erosion)

Nếu như phép dẫn có thể nói là thêm điểm ảnh vào trong đối tượng ảnh, làm cho đối tượng ảnh trở nên lớn hơn thì phép co sẽ làm cho đối tượng ảnh trở nên nhỏ hơn, ít điểm ảnh hơn(ở đây ta vẫn quan niệm rằng đối tượng ảnh là những điểm ảnh đen ). Trong trường hợp đơn giản nhất, một phép co nhị phân sẽ tách lớp điểm ảnh bao quanh đối tượng ảnh, chẳng hạn 2.1b là kết quả của phép co được áp dụng đối với 2.1c. Để hiểu hơn, ta tưởng tượng rằng một ảnh nhị phân có những điểm ảnh đen(đối tượng ảnh ) và điểm ảnh trắng (nền ). Từ ảnh ban đầu, ta thay các điểm đen mà lân cận của nó có ít nhất một điểm trắng thành trắng. Khi đó ảnh nhận được là ảnh được co bằng phép co đơn giản. Trong phép co này, mẫu được dùng chính là mảng  $3 \times 3$  của các điểm ảnh đen, đã được nói đến trong phép dẫn nhị phân trước đây.

Nhìn chung, phép co một ảnh A bởi cấu trúc B có thể được định nghĩa như là tập:

$$A \ominus B = \{c \mid (B)_c \subseteq A\}$$

Nói cách khác, đó là tập hợp các điểm ảnh  $c \in A$ , mà nếu cấu trúc B dịch chuyển theo các tọa độ của c, thì B vẫn nằm trong đối tượng ảnh A, tức B là một tập con của đối tượng ảnh cần co A. Tuy nhiên điều đó sẽ chưa chắc đã đúng nếu như phần tử cấu trúc B không chứa gốc (tức điểm ảnh gốc màu trắng ). Đầu tiên, ta hãy xét một ví dụ đơn giản sau đây: Xét phần tử cấu trúc  $B = \{(0, 0) (0, 1)\}$  và đối tượng ảnh

$A = \{(3, 3) (3, 4) (4, 3) (4, 4)\}$  như trong hình 2.2. Không cần thiết phải quan tâm đến tọa độ các điểm đen của A, mà chỉ cần quan tâm đến những tọa độ của các điểm đen của A mà khi ta di mẫu B trên đối tượng ảnh A thì gốc của B trùng một điểm ảnh đen của A. Ở đây ta quan tâm tới bốn tọa độ của bốn điểm đen của A sau:

$B_{(3, 3)} = \{(3, 3) (3, 4)\}$ , tức là dịch các điểm ảnh của B sang phải 3 và xuống dưới 3

Tương tự có:

$$B_{(4, 3)} = \{(4, 3) (4, 4)\}$$

$$B_{(3, 4)} = \{(3, 4) (3, 5)\}$$

$$B_{(4, 4)} = \{(4, 4) (4, 5)\}$$

Trong hai trường hợp đầu,  $B_{(3, 3)}$  và  $B_{(4, 3)}$ , tập hợp các điểm đen mà B dịch chuyển theo các tọa độ của chúng sao cho vẫn thuộc A sẽ xuất hiện trong phép co A bởi B. Điều này sẽ được minh họa rõ ràng qua 2.5.

Nếu như trong cấu trúc B không chứa gốc, ta gọi là cấu trúc  $B2 = \{(0, 1)\}$ . Khi đó cách tính toán tương tự như trên, nhưng không nhất thiết gốc phải trùng điểm ảnh đen khi ta di mẫu trên đối tượng ảnh A. Lúc này, kết quả như sau:

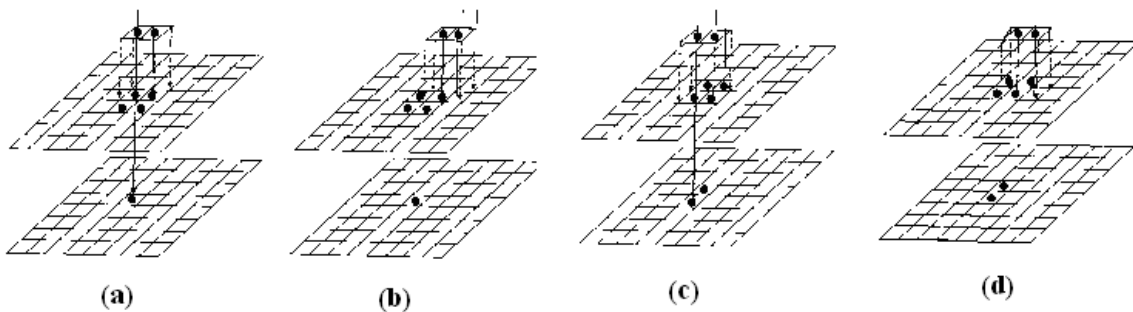
$$B_{(3, 2)} = \{(3, 3)\}$$

$$B_{(4, 2)} = \{(4, 3)\}$$

$$B_{(3, 3)} = \{(3, 4)\}$$

$$B_{(4, 3)} = \{(4, 4)\}$$

Điều này có nghĩa kết quả của phép co là  $\{(3, 2) (4, 2) (3, 3) (4, 3)\}$ , thế nhưng lại không phải là một tập con của A, mà lí do gây ra điều đó chính là gốc không được chứa trong mẫu B2.



Hình 2.5: Phép co nhị phân

(a) Phần tử cấu trúc được dịch chuyển đến vị trí một điểm đen trong ảnh. Trong trường hợp này, các thành viên của cấu trúc đều phù hợp với những điểm đen của ảnh cho nên cho kết quả điểm đen.

(b) Phần tử cấu trúc dịch chuyển tới điểm ảnh tiếp theo trong ảnh, và có một

Ta nhận thấy một điều quan trọng là: Phép co và phép dẫn không phải là những thao tác ngược nhau. Có thể trong một số trường hợp đúng là phép co sẽ giải hoạt hiệu quả của phép dẫn. Nhưng nhìn chung thì điều đó là không đúng, ta sẽ quan sát chúng một cách cụ thể hơn ở sau. Tuy nhiên, giữa phép co và phép dẫn có mối quan hệ qua biểu thức sau đây:

$$(\mathbf{B} \ominus \mathbf{A})^c = \mathbf{B}^c \oplus \hat{\mathbf{A}} \quad (2)$$

(chú ý:  $\hat{\mathbf{A}} = \{c \mid c = -a, a \in A\}$ )

Tức là phần bù của phép co ảnh A bởi B được coi như phép dẫn phần bù của A bởi tập đối của B. Nếu như cấu trúc B là đối xứng (ở đây ta quan niệm đối xứng theo toạ độ) thì tập đối của B không thay đổi, nghĩa là  $\hat{\mathbf{A}} = \mathbf{A}$

Khi đó:

$$(\mathbf{B} \ominus \mathbf{A})^c = \mathbf{B}^c \oplus \mathbf{A} \quad (3)$$

Hay, phần bù của phép co A bởi B được coi như phép dẫn nền của ảnh A (ta quy ước trong ảnh nhị phân rằng: đối tượng ảnh là những điểm đen quan sát, ảnh A là bao gồm cả điểm đen và nền).

Ta sẽ chứng minh biểu thức (3)

Theo định nghĩa của phép co ở trên, ta có:

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

Khi đó

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

$$\text{Xét về trái} = (\mathbf{B} \ominus \mathbf{A})^c = \{z \mid (\mathbf{A})_z \subseteq \mathbf{B}\}^c$$

$$\begin{aligned}
 &= \{z \mid (A)_z \cap B \neq \emptyset\}^c \\
 &= \{z \mid (A)_z \cap B^c = \emptyset\}^c \\
 &= \{z \mid (A)_z \cap B^c \neq \emptyset\}
 \end{aligned}$$

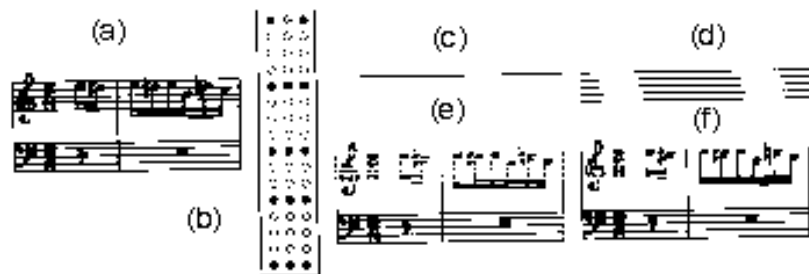
Mặt khác

$$\begin{aligned}
 (A)_z &= \{c \mid c = a + z, a \in A\}, \text{ do đó:} \\
 (B \ominus A)^c &= \{z \mid (a + z) \in B^c, b \in B\} \\
 &= \{z \mid a + z = b, b \in B^c, a \in A\} \\
 &= \{z \mid z = b - a, b \in B^c, a \in A\} \\
 &= B^c \oplus \hat{A} = \{z \mid (A)_z \subseteq B\}^c
 \end{aligned}$$

Đó là điều cần chứng minh  $\square$ .

Thao tác co ảnh cũng đưa ra một vấn đề mà không hề liên quan đến phép dẫn, vấn đề đó có thể hiểu là phần tử cấu trúc có thể “tạm bỏ qua”. Tức là khi ta sử dụng một cấu trúc nhị phân chặt chẽ để thực hiện một phép co ảnh, những điểm ảnh đen trong cấu trúc phải phù hợp với các đối tượng nhằm mục đích sao cho điểm ảnh cần quan tâm phải được đưa vào ảnh kết quả. Tuy nhiên điều đó không đúng cho những điểm ảnh trắng trong cấu trúc. Ý nghĩa của “tạm bỏ qua” chính là ở chỗ: Ta không quan tâm đến sự phù hợp giữa điểm ảnh trong đối tượng ảnh và điểm trắng trong cấu trúc, nói cách khác không quan tâm đến những điểm ảnh trắng trong cấu trúc trong trường hợp này.

Hình vẽ 2.6 dưới đây sẽ minh họa rõ hơn cách dùng cấu trúc cho phép co ảnh trong ngữ cảnh thực tế.



**Hình 2.6:** Xoá hình thái những dòng ngang của khuông nhạc

a. Ảnh gốc

b. Phần tử cấu trúc

c. Kết quả phép co (a) bởi (b)

d. Kết quả phép dẫn cùng cấu trúc

e. Lấy (a) trừ (d)

f. Sử dụng toán tử hình thái đơn giản để điền vào ô trống

### 2.1.3. Các phép toán đóng mở ảnh (closing and opening)

#### 2.1.3.1. Phép mở

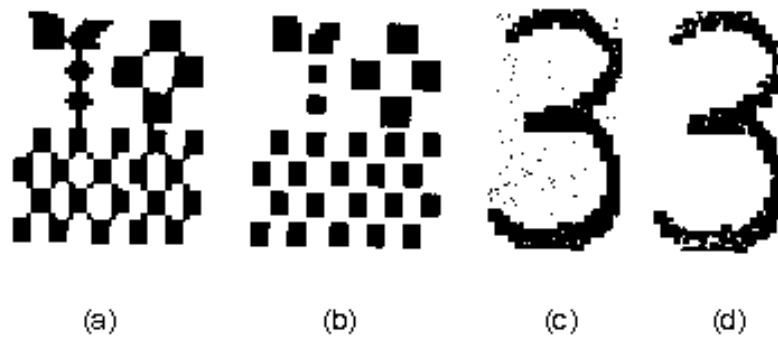
Nếu như ta áp dụng phép co ảnh đối với một ảnh và sau đó lại áp dụng tiếp phép dẫn ảnh đối với kết quả trước thì thao tác đó được gọi là phép mở ảnh, hay với  $I$  là ảnh,  $D$  là Dilation(dẫn) và  $E$  là Erosion(co).

$$\text{Opening}(I) = D(E(I))$$

Tên của phép toán ” mở “ ảnh dường như đã phản ánh rõ tác dụng của nó. Tác dụng của nó chính là ”mở” những khoảng trống nhỏ giữa các phần tiếp xúc trong đối tượng ảnh, làm cho ảnh dường như bớt ”gai”.Hiệu quả này dễ quan sát nhất khi sử dụng cấu trúc đơn giản. Hình 2.7 trình bày ảnh có những phần của nó tiếp xúc nhau. Sau thao tác mở đơn giản đối tượng ảnh đã dễ nhận hơn so với ban đầu.

Hình 2.7 cũng minh hoạ một đối tượng khác, hoàn toàn tương tự, sử dụng phép mở ảnh và nhiễu ở giữa số 3 đã biến mất. Bước co trong phép mở ảnh sẽ xoá những điểm ảnh cô lập được coi như những biên, và phép dẫn ảnh tiếp sau sẽ khôi phục lại các điểm biên và loại nhiễu. Việc xử lý này dường như chỉ thành công với những nhiễu đen còn những nhiễu trắng thì không.

Ví dụ mà ta đã xét 2.6 cũng có thể coi là một phép mở nhưng phần tử cấu trúc ở đây phức tạp hơn. Ảnh được xói mòn chỉ còn lại một đường ngang và sau đó được dẫn ra bởi phần tử cấu trúc tương tự. Lại quay về ảnh 2.7 và ta thử xem cái gì đã được xói mòn? Đó là các điểm đen trừ những hình vuông nhỏ màu đen, hay có thể nói rằng nó xoá mọi thứ trừ những cái mà ta cần quan tâm.



Hình 2.7: Sử dụng phép toán mở

- Một ảnh có nhiều vật thể được liên kết
- Các vật thể được cách ly bởi phép mở với cấu trúc đơn giản
- Một ảnh có nhiễu
- Ảnh nhiễu sau khi sử dụng phép mở, các điểm nhiễu đen đã biến mất

### 2.1.3.2. Phép đóng

Tương tự phép mở ảnh nhưng trong phép đóng ảnh, thao tác dẫn ảnh được thực hiện trước, sau đó mới đến thao tác co ảnh và cùng làm việc trên cùng một phần tử cấu trúc.

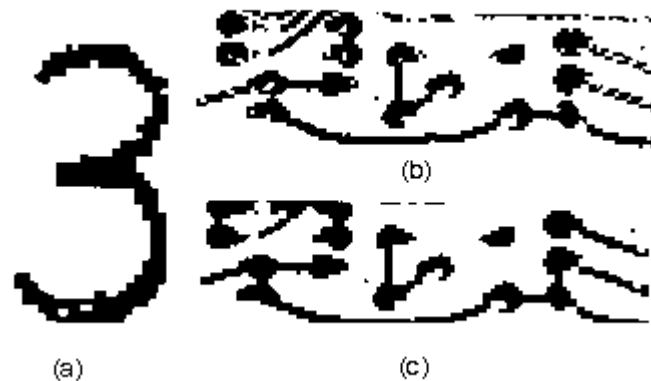
$$\text{Close (I)} = \text{E(D(I))}$$

Nếu như phép mở ảnh tạo ra những khoảng trống nhỏ trong điểm ảnh thì trái lại, phép đóng ảnh sẽ lấp đầy những chỗ hổng đó. Hình 2.8a trình bày trình bày một thao tác đóng ảnh áp dụng cho hình 2.7d, mà bạn nhớ rằng đó là kết quả của việc xóa nhiễu. Phép đóng ảnh quả là có tác dụng trong việc xóa những nhiễu trắng trong đối tượng ảnh mà phép mở ảnh trước đây chưa thành công.

Hình 2.8b và 2.8c trình bày một ứng dụng của phép co ảnh nhằm nối lại những nét gãy. ảnh ban đầu 2.8b là một bản mạch, sau khi sử dụng phép co các điểm gãy đã được liên kết nhau ở một số điểm ảnh. Phép đóng ảnh này đã gắn được nhiều điểm ảnh gãy, nhưng không phải là tất cả. Điều quan trọng nhận thấy rằng khi sử dụng những ảnh thực, thật hiếm khi xử lý ảnh một cách hoàn chỉnh mà chỉ cần một kĩ

thuật, phải sử dụng nhiều phần tử cấu trúc mà có khi có những kĩ thuật nằm ngoài Hình thái học (phép toán hình thái)

Đóng ảnh cũng có thể được sử dụng để làm trơn những đường viền của những đối tượng trong một ảnh. Thỉnh thoảng, việc phân ngưỡng có thể đưa ra một sự xuất hiện những điểm “nhám” trên viền; Trong những trường hợp khác, đối tượng “nhám” tự nhiên, còn “nhám” do ảnh chụp có thể dùng phương pháp đóng ảnh để xử lý. Tuy nhiên có thể phải sử dụng nhiều hơn một mẫu cấu trúc, kể từ khi cấu trúc đơn giản chỉ sử dụng cho việc xoá hoặc làm trơn những điểm ảnh cá biệt. Khả năng khác chính là việc lặp lại số phép co tương tự sau khi thực hiện số phép giãn nào đó.



**Hình 2.8:** Phép đóng

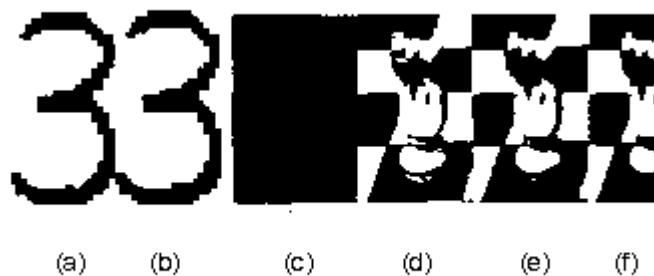
- Kết quả đóng của hình 2.8d sử dụng cấu trúc đơn giản
- Ảnh của một bảng mạch được phân ngưỡng và có các vết đứt
- Ảnh tương tự sau khi đóng nhưng những nét đứt đã được nối liền.

Trước tiên, quan tâm đến những ứng dụng làm trơn và vì mục đích này ra sẽ sử dụng để làm thí dụ. Trong ảnh 2.9a đã được thực hiện cả 2 phép đóng và mở và nếu thực hiện tiếp phép đóng sẽ không gây thêm bất kì một thay đổi nào. Tuy nhiên viền



của đối tượng ảnh vẫn còn gai và vẫn có những lỗ hổng trắng bên trong của đối tượng. Sử dụng phép mở với độ sâu 2, tức là sau khi co 2 lần thì dẫn 2 lần, khi đó nó sẽ cho ta kết quả là hình 2.9a. Chú ý rằng những lỗ trước đây đã được đóng và viền bây giờ có vẻ như “trơn” hơn so với trước. Phép mở 3 chiều, tương tự chỉ gây ra thay đổi rất nhỏ so với 2 chiều (2.9b), chỉ có thêm một điểm ảnh bên ngoài được xoá. Nhìn chung, sự thay đổi không đáng kể.

Hầu hết những phép đóng mở ảnh sử dụng những phần tử cấu trúc trong thực tế. Cách tiếp cận cổ điển để tính toán một phép mở với độ sâu N cho trước là thực hiện N phép co nhị phân và sau đó là N phép dẫn nhị phân. Điều này có nghĩa là để tính tất cả các phép mở của một ảnh với độ sâu 10 thì phải thực hiện tới 110 phép co hoặc phép dẫn. Nếu phép co và dẫn lại được thực hiện một cách thủ công thì phải đòi hỏi tới 220 lần quét qua ảnh.

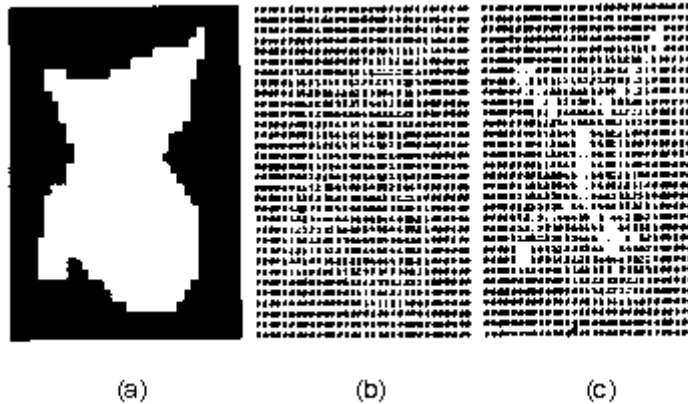


**Hình 2.9:** Phép đóng với độ sâu lớn

- Một 1a. Từ 2.8a, sử dụng phép đóng với độ sâu 2  
đây giá trị 1b. Phép đóng với độ sâu 3
- 
- c. Một vùng bàn cờ  
d. Vùng bàn cờ được phân ngưỡng thể hiện những điểm bất quy tắc và một vài lỗ.  
e. Sau khi thực hiện phép đóng với độ sâu 1

cách của điểm ảnh đó so với điểm ảnh nền gần nhất. Những điểm ảnh trên một đường viền sẽ mang giá trị 1, có nghĩa là chúng có độ dày 1 tính từ điểm ảnh nền gần nhất, tương tự, nếu cách điểm ảnh nền 2 điểm thì mang giá trị 2, và cứ như thế. Kết quả có sự xuất hiện của bản đồ chu tuyến; ở trong bản đồ đó, những chu tuyến đại diện cho khoảng cách xét từ viền vào. Ví dụ, đối tượng được trình bày trong 2.10a có bản đồ khoảng cách được trình bày trong 2.10b. Bản đồ khoảng cách chứa đủ thông tin để thực hiện phép co với bất kì số điểm ảnh nào chỉ trong một lần di mẫu qua ảnh; mặt khác, tất cả các phép co đã được mã hoá thành một ảnh. Ảnh co tổng thể này có thể được tạo ra chỉ trong 2 lần di qua ảnh gốc và một phép phân ngưỡng đơn giản sẽ đưa cho ta bất kì phép co nào mà ta muốn.

Cũng có một cách tương tự cách của phép co tổng thể, mã hoá tất cả các phép mở có thể thành một ảnh chỉ một mức xám và tất cả các phép đóng có thể được tính toán đồng thời. Trước hết, như phép co tổng thể bản đồ khoảng cách của ảnh được tìm ra. Sau đó tất cả các điểm ảnh mà không có tối thiểu một lân cận gần hơn đối với nền và một lân cận xa hơn đối với nền, sẽ được định vị và đánh dấu: Những điểm ảnh này sẽ được gọi là những điểm nút. Hình 2.10c trình bày những điểm nút có liên quan đến đối tượng hình 2.10a. Nếu bản đồ khoảng cách được nghĩ như một bề mặt ba chiều, mà trong đó khoảng tính từ nền được xem như chiều cao, do vậy mà mỗi điểm ảnh có thể được nghĩ như chóp của một tháp với độ nghiêng được tiêu chuẩn hoá. Những chóp đó không được bao gồm trong bất kì một tháp khác là những điểm nút. Một cách để định vị những điểm nút là quét bản đồ khoảng cách, quan sát các điểm ảnh đối tượng; tìm giá trị MIN và MAX của các lân cận của điểm ảnh quan tâm, và tính (MAX - MIN): Nếu giá trị này nhỏ hơn MAX có thể, nó là 2 khi sử dụng 8 khoảng cách, thì điểm đó chính là nút.

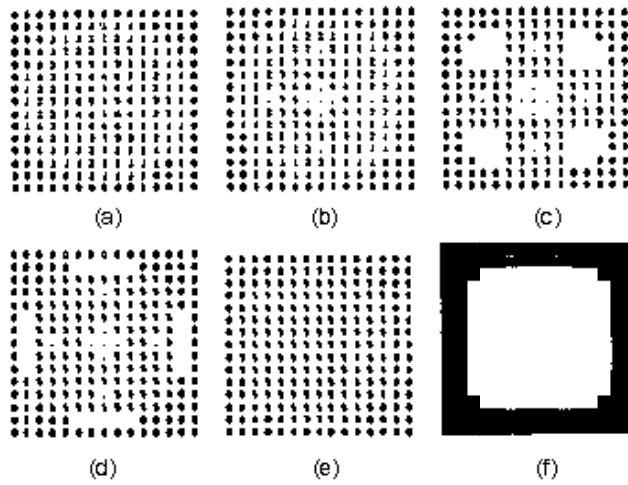


**Hình 2.10:** Phép cơ sử dụng một bản đồ khoảng cách

- a. Giọt nước
- b. Bản đồ khoảng cách của ảnh giọt nước
- c. Những điểm nút trong ảnh này hiện lên như một chu trình.

Để mã hoá tất cả các phép mở của đối tượng, đặt một đĩa số sao cho tâm chính là mỗi điểm nút. Khi đó những giá trị của điểm ảnh trong đĩa sẽ mang giá trị của nút. Nếu một điểm ảnh đã được hút, khi đó nó sẽ nhận giá trị lớn hơn giá trị hiện tại của nó hoặc một điểm ảnh mới được vẽ. Đối tượng kết quả có đường biên tương tự như ảnh nhị phân gốc, do vậy mà ảnh đối tượng có thể được tái tạo chỉ từ những điểm nút. Thêm vào đó, những mức xám của ảnh được mở tổng thể này đại diện một cách mã hoá tất cả các phép mở có thể. Như một ví dụ, hãy xét đối tượng được định dạng hình đĩa trong hình 2.11a và bản đồ khoảng cách tương ứng trong 2.11b. Có 9 điểm nút: 4 điểm có giá trị 3 và còn lại là giá trị 5. Phân ngưỡng ảnh được mã hoá mang lại một phép mở có độ sâu tương tự ngưỡng.

Tất cả các phép đóng có thể được mã hoá song song với các phép mở nếu bản đồ khoảng cách được thay đổi gồm khoảng cách của những điểm ảnh nền từ một đối tượng. Những phép đóng thành những giá trị nhỏ hơn giá trị trung tâm tùy ý và những phép mở được mã hoá thành những giá trị lớn hơn giá trị trung tâm này



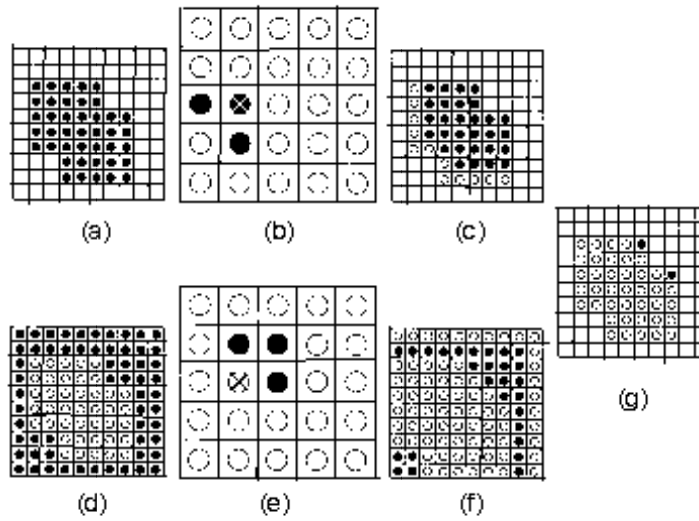
#### 2.1.4. Đánh trúng và đánh trượt

“Đánh trúng và đánh trượt” là một phép toán Hình thái học được thiết kế để định vị những hình dạng đơn giản bên trong một ảnh. Nó dựa trên phép co, thật bình thường đó là phép co A bởi cấu trúc S bao gồm chỉ những điểm ảnh (đúng hơn là những vị trí ) mà theo nó, S được chứa trọn bên trong A (theo như trước đây ) cho đến chỉ cần thoả mãn tập hợp điểm ảnh trong một vùng nhỏ của A. Tuy nhiên vậy thì nó cũng bao gồm cả những vùng mà ở vùng đó, những điểm ảnh nền lại không phù hợp với những điểm ảnh nền của cấu trúc S và những vị trí đó sẽ không được nghĩ là phù hợp theo nghĩa thông thường. Cái mà chúng ta cần quan tâm đó chính là một thao tác mà phù hợp với cả hai: Những điểm ảnh nền và những điểm ảnh đối tượng (ta coi ảnh gồm đối tượng và nền ) của cấu trúc S trong A. Nếu những điểm ảnh đối tượng trong S phù hợp với những điểm ảnh đối tượng trong A được gọi là “đánh

trúng “ và được hoàn chỉnh bởi một phép co đơn giản  $A \ominus$  . Những điểm ảnh nền trong  $A$  được coi là những điểm ảnh đối tượng trong  $A^c$  và trong khi chúng ta có thể sử dụng  $S^c$  như nền của  $S$ . Coi  $T$  như là một cấu trúc mới,  $A$  “đánh trúng ” nền gọi là “đánh trượt ” và được coi như phép  $A^c \ominus T$ . Chúng ta muốn những vị trí mà cả “đánh trúng và đánh trượt ”, đó là những điểm ảnh thoả mãn:

$$A \otimes (S, T) = (A \ominus S) \cap (A^c \ominus T) \tag{4}$$

Coi như một ví dụ, ta hãy sử dụng sự đối dạng để tách ra những góc phía trên bên phải. Hình 2.12a trình bày một đối tượng ảnh giống 2 hình vuông đè lên nhau góc phần tư.



Cung phai chu y rang cau truc dahn cho anh nen 2.12a lai khong phai la phan bu của cấu trúc dành cho ảnh gốc 2.12a. Thực vậy, nếu nó là phần bù thì kết quả sẽ là một ảnh rỗng. Nhân tiện cũng phải nói rằng những điểm ảnh phía trên bên phải trong 2.12f là trắng bởi vì chúng phù hợp với những vị trí mà ở đó cấu trúc 2.12e có những

điểm ảnh đen được đặt bên ngoài của những viền trong ảnh. Phép toán phân bù tạo ra một ảnh cỡ tương tự như ảnh được lấy phân bù dù rằng khi sử dụng trong tập hợp, điều này không đúng. Điều này có thể được tránh bằng việc sao chép ảnh vào thành một ảnh lớn hơn trước khi lấy phân bù của ảnh đó.

### 2.1.5. Phép toán dẫn nở có điều kiện

Trong thực tế, có nhiều lúc ta thực sự cần phải dẫn một đối tượng bằng một cách nào đó sao cho những điểm ảnh còn lại không bị ảnh hưởng. Chẳng hạn như nếu ta mong một đối tượng trong một ảnh không chiếm phần nào đó trong ảnh đó thì phép dẫn đối tượng phải không được làm cho đối tượng lấn vào khu vực đó. Trong trường hợp như vậy, ta phải dùng phép dẫn theo điều kiện. Khu vực cấm của ảnh đó được coi như là một ảnh thứ hai mà trong ảnh thứ hai đó, những điểm ảnh bị cấm là đen (mang giá trị 1). Phép dẫn có điều kiện được kí hiệu:

$$A \oplus (S_e, A') \quad (5)$$

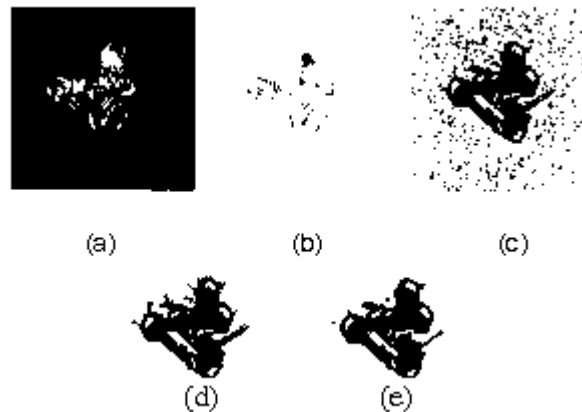
Trong đó,  $S_e$  là cấu trúc được sử dụng trong phép dẫn và  $A'$  là ảnh đại diện cho tập hợp những điểm ảnh bị cấm.

Kĩ thuật trên rất hữu ích cho việc phân đoạn một ảnh. Chọn được một ngưỡng tốt cho việc phân đoạn mức xám có thể là rất khó khăn. Tuy nhiên, hai ngưỡng tồi có thể được sử dụng để thay cho một ngưỡng tốt. Nếu một ngưỡng rất cao được áp dụng cho một ảnh thì những điểm ảnh thỏa mãn ngưỡng đó chắc chắn sẽ là những điểm ảnh của đối tượng thế nhưng như vậy ta cũng dễ bỏ qua nhiều điểm ảnh khác của đối tượng. Còn nếu ngưỡng áp dụng mà quá thấp thì ta dễ chọn phải nhiều điểm ảnh mà không phải là điểm ảnh của đối tượng.

Do đó mà phép dẫn theo điều kiện dưới đây được thực hiện:

$$R = I_{\text{high}} \oplus (\text{Simple}, I_{\text{low}}) \quad (6)$$

Ảnh  $R$  bây giờ là ảnh được phân đoạn từ ảnh gốc và kết quả của nó khá tốt trong một số trường hợp.



**Hình 2.13:** Dẫn theo điều kiện

- a. ảnh một chồng chìa khoá
- b. ảnh phủ định do sử dụng ngưỡng cao
- c. Kết quả của sử dụng ngưỡng thấp
- d. Phép dẫn có điều kiện của (b) sử dụng phần tử cấu trúc đơn giản, điều kiện theo (c)
- e. Kết quả sau khi “làm sạch” nhiều, bằng sử dụng phép mở.

### 2.1.6. Kỹ thuật đếm vùng

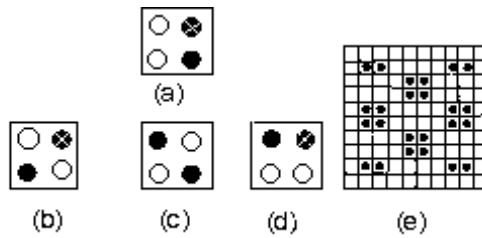
Được coi như một ví dụ cuối cùng trong Đồ án về cách sử dụng những toán tử hình thái trong ảnh nhị phân. Có thể sử dụng những toán tử hình thái dùng để đếm số vùng trong một ảnh. Phương pháp này đầu tiên được đưa ra bởi Levialdi và sử dụng tới 6 phần tử cấu trúc: 4 phần tử đầu được dùng để co ảnh và được lựa chọn một cách cẩn thận sao cho không làm thay đổi sự nối kết giữa những vùng được co. Hai phần tử cấu trúc cuối được dùng để đếm những điểm ảnh điểm ảnh “1” bị cô lập. Số vùng ban đầu là số điểm ảnh bị cô lập trong ảnh vào  $A$  và ảnh của lần lặp thứ 0 là  $A$ , hay kí hiệu:

$$A_0 = A \quad (10)$$

Ảnh của các lần lặp tiếp theo là hợp của bốn phép co với bốn phần tử cấu trúc ban đầu với ảnh của lần lặp hiện tại, tức là:

$$A_{n+1} = (A_n \ominus L_1) \cup (A_n \ominus L_2) \cup (A_n \ominus L_3) \cup (A_n \ominus L_4) \quad (7)$$

trong đó  $L_1, L_2, L_3, L_4$  là bốn cấu trúc ban đầu (xem hình vẽ 2.14)



**Hình 2.14:** Đếm vùng

a.b.c.d. Các cấu trúc

e. Ví dụ của ảnh có 8 vùng. Thuật toán làm việc đúng.

Số vùng trong lần lặp  $i$  chính là số điểm ảnh bị cô lập trong ảnh lặp thứ  $i$  và phép lặp này sẽ dừng khi  $A_n$  trở thành rỗng (tức là tất cả các ảnh đều mang giá trị 0). Toàn bộ tổng số vùng là tổng của tất cả các giá trị của số vùng trong các lần lặp.

## 2.2. Thao tác trên ảnh xám

### 2.2.1. Phép co và phép dẫn

Sử dụng đa mức xám đem lại một sự phiền hà lớn trong cả lý thuyết và thực nghiệm. Cũng đã có vài câu hỏi được đặt ra, giả sử như trong ảnh xám thì phép dẫn trong ảnh xám để làm gì và cách sử dụng như thế nào ?

$$a \oplus sT[i, j] = \underset{r, c}{\text{MAX}} |A[i - r, j - c] + S[r, c]|, [i - r, j - c] \in A, [r, c] \in SQ \quad (8)$$

Trong đó,  $S$  là cấu trúc đơn giản và  $A$  là ảnh xám được dẫn. Đây là một định nghĩa của phép dẫn ảnh xám và nó có thể được tính toán theo các bước sau:

- i. Vị trí gốc của phần
- ii. từ cấu trúc đặt trên điểm ảnh đầu tiên của ảnh được dẫn ta gọi vị trí đó là  $m$
- iii. Tính tổng của mỗi cặp điểm tương ứng của phần tử cấu trúc và ảnh
- iv. Tìm giá trị lớn nhất của tất cả những giá trị tổng ở trên và đặt giá trị lớn nhất vào vị trí  $m$  của ảnh kết quả.
- v. Lặp lại quá trình từ  $i$  tới  $iii$  cho các điểm còn lại của ảnh

Những giá trị của những điểm ảnh trong phần tử cấu trúc là những giá trị xám có thể và có thể mang giá trị âm. Nhưng những điểm ảnh được định giá trị âm không thể được hiển thị và có hai cách để giải quyết vấn đề này: Ta có thể đặt các giá trị âm bằng 0 hoặc tìm giá trị nhỏ nhất trong ảnh kết quả và đặt nó là 0, các giá trị còn lại sẽ



được cộng thêm một lượng tương ứng, sao cho các giá trị ảnh chênh nhau một lượng không đổi.

Tương tự ta sẽ đưa ra một định nghĩa cho phép co đối với ảnh xám như sau:

$$(A \ominus S)[i, j] = \text{MIN}\{ A[i-r, j-c] - S[r, c]; [i-r, j-c] \in A, [r, c] \in S \} \quad (9)$$

Cũng giống như phép dẫn xám, phép co thực hiện tính toán theo 4 bước:

- i. Vị trí gốc của phần tử cấu trúc đặt trên điểm ảnh đầu tiên của ảnh được co ta gọi vị trí đó là m
- ii. Tính hiệu của mỗi cặp điểm tương ứng của phần tử cấu trúc và ảnh
- iii. Tìm giá trị nhỏ nhất của tất cả những giá trị hiệu ở trên và đặt giá trị nhỏ nhất vào vị trí m của ảnh kết quả.
- iv. Lặp lại quá trình từ i tới iii cho các điểm còn lại của ảnh.

### 2.2.2. Các phép toán đóng, mở

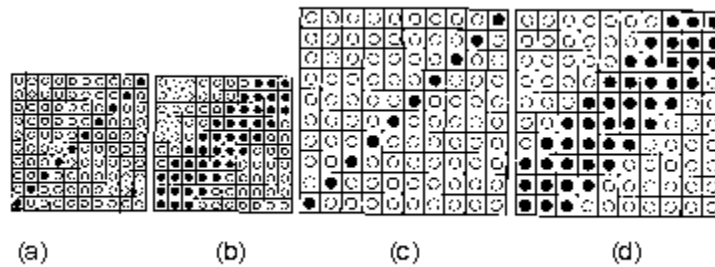
Phép đóng và mở một ảnh xám được thực hiện theo cách tương tự trước đây, ngoại trừ rằng phép co và dẫn ảnh xám được sử dụng; đó là, một phép mở là một phép co theo sau bằng một phép dẫn sử dụng cùng cấu trúc và phép đóng thì theo thứ tự ngược lại, dẫn trước rồi co sau. Tuy nhiên, sử dụng một mẫu hình học để miêu tả nó sẽ dễ dàng quan sát hơn. Coi một ảnh xám như một bề mặt ba chiều, trong đó trục ngang x(dòng) và trục dọc y(cột) vẫn giữ nguyên như trước, còn trục cao z đặc trưng cho giá trị mức xám. Phần tử cấu trúc cũng sẽ là một ảnh xám. Ta hãy xét thử một ảnh xám cấu trúc dạng hình cầu:

```

0000000000
00111311100
01122322110
01233433210
01235553210
02345654320
01235553210
01233433210
01122322110
00111211100
0000000000

```

Lẽ dĩ nhiên, đây mới chỉ là gần giống hình cầu chứ chưa phải là hình cầu thực sự do lỗi cắt và lấy mẫu. Tuy nhiên, ta có thể tưởng tượng rằng: phân tử cấu trúc như một hình cầu, ảnh xám của ta như một bề mặt ba chiều (mặt cong), độ lồi lõm của mặt cong đó tượng trưng cho giá trị mức xám (trục cao  $Z$ ), và khi đó phép đóng phép mở có thể hiểu như sau Ta lăn cầu ở phía dưới của bề mặt cong đó, những điểm nào của bề mặt cong đó tiếp xúc với hình cầu thì ta chấp nhận (đó chính là giá trị xám được giữ lại) và khi đó ta gọi là phép mở. Tương tự đối với phép đóng, nhưng thay vì lăn cầu ở phía dưới, ta lăn cầu ở phía trên của bề mặt cong và cũng lấy những điểm tiếp xúc cầu. Cụ thể hơn, ta hãy xem hình 2.15. ở hình 2.15 chỉ ra quá trình xử lý này trong hai chiều, hình cầu được thay bằng hình tròn. Trong trường hợp này phép mở có thể được coi như một quá trình làm trơn giảm mức xám trung bình của các điểm ảnh, trái lại phép đóng xuất hiện làm tăng mức xám trung bình của các điểm ảnh.



**Hình 2.15:** Phép dẫn đa cấp xám

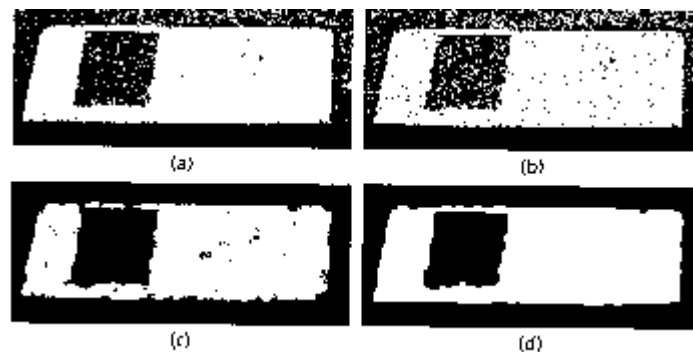
- a. ảnh 2 mức của một đường thẳng
- b. Phép dẫn nhị phân từ (a) bởi cấu trúc đơn giản
- c. Một ảnh đa cấp xám; nền có mức xám 0 nhưng đường thẳng có mức xám 20
- d. Sau khi thực hiện phép dẫn với đường thẳng.

Một ứng dụng thú vị của phép mở và phép đóng đó là ứng dụng trong việc kiểm tra chính xác đối tượng bằng mắt. Chẳng hạn, khi một đối tượng được cắt hay đánh bóng, khi đó có những vết xước còn lại trong kim loại và những vết xước đó có thể được nhận dạng một cách dễ dàng hơn nếu ta dùng ánh sáng chiếu vào bề mặt kim loại và nhờ sự phản xạ trên bề mặt để đánh giá.

### 2.2.3. Làm trơn

Thao tác làm trơn trên ảnh xám có thể được coi như là một phép mở mà tiếp theo sau đó là một phép đóng ảnh. Hiệu quả của thao tác này là nó sẽ xoá đi những điểm quá sáng hoặc quá tối trên ảnh gốc. Do vậy, có những điểm ảnh thực sự là nhiễu sẽ được xử lý, nhưng cũng không tránh khỏi những giá trị ảnh thực sự cũng bị ảnh hưởng và nhìn chung, giá phải trả cho việc giảm nhiễu là ảnh bị mờ đi so với ban đầu.

Hình 2.16 a miêu tả ảnh của một chiếc bảo vệ đĩa mà đã được liệt vào dạng nhiễu Gauss (theo phân loại thông thường) với độ lệch chuẩn 30. Hình 2.16c trình bày kết quả của phép làm nhiễu hình thái được áp dụng cho ảnh này; ta thấy có khi ảnh được làm trơn ở 2.16 c lại không rõ bằng ảnh ban đầu 2.16 a. Tuy nhiên, so hai ảnh 2.16b và 2.16d ta thấy sự khác biệt giữa chúng là: Ảnh lúc đầu được phân ngưỡng, sau đó ta mới làm trơn thì kết quả thật là tuyệt vời, ảnh thu được 2.16d đã hết nhiễu. Điều đó cho ta thấy rằng, việc phân ngưỡng quả là phân ngưỡng quả là có tác dụng tốt đối với phép làm trơn. Phần tử cấu trúc được sử dụng ở đây chỉ là đơn giản, nhưng việc lựa chọn phần tử cấu trúc nào còn phải tùy thuộc vào kiểu nhiễu nào để mà sử dụng cho thích hợp.



**Hình 2.16** Làm trơn đa cấp xám

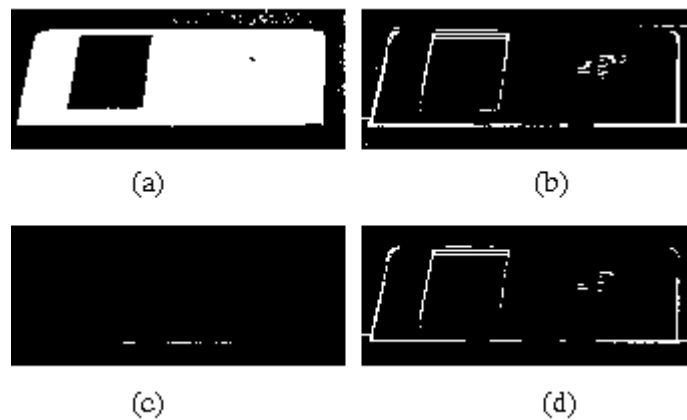
- ảnh miếng bảo vệ đĩa có kèm theo nhiễu Gauss có độ lệch chuẩn 30
- Sau khi phân ngưỡng từ (a), ảnh như được rắc thêm “muối và hạt tiêu”
- ảnh (a) sau khi được làm trơn
- ảnh được làm trơn và sau khi phân ngưỡng, nó đã hết nhiễu.

## 2.2.4. Gradient

Như trước đây, phương pháp để dò biên của một đối tượng ảnh hai mức xám đã được thảo luận. Ý tưởng chính là co một ảnh sử dụng phần tử cấu trúc đơn giản và ảnh co sau đó có thể bị trừ bởi ảnh gốc, để lại chỉ những điểm ảnh đã được co. Điều này cũng có thể thực hiện với ảnh đa cấp. Bởi vì sự tương phản trong ảnh đa cấp xám không tốt bằng trong ảnh nhị phân, do vậy mà hiệu quả của việc dò biên không tốt bằng. Tuy nhiên, ta có thể cải thiện tình trạng này bằng cách áp dụng công thức sau:

$$\mathbf{G} = (\mathbf{A} \oplus \mathbf{S}) - (\mathbf{A} \ominus \mathbf{S}) \quad (10)$$

Trong đó  $\mathbf{S}$  là phần tử cấu trúc. Thay vì lấy ảnh ban đầu trừ ảnh co, ở đây ta lấy ảnh dẫn trừ đi ảnh được co. Điều này sẽ làm tăng sự tương phản và bề rộng của những cạnh được trích chọn. Phương trình (10) là định nghĩa của gradient hình thái. Hai thuật toán: dò biên (dựa phương trình 5) và gradient hình thái (dựa phương trình 10) trong hình 2.17 được áp dụng cho ảnh của miếng bảo vệ đĩa, ở đây cũng coi phần tử cấu trúc là đơn giản.



**Hình 2.17:** Đường dốc hình thái

a. ảnh bảo vệ đĩa

b. Các cạnh được làm rõ bằng phương pháp trích biên sau đó được phân ngưỡng (đề cập trước đây)

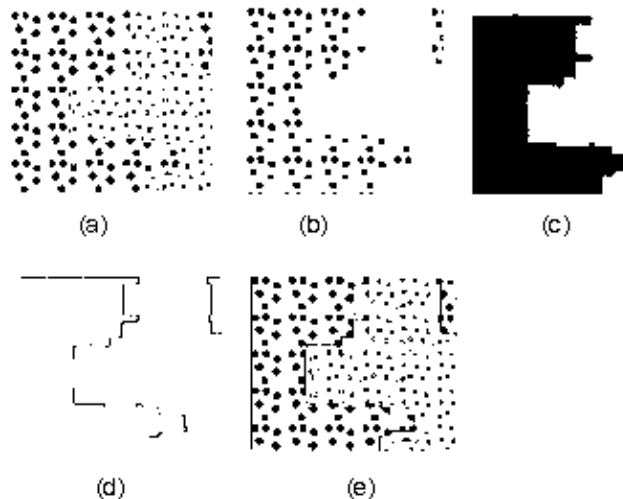
c. Đường dốc hình thái

d. ảnh (c) sau khi phân ngưỡng.

### 2.2.5. Phân vùng theo cấu trúc

Phép đóng xoá đi những chi tiết tối và phép mở sẽ xoá đi những vùng tối trong ảnh. Điều này dẫn ta đến một ứng dụng dành cho cấu trúc và việc nhận dạng những vùng trong ảnh dựa trên chính cấu trúc của vùng đó. Nếu ta có một cấu trúc gồm những vết nhỏ xen lẫn cấu trúc gồm những vết lớn, thì phép đóng với kích cỡ của những vết nhỏ sẽ có tác dụng xoá chúng, thế nhưng sẽ để lại những vết trên cấu trúc lớn. Khi đó ta thực hiện phép mở với kích cỡ của những khoảng trống giữa những vết lớn giữa những vết lớn trong cấu trúc lớn sẽ nối chúng lại thành một vùng đen lớn. Khi đó đường viền hai vùng sẽ được nhận dạng một cách dễ dàng hơn.

Ví dụ minh họa được nhắc tới trong vùng 2.18. Ảnh ban đầu có hai khu vực được điền với những cấu trúc khác nhau. Cấu trúc bao gồm những vết đen lớn ta gọi là cấu trúc lớn, và cấu trúc bao gồm những vết đen nhỏ ta gọi là cấu trúc nhỏ. Phép đóng đầu tiên sẽ xoá đi vùng chứa cấu trúc nhỏ và nó tạo ra một vùng đen đặc, nơi mà chứa cấu trúc lớn trước đây. Khi đó thủ tục trích biên hình thái được áp dụng và nó tạo ra cho ta một đường biên liền nét giữa hai vùng cấu trúc. Ta sẽ thấy xuất hiện những vết nhoè nằm rải trên đường liền nét, đó chính là những điểm đen trong cấu trúc lớn bị cắt.



**Hình 2.18:** Phân đoạn cấu trúc

- (a) ảnh được phân đoạn Sau khi đóng bằng cỡ của những vết trong cấu trúc nhỏ  
 (b) Sau khi đóng bằng kích cỡ của khoảng cách giữa các vết của cấu trúc lớn (c)  
 (d) Viền của (c) (e) Viền trong ảnh gốc (a).

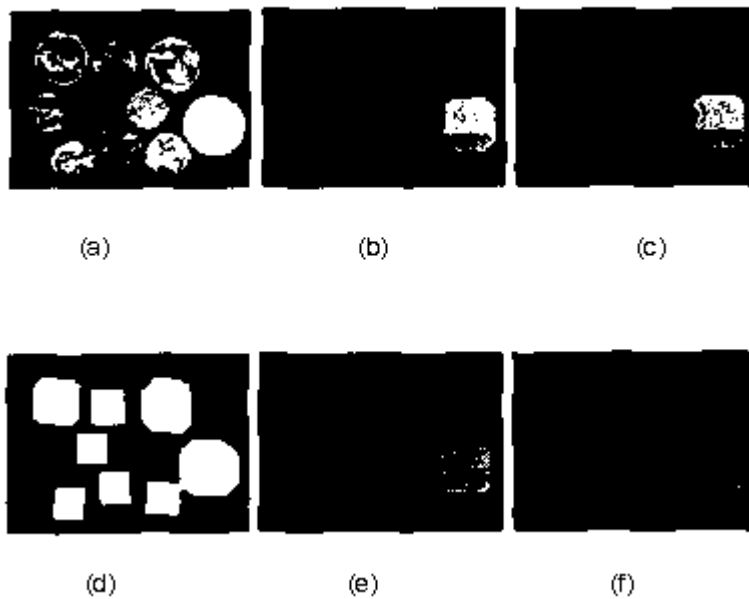
## 2.26. Phân loại cỡ đối tượng.

Sử dụng Hình thái học cho việc phân vùng bằng cấu trúc đưa ta tới một ứng dụng khác của phép toán hình thái - đó là sự phân lớp đối tượng dựa trên kích cỡ hoặc hình dạng của chúng. Khi nào mà việc sử dụng hình dáng cho phép toán hình thái đòi hỏi phải thử nghiệm ít mẫu cấu trúc (phần tử cấu trúc) thì việc phân cỡ của chúng vẫn còn được quan tâm. Một tập hợp các đối tượng khác nhau được phân lớp một cách có quy tắc dựa theo kích cỡ của chúng, từ rất nhỏ như các vi sinh vật đến các vật lớn hơn như quả trứng, quả táo... và người ta cũng có thể tạo ra một chương trình cho việc phân loại trứng bằng Hình thái học đa cấp xám. Tuy nhiên do trứng gà còn có một cách phân loại tốt hơn đó là phân loại theo trọng lượng, nên ta hãy bỏ qua chúng và xét một trường hợp gần gũi với ta hơn, đó là tiền.

Thật là ngẫu nhiên, mà những đồng xu thường có giá trị lớn hơn những đồng xu bé. Chẳng hạn một đồng hào thì nhỏ hơn đồng đôla xu. Hình 2.19 a miêu tả một ảnh của một tập hợp những đồng xu trên một nền tối. Nó được trộn lẫn giữa xu Canada và Mỹ.

Sử dụng phép mở đa cấp xám sẽ làm mức xám của một đối tượng và ảnh được mở một cách từ từ cùng với việc tăng dần bán kính của cấu trúc tròn (đã từng nói trước đây). Tại một vài điểm, khi bán kính của phần tử cấu trúc lớn hơn bán kính của đồng xu thì đồng xu khi đó sẽ được xoá khỏi ảnh, ở đây ta sử dụng bán kính trong phạm vi 5 - 14; phép mở bằng một cấu trúc tròn với bán kính 14 sẽ xoá đi hết những đồng xu, để lại một ảnh tối và trống.

Sự thay đổi đầu tiên xuất hiện đó là khi sử dụng bán kính 6, 5 (đường kính 13), khi đó những đồng hào bị nhỏ dần cho đến khi phép phân ngưỡng đủ xoá chúng đi. Tăng dần bán kính cho đến khi phép mở sử dụng bán kính 10 chỉ để lại những đồng đôla là những đồng có kích cỡ to nhất. Bằng việc đếm số vùng bị biến mất sau mỗi phép lặp (thường dễ dàng hơn nếu đếm trực tiếp các đồng xu cùng cỡ) có thể giúp ta ước tính được tổng giá trị của những đồng xu trong ảnh. Ở nhiều nước, tiền giấy cũng có giá trị tùy theo cỡ và việc phân lớp các hồi phiếu theo cỡ tiền cũng thật quan trọng...



Hình 2.19: Phân lớp những đồng xu

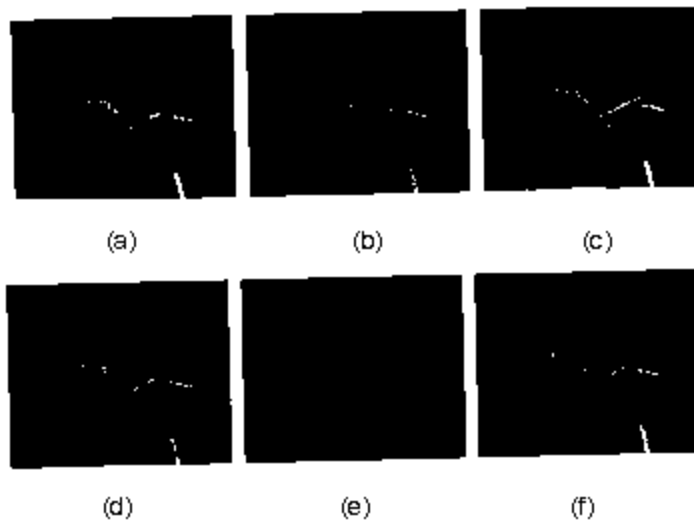
- a. ảnh những đồng xu cần phân lớp
- b. Sau khi mở bằng cấu trúc bán kính 6
- c. Sau khi mở bằng cấu trúc bán kính 6.5
- d. ảnh (c) sau khi phân ngưỡng thấy đồng hào có thể bị xoá
- e. Sau khi mở bằng bán kính 8; nhận thấy đồng xu đã bị xoá
- f. Sau khi mở bằng bán kính 10; chỉ còn lại đồng 1 đồng.

### 2.3. Thao tác trên ảnh màu

Màu có thể được dùng theo hai cách. Chúng ta có thể giả thiết rằng sự tồn tại của ba màu thành phần (red, green, blue) là sự mở rộng dựa trên mức xám, mỗi màu có thể được coi như một miền giá trị độc lập chứa thông tin mới (chẳng hạn có 256 mức cho red, 256 mức cho blue, 256 cho green). Ta xét một ví dụ đơn giản của phép toán hình thái trên ảnh màu.

Hình 2.20a là ảnh xám được lấy ra từ một ảnh màu. Nó chụp cảnh một con châu chấu bám trên một chiếc lá. Cả hai, châu chấu và nền hầu như cùng màu nên việc xác định đâu là châu chấu đâu là nền quả là khó khăn một chút. Bằng việc kiểm tra chặt

chẽ trên ba ảnh với ba màu riêng biệt (red, green, blue) thì thấy rằng chúng có một sự khác nhau chút ít: Châu chấu dường như sáng hơn trong ảnh màu đỏ và xanh dương(blue), trái lại trong ảnh green thì nền lại có vẻ như sáng hơn. Phép đóng ảnh đỏ và xanh dương làm sáng châu chấu hơn, phép mở ảnh xanh lá cây sẽ làm cho viền sáng hơn một chút. Một phần tử cấu trúc tròn với bán kính 4 được sử dụng trong mỗi trường hợp. Sau những phép đóng mở, ba thành phần này có vẻ như dần hợp lại thành một ảnh màu.



Hình 2.20: Hình thái học màu

- a. ảnh một con châu chấu
- b. Thành phần Red trong ảnh màu R,G,B
- c. Thành phần Green
- d. Thành phần Blue
- e. ảnh nhận được từ việc đóng thành phần Red và Blue và mở phần Green
- f. ảnh gốc được che với một ảnh được xử lý cho ta thấy hình con châu chấu.

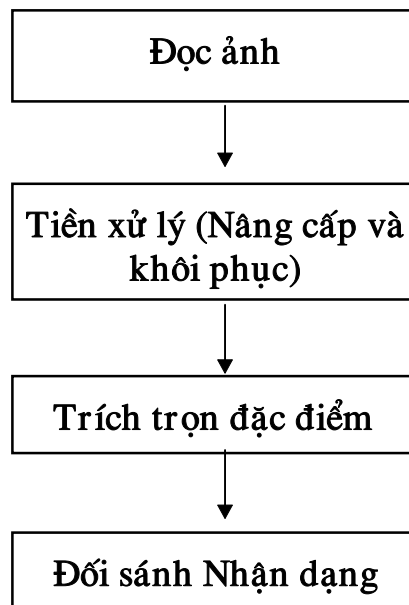


## CHƯƠNG III

### ỨNG DỤNG CỦA HÌNH THÁI HỌC

#### 3.1. Ứng dụng thực tiễn

Trong xử lý ảnh và nhận dạng ảnh, có một số loại ảnh đường nét gồm các đối tượng (objects) là các đường cong có độ dài lớn hơn nhiều so với độ dày của nó, ví dụ như là ảnh các kí tự, dấu vân tay, sơ đồ mạch điện tử, bản vẽ kĩ thuật, bản đồ v.v... Để xử lý các loại ảnh này người ta thường xây dựng các hệ mô phỏng theo cách phân tích ảnh của con người gọi là hệ thống thị giác máy (Computer Vision System). Có nhiều hệ thống được cài đặt theo phương pháp này (xem hình 3.1) như hệ thống nhận dạng chữ viết bằng thiết bị quang học OCR (Optical Character Recognition ), hệ thống nhận dạng vân tay AFIS (Automated fingerprint Identification System) v.v..



**Hình 3.1 :** Mô hình tổng quát của hệ thống nhận dạng ảnh

Có nhiều phương pháp trích chọn đặc điểm được biết tới như phương pháp sử dụng sóng ngắn (Wavelet), sử dụng hệ số Fourier, sử dụng các mô men bất biến, sử dụng các đặc trưng của biên như tính trơn và các điểm đặc biệt, sử dụng các đặc trưng tô pô dựa trên xương của đường nét... Phương pháp trích chọn đặc điểm sử dụng ảnh đã mảnh được sử dụng nhiều vì việc trích chọn đặc điểm trở nên dễ dàng. Sau bước này các đường nét đã mảnh được vectơ hoá ảnh phục vụ việc nén dữ liệu, nhằm giảm thiểu yêu cầu về không gian lưu trữ, xử lý và thời gian xử lý. Kỹ thuật *làm mảnh* là một trong nhiều ứng dụng của phép toán hình thái học (Morphology).

Thông thường các thuật toán làm mảnh thường bao gồm nhiều lần lặp, trong mỗi lần lặp tất cả các điểm của đối tượng sẽ được kiểm tra nếu như chúng thoả mãn điều kiện xoá nào đó tùy thuộc vào thuật toán thì nó sẽ bị xoá đi. Quá trình được lặp lại cho đến khi không còn điểm biên nào được xoá. Đối tượng được bóc dần lớp biên cho đến khi bị thu mảnh lại thành một đường duy nhất có bề dày 1 điểm ảnh. Nhưng trong thực tế, chẳng hạn khi sử dụng các phép toán hình thái nhằm lấp đầy các lỗ hổng, làm trơn biên và nối số đường đứt nét, đôi khi ta chỉ bóc một số lớp nhất định để làm mảnh đối tượng đến một độ nhất định và bản thân trong mỗi phần trong cùng một ảnh lại cần làm mảnh với một số lớp khác nhau.

### 3.2. Xương và làm mảnh

Như chúng ta đã biết rằng, phép toán dẫn nở ảnh cho phép lấp đầy các lỗ hổng, làm trơn biên và nối một số đường đứt nét. Sau giai đoạn nối các đường đứt nét cần giảm độ dày của đường do phép toán dẫn nở, khi đó phải sử dụng phép co. Trong một số trường hợp thì nhược điểm của phép co ảnh là làm đứt nét các đường, làm mất tính liên thông của đường. Do đó ta phải tìm cách khắc phục nhược điểm đó và phép toán làm "mảnh ảnh" hay "tìm xương" đã ra đời. Đó là một trong nhiều ứng dụng của phép toán Hình thái học (Morphology), mà trong Đề án này ta muốn sơ qua.

Mọi người đang làm việc trong lĩnh vực thị giác máy tính đều biết làm mảnh (thinning) là gì ?. Đó là việc bạn phải làm gì để xác định xương (Skeleton) của một đối tượng, thường là của một đối tượng nhị phân. Vậy ta có thể đặt ra một câu hỏi: "Xương là gì ?". Bây giờ chúng ta sẽ tìm hiểu chúng. Như với cấu trúc (texture), không có một định nghĩa chung nào cho khái niệm một xương là gì ?. Và tệ hơn

không giống với cấu trúc, chúng ta có thể không nhận biết được một xương khi chúng ta nhìn thấy nó. Đây là một điều đáng tiếc bởi vì sự phát sinh của một xương số (digital skeleton) thường là một trong các bước xử lý đầu tiên thực hiện bởi một hệ thống thị giác máy khi thử trích ra các đặc tính (feature) từ một đối tượng trong một ảnh. Một xương được xem như dùng để mô tả hình dạng của đối tượng theo một số ít các điểm ảnh có liên quan, tất cả các điểm ảnh đó (trong một vài khả năng) thuộc về cùng một cấu trúc (structural) và do đó nó rất cần thiết. Trong các ảnh đoạn, xương truyền đạt tất cả các thông tin được thấy trong ảnh nguyên bản ban đầu, trong xương các giá trị như: Vị trí, phương hướng và độ dài của các đoạn thẳng chính là những đặc trưng cốt yếu của các đường trong ảnh ban đầu. Điều này đơn giản hoá việc biểu diễn các bộ phận của ảnh đoạn. Do đó *làm mảnh ảnh có thể được định nghĩa như là hoạt động của việc nhận dạng (identifying) các điểm ảnh của một đối tượng mà các điểm ảnh đó là các điểm cốt yếu cho việc mô tả hình dạng của đối tượng*: Đó là các điểm xương và các điểm xương đó tạo thành một tập các điểm xương; hay nói cách khác làm mảnh có thể được xem là việc đồng nhất các điểm ảnh của một đối tượng mà các điểm ảnh đó chứa thông tin về hình dáng của đối tượng, các điểm ảnh này được gọi là các điểm xương ảnh, và là một bộ mẫu. Cũng có một số xương dạng số được đưa ra nhưng vẫn chưa hoàn toàn được chấp nhận. Hàng trăm bài báo dựa trên chủ đề của việc làm mảnh được in ấn; phần lớn chúng quan tâm đến việc thực hiện một sự thay đổi (biến tấu) trên một phương pháp làm mảnh đang tồn tại, trong đó các hướng mới lạ được liên kết (liên hệ) cho việc thực hiện thuật toán. Nhiều thuật toán làm mảnh gần đây được thiết kế với một tốc độ đáng kể. Tốc độ của thuật toán được cải tiến không ngừng; Chất lượng của xương cũng ngày càng được cải tiến.

Trong chương này một số tiếp cận về làm mảnh ảnh sẽ được khảo sát (xem xét) và chúng ta sẽ luôn trở lại kết quả nguyên bản của định nghĩa ngoại trừ việc tìm kiếm một cách giải quyết. Tuy nhiên, có 3 điều cần lưu ý về làm mảnh:

- 1) Không phải tất cả các đối tượng đều có thể và phải được làm mảnh, việc làm mảnh là hữu dụng (có ích) cho các đối tượng ăn khớp của các đoạn, nghĩa là chúng chỉ thẳng hoặc cong và việc làm mảnh là không hữu dụng (không có ích) cho các đối tượng có hình dạng khép kín một vùng. Ví dụ, một đường cong có thể được làm mảnh nhưng một hình đĩa không thể làm mảnh một cách đầy đủ.

2) Những gì hoạt động như là một xương trong tình huống này có thể không hoạt động trong tình huống khác. Làm mảnh thường là một bước chuẩn bị một ảnh cho các bước xử lý tiếp theo. Tất nhiên, các bước tiếp theo sau thường làm việc với các đặc trưng (thuộc tính) cần thiết của xương.

3) Làm mảnh là hoạt động của việc nhận dạng xương và không được xác định bằng thuật toán đã dùng. Đặc biệt, việc làm mảnh không phải luôn luôn làm công việc xử lý lặp lại việc lột bỏ đi lớp bên ngoài của các điểm ảnh.

### 3.3. Các phương pháp lặp hình thái học

Phần lớn các thuật toán làm mảnh dựa trên một vòng lặp lột bỏ dần đi các lớp điểm ảnh cho đến khi không còn nhiều hơn một lớp được xoá bỏ. Có một tập các quy tắc để xác định các điểm ảnh cần loại bỏ và thông thường một vài dạng của cấu trúc mẫu phù hợp (template-matching) được dùng để thực hiện các quy tắc đó. Thông thường các quy tắc được thiết kế sao cho dễ dàng nhận biết được khi nào thì kết thúc: Đó là khi không có sự thay đổi nào xảy ra sau 2 lần duyệt qua ảnh.

Thuật toán đầu tiên (thuật toán Stentiford) được đề xuất năm 1983 là điển hình của kiểu này. Nó sử dụng các mẫu 3x3 và cách thức hoạt động của nó như sau: Di mẫu trên ảnh, nếu như mẫu phù hợp ảnh thì loại bỏ (đặt trắng) điểm ảnh trung tâm.

Thuật toán cơ bản như sau:

1) Tìm một vị trí điểm ảnh  $(i, j)$ , vị trí mà các điểm ảnh trong ảnh I phù hợp với các điểm ảnh trong mẫu M1 (Hình 3.2a).

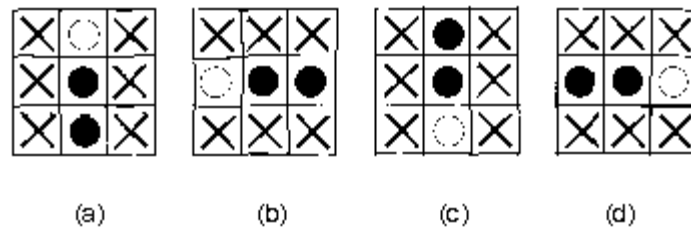
2) Nếu điểm ảnh trung tâm không phải là điểm cuối (endpoint) và có giá trị liên kết là 1 thì đánh dấu điểm này cho lần xoá sau đó.

3) Lặp lại bước 1 và 2 cho tất cả các vị trí điểm ảnh phù hợp với mẫu M1.

4) Lặp lại bước 1-3 lần lượt cho các mẫu còn lại: M2, M3 và M4.

5) Nếu bất kỳ điểm ảnh nào được đánh dấu cho thao tác xoá bỏ thì xoá chúng bằng cách tạo cho chúng thành màu trắng.

6) Nếu bất kỳ điểm ảnh nào đã được xoá ở bước 5) thì lặp lại toàn bộ quá trình xử lý từ bước 1), còn không thì thuật toán dừng.



**Hình 3.2:** Các mẫu dùng cho việc nhận dạng những điểm ảnh có thể bị xoá trong thuật toán làm mảnh Stienford

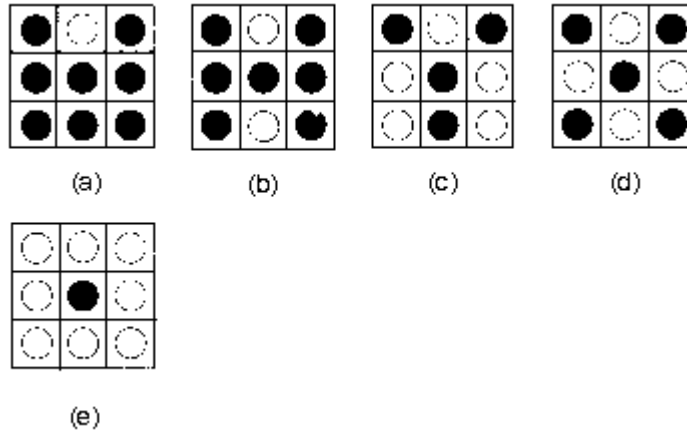
- a. Mẫu M1
- b. Mẫu M2
- c. Mẫu M3
- d. Mẫu M4

Những điểm đen và trắng xác định trong mẫu phải tương ứng những điểm cùng màu trong ảnh; Những điểm X quyết định chỗ nào trong ảnh ta không cần quan tâm tới màu của nó.

Ảnh phải được quét theo một thứ tự riêng biệt đối với từng mẫu. Chức năng của mẫu M1 là tìm các điểm ảnh có khả năng được xoá dọc theo cạnh trên cùng của đối tượng và chúng ta tìm kiếm cho một sự phù hợp từ trái sang phải, sau đó từ trên xuống dưới. Mẫu M2 phù hợp với một điểm ảnh phía bên trái của một đối tượng, mẫu này xoá từ dưới lên trên ảnh, từ trái sang phải. Mẫu M3 sẽ định vị các điểm ảnh dọc theo cạnh dưới và xoá từ phải sang trái, từ dưới lên trên. Cuối cùng, tìm các điểm ảnh có thể xoá phía bên phải của đối tượng, phù hợp với mẫu M4 tính từ trên xuống dưới, từ phải sang trái. Phương hướng và thứ tự xác định này áp dụng cho các mẫu đảm bảo rằng các điểm ảnh sẽ bị xoá theo cách đối xứng mà không cần bất cứ một đường chéo định hướng nào.

Có hai vấn đề tiếp tục cần giải quyết mà cả 2 vấn đề này đều xuất phát từ bước 2. Một điểm ảnh là một điểm cuối (endpoint) nếu nó chỉ được liên kết với một điểm ảnh khác, nghĩa là, nếu một điểm ảnh đen chỉ có một điểm đen bên ngoài 8-láng giềng của nó. Nếu các điểm cuối bị xoá thì bất kỳ các đường thẳng và các đường cong mở nào cũng sẽ bị xoá hoàn toàn, điều này phần nào giống như việc mở một dây khoá quần áo(zipper).

Khái niệm số liên kết (connectivity number) là một chút thách thức hơn cho chúng ta. Bởi vì chúng ta chỉ đang sử dụng các phần rất nhỏ của một ảnh. Vai trò của các đoạn ảnh đó trong toàn bộ bức ảnh không được rõ ràng. Đôi khi, một điểm ảnh đơn kết nối 2 phần lớn hơn của một đối tượng và đó là trực giác tất nhiên mà như vậy một điểm ảnh không thể được xoá. Để làm được như vậy ta sẽ phải tạo 2 đối tượng trong đó chỉ có một đối tượng nguyên bản.



**Hình 3.3:** Một minh hoạ về số liên kết

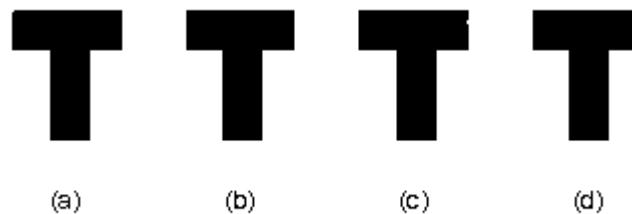
- a. Điểm trung tâm không liên kết với bất cứ vùng nào và có thể bị xoá. Số liên kết bằng 1
- b. Nếu điểm trung tâm đã bị xoá, hai phần trái và phải sẽ trở thành không liên kết. Số liên kết bằng 2
- c. Số liên kết bằng 3
- d. Số liên kết bằng 4, cực đại
- e. Liên kết bằng 0.

Số kết nối chính là một sự đo lường xem có bao nhiêu đối tượng mà một điểm ảnh có thể kết nối. Một cách đo lường các kết nối, được thấy như trong hình 3.3 (đẳng thức Yokoi 1973) là:

$$C_n = \sum_{k \in S} N_k - (N_k * N_{k+1} * N_{k+2}) \quad (Eq 3.1)$$

Trong đó  $N_k$  là giá trị màu của một trong các 8\_láng giềng của điểm ảnh được liên kết và  $S = \{1, 3, 5, 7\}$ .  $N_1$  là giá trị màu của điểm ảnh bên phải của điểm ảnh trung tâm và chúng được số hoá theo thứ tự ngược chiều kim đồng hồ, xung quanh điểm ảnh trung tâm. Giá trị của  $N_k$  là 1 nếu điểm ảnh là điểm trắng (Điểm ảnh nền) và giá trị của  $N_k$  là 0 nếu điểm ảnh là điểm đen (điểm ảnh thuộc đối tượng). Điểm ảnh trung tâm là  $N_0$  và  $N_k = N_k - 8$  nếu  $k > 8$ . Một cách khác mà giá trị liên kết có thể được tính toán bằng cách xét các điểm láng giềng theo thứ tự:  $N_1, N_2, \dots, N_8, N_1$ . Số các thay đổi màu (đen-trắng) được dùng để đếm số vùng điểm ảnh trung tâm kết nối.

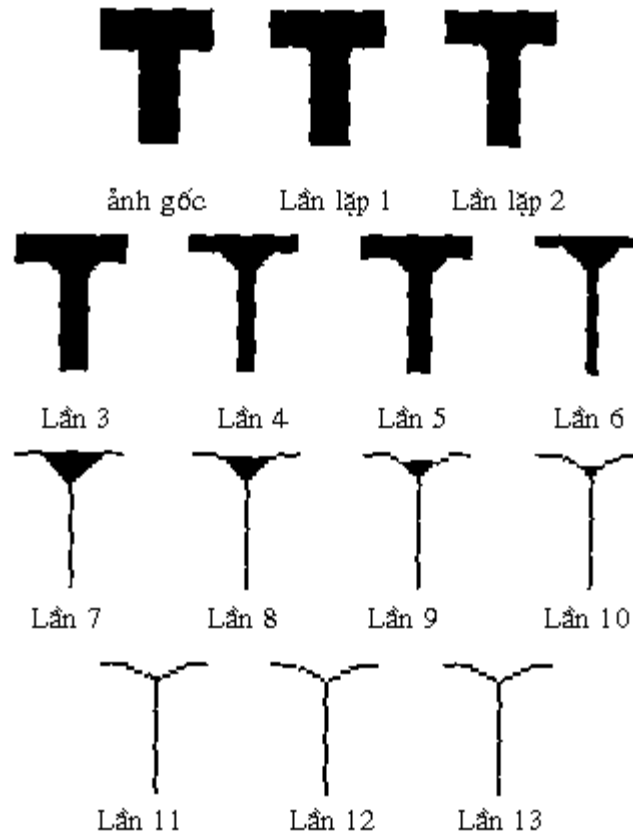
Hình 3.4 trình bày một vòng lặp (đầu tiên) của thuật toán làm mảnh áp dụng cho đối tượng có hình dạng chữ T. Một vòng lặp bao gồm một quá trình duyệt qua đối với mỗi mẫu trong 4 mẫu đã cho. Các điểm đen được đánh dấu cho thao tác xoá và điều đó dễ nhận ra trong sơ đồ một cách chính xác những gì mỗi mẫu thực hiện. Mỗi vòng lặp hoàn thành có hiệu quả xói mòn một lớp các điểm ảnh từ bên ngoài của đối tượng nhưng không giống với phép co ảnh hình thái chuẩn, việc xoá bỏ của một điểm ảnh không làm mất tính liên thông.



**Hình 3.4:** Bốn phần của mỗi phép lặp trong phương pháp làm mảnh Stentiford

- a. Sau khi áp dụng mẫu M1
- b. Sau mẫu M2
- c. Sau M3
- d. Sau M4. Trong mỗi trường hợp, những điểm đen đại diện cho chúng bị xoá trong lần lặp này.

Để hoàn chỉnh việc làm mảnh đối tượng này đòi hỏi 13 vòng lặp (việc đếm vòng lặp cuối cùng mà không có thao tác nào ngoại trừ những hiển thị cho chúng ta kết thúc). Hình 3.5 trình bày ảnh kết quả sau mỗi vòng lặp.



**Hình 3.5:** Tất cả các phép lặp của thuật toán làm mảnh Stienford được áp dụng cho chữ T. hai vòng lặp cuối tương tự, thêm một lần quét phụ đảm bảo rằng xương hoàn chỉnh.

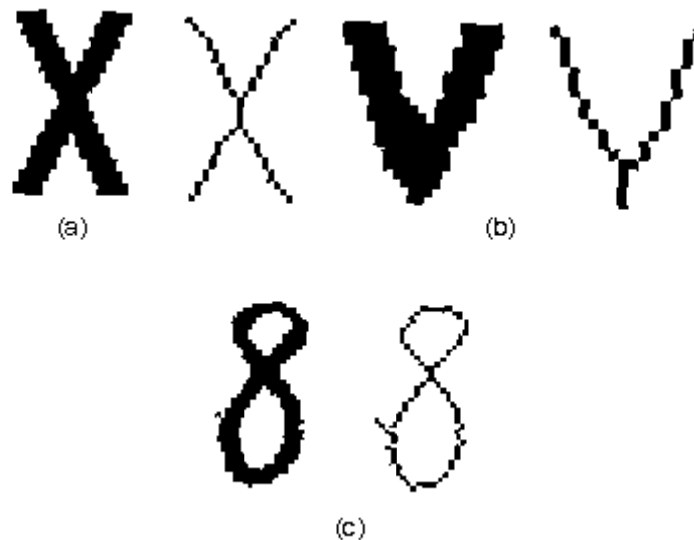
Một vòng lặp thực hiện 4 lần duyệt ảnh mà trong trường hợp này duyệt qua 60x60 điểm ảnh hay 3600 điểm ảnh. Như vậy, 187, 000 điểm ảnh đã được kiểm tra chỉ để làm mảnh một ảnh đơn giản này. Điều đó trở nên tồi tệ hơn: Mỗi quá trình áp dụng mẫu xem xét kiểm tra 3 điểm ảnh và mỗi lần có sự phù hợp mẫu xảy ra, 18 điểm ảnh khác được xem xét kiểm tra (giới hạn trên là: 10108800 điểm ảnh, nhưng chỉ có một phần trong chúng được kiểm tra trong thực hành). Cuối cùng, sẽ có thêm một quá trình duyệt mỗi vòng lặp để xoá các điểm ảnh đã đánh dấu(10, 102, 000 ).



Đây là một cách làm tốn kém để làm mảnh một ảnh nhỏ nhưng là phương pháp điển hình hoàn chỉnh của các thuật toán đánh dấu và xoá mẫu cơ bản.

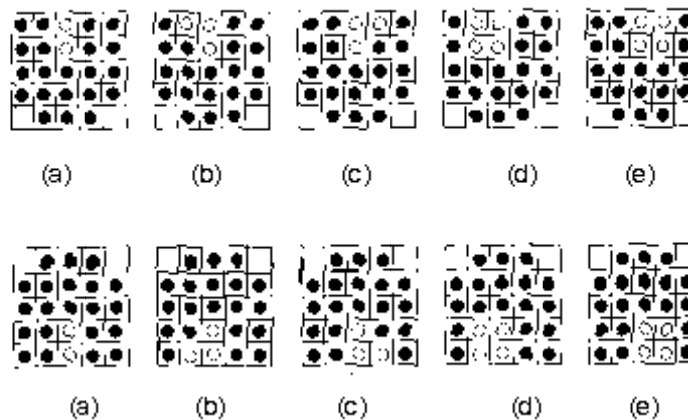
Có một vài vấn đề cố hữu cùng với thuật toán làm mảnh này mà chúng trình bày dưới đây như như là các tạo tác trong xương. Chúng là cố hữu bởi vì chúng có khuynh hướng xuất hiện trong rất nhiều thuật toán kiểu này, các nhà nghiên cứu trong lĩnh vực này đã nhận thức được để đoán nhận chúng.

Thuật toán đầu tiên được gọi là “necking” mà trong đó một điểm hẹp ở giao điểm của hai đường thẳng được kéo dài ra thành một đoạn thẳng nhỏ (hình 3.6a). Các phần đuôi có thể được tạo nơi không tồn tại do việc làm mảnh quá mức nơi hai đường gặp nhau ở một góc nhọn (hình 3.6b). Cuối cùng, có lẽ phổ biến, là sự khởi tạo của các đoạn thẳng phụ ngoài để chấp nối một đoạn xương thực sự. Nó được gọi là một phép chiếu giả mạo, những sợi tóc (Hình 3.6).



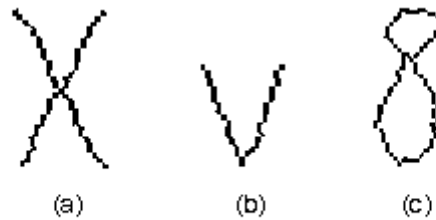
Stentjord đề nghị một giai đoạn tiền xử lý để cực tiểu hoá các chế tác làm mảnh đó. Do bởi các đường sơ thường được tạo ra bởi những bất quy tắc nhỏ theo đường biên ngoài của đối tượng, nên phải tiến hành làm trơn trước khi làm mảnh để xóa bỏ chúng. Điều cơ bản là một quá trình duyệt được thực hiện trên tất cả các điểm ảnh, xoá bỏ các điểm ảnh có hai hoặc ít hơn các điểm láng giềng đen và có một giá trị liên kết nhỏ hơn 2.

Để xử lý với “necking”, ông đề nghị một thủ tục được gọi là thủ tục phân giác góc nhọn (acute angle amphasis), mà trong đó các điểm ảnh gần khớp nối giữa hai dòng được tạo thành màu trắng nếu chúng khép lại tạo thành một góc nhọn. Điều này được thực hiện bằng cách dùng mẫu như đã thấy trong hình 3.7. Một sự phù hợp với bất kì mẫu nào đánh dấu điểm ảnh trung tâm cho thao tác xoá và tạo ra vòng lặp khác của một số ít các phân giác góc nhọn quan trọng chỉ dùng ba mẫu đầu tiên của mỗi kiểu. Nếu bất kỳ điểm ảnh nào đã được xoá bỏ, một lần duyệt cuối cùng chỉ dùng các mẫu đầu tiên của mỗi kiểu được thực hiện.



úc nhọn.

Làm trơn (Smoothing) được hoàn thành đầu tiên, tiếp theo là tất cả các quá trình duyệt qua ảnh của các phân giác góc nhọn. Cuối cùng là các bước làm mảnh ảnh. Hình 3.8 trình bày các xương kết quả cuối cùng của các ký tự trong hình 3.8. Khi các bước tiền xử lý được gộp vào.



**Hình 3.8:** Những kí tự được làm mảnh cuối cùng, sau hai bước xử lý và làm mảnh.

Hơn nữa, hầu hết các xương đó xuất hiện khi dùng phương pháp này vẫn bị rạn nứt. Cách dùng 3 giai đoạn của các phân giác góc nhọn sẽ không hiệu quả đối với các ký tự rất dày, và các mẫu không phù hợp với tất cả các tình huống mà có thể gây ra cổ cột và đuôi cột. Cũng như vậy, bước làm tron sẽ không bắt gặp các bất quy tắc mà các bất quy tắc này có thể tạo nên các đường sơ.

Mặc dù vậy, việc hoàn chỉnh thuật toán sẽ không được như mong đợi và phương pháp là tương đối tốt, đặc biệt là bước tiền xử lý cho việc nhận dạng ký tự.

Một thuật toán làm mảnh đường như là công cụ cho mọi người, đó là thuật toán Zhang\_Suen(Zhang 1984). Thuật toán này được sử dụng như một nền tảng cơ sở cho việc so sánh các thuật toán làm mảnh trong nhiều năm, và nó nhanh, đơn giản khi thực hiện. Thuật toán này là một phương pháp song song, có nghĩa là giá trị mới cho bất kỳ điểm ảnh nào có thể được tính toán chỉ dùng các giá trị đã biết từ trong vòng lặp trước. Do đó, nếu máy tính có một CPU cho mỗi điểm ảnh đã được cung cấp trước, nó có thể xác định toàn bộ quá trình lặp tiếp theo một cách đồng thời. Vì hầu hết chúng ta không có một máy tính có kích cỡ như vậy, do đó, chúng ta chỉ xem xét phiên bản của chương trình mà nó chỉ dùng 1 CPU.

Thuật toán được ngắt thành hai vòng lặp con, ví dụ, thay vì 4 vòng lặp con của thuật toán Stentiford. Trong một vòng lặp con, một điểm ảnh  $I(i, j)$  được xoá (hay được đánh dấu cho thao tác xoá bỏ) nếu 4 điều kiện sau đây được thoả mãn:

- 1) Giá trị liên kết của nó là 1.
- 2) Nó có 2 điểm láng giềng đen nhỏ nhất và không lớn hơn 6.
- 3) Một trong các điểm đen nhỏ nhất:  $I(i, j+1)$ ,  $I(i-1, j)$  và  $I(i, j-1)$  là điểm nền(điểm màu trắng).

4) Một trong các điểm nhỏ nhất:  $I(i-1, j)$ ,  $I(i+1, j)$  và  $I(i, j-1)$  là điểm ảnh nền. Tại cuối vòng lặp con này các điểm đã đánh dấu được xoá bỏ.

Vòng lặp con tiếp theo sau làm tương tự ngoại trừ bước 3 và 4.

1) Một trong các điểm đen nhỏ nhất:  $I(i, j+1)$ ,  $I(i-1, j)$  và  $I(i, j-1)$  là điểm nền(màu trắng).

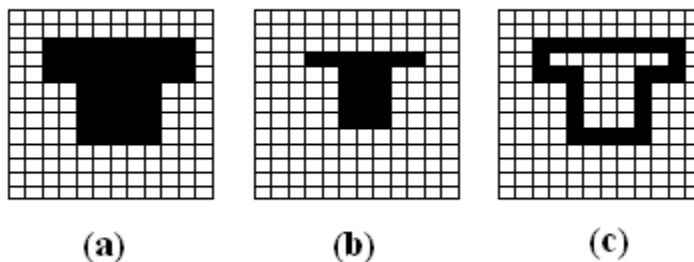
2) Một trong các điểm nhỏ nhất  $I(i-1, j)$ ,  $I(i+1, j)$  và  $I(i, j-1)$  là điểm ảnh nền. Trở lại, bất kỳ điểm ảnh nào đã đánh dấu đều được xoá bỏ.

Nếu ở cuối vòng lặp con khác không có điểm nào được xoá thì xương hoàn toàn được xác định và chương trình kết thúc.

### 3.4. Nhận dạng biên

Những điểm ảnh trên biên của một đối tượng là những điểm ảnh trên biên mà có ít nhất một điểm ảnh lân cận thuộc nền. Do bởi lân cận nền cụ thể là không biết trước mà phải tìm, và lại không thể tạo ra được một cấu trúc đơn mà cho phép phép co hoặc phép dẫn dò ra biên, mặc dầu rằng trong thực tế, một phép co bởi phần tử cấu trúc đơn giản chính xác là có thể xoá những điểm biên. Mặt khác ta lại có thể áp dụng điều này để thiết kế một phép toán hình thái dò biên. Biên có thể được tách ra bằng cách sử dụng một phép co và ảnh được co sau đó được trừ đi bởi ảnh gốc. Tương tác này sẽ để lại cho ta những điểm ảnh mà được co, đó chính là biên. Điều này được viết như sau:

$$\text{Biên} = A - (A \ominus \text{Cấu trúc đơn giản})$$



**Hình 3.9:** Kết quả làm mảnh

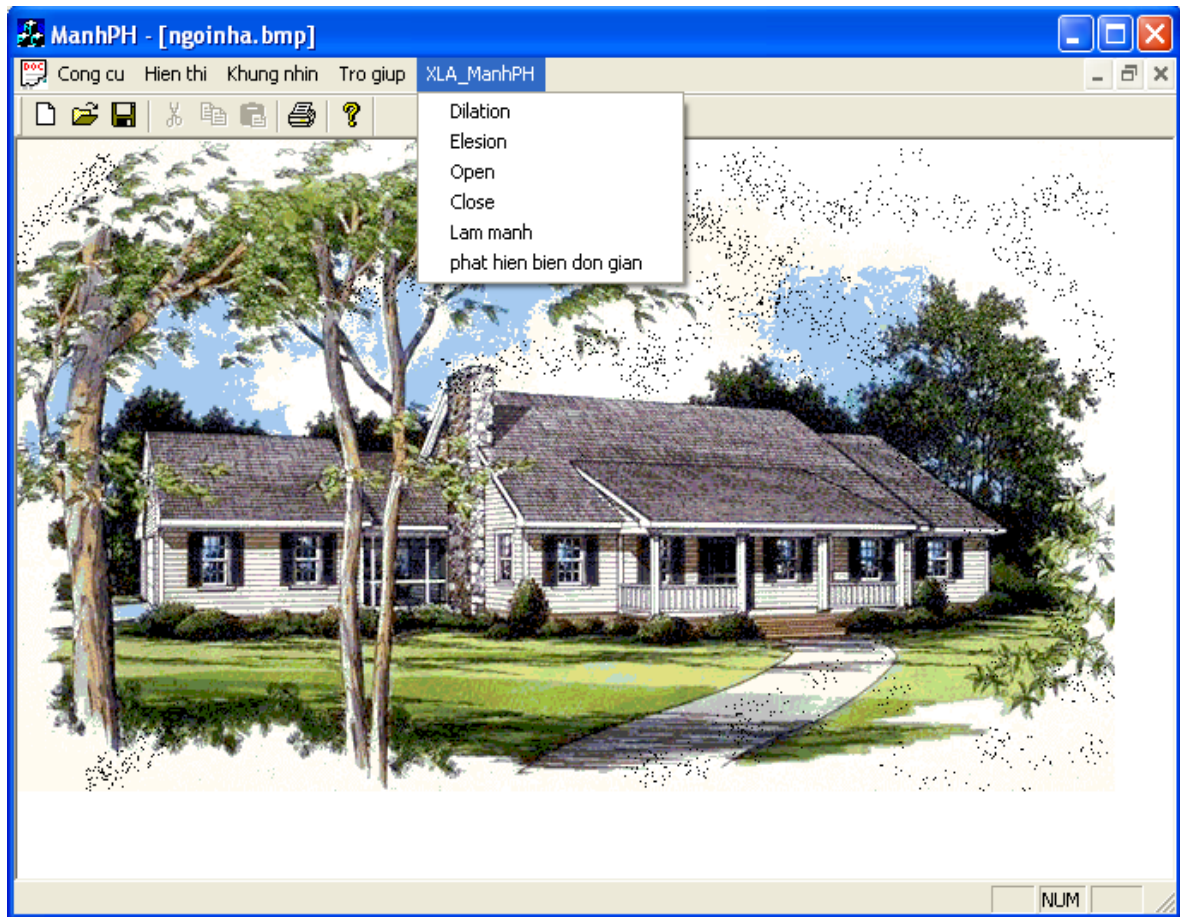
- a) Ảnh ban đầu
- b) Áp dụng Erosion
- c) Ảnh ban đầu – đi ảnh đã biến đổi

## CHƯƠNG IV:

### CÀI ĐẶT

Cài đặt thử nghiệm các phép toán hình thái : co ảnh, dẫn ảnh ,open , close , làm mảnh và phát hiện biên trên ngôn ngữ Vidual C.

Giao diện chương trình



## Một số kết quả chương trình đạt được

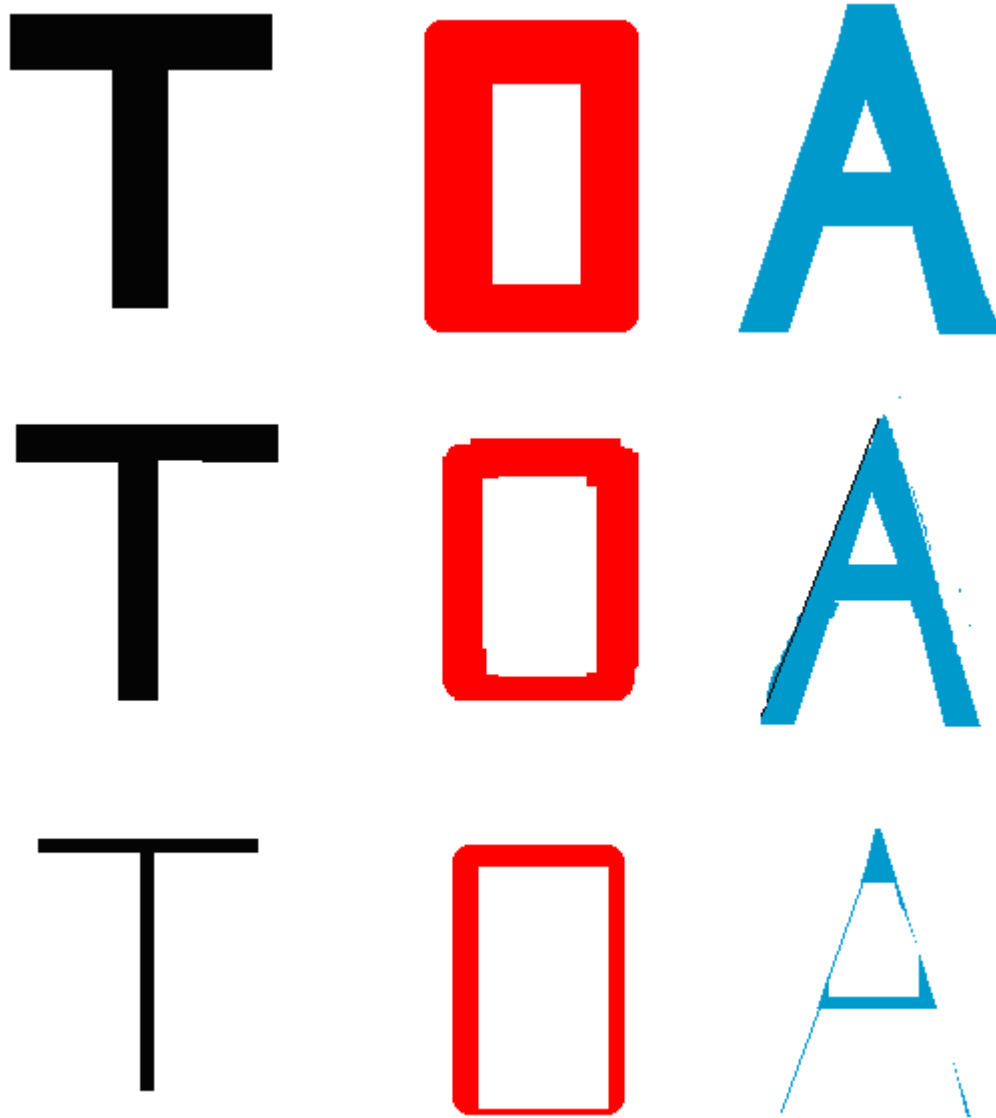
### ❖ Kết quả của phép đóng(close)



### ❖ Kết quả của phép mở(open)



❖ Kết quả của làm mảnh



## KẾT LUẬN

Đồ án đã có một cách nhìn tổng quát về Morphology và cách sử dụng chúng.

★ Sơ lược về xử lý ảnh và Morphology đã nêu ra :

- Hệ thống xử lý ảnh tổng quát
- Các giai đoạn của quá trình xử lý ảnh
- Khái niệm về xử lý ảnh, Morphology, ảnh nhị phân

★ Thao tác với Morphology giới thiệu :

- Định nghĩa các phép toán hình thái như : dẫn, co, đóng, mở, nhận dạng biên, đánh trùng, đánh trượt...

- Cách sử dụng của các thao tác, chủ yếu đối với ảnh nhị phân và ảnh xám, có kèm các minh hoạ về sử dụng chúng.

★ Ứng dụng của Morphology trình bày :

- Ứng dụng của Morphology trong thực tiễn  
- Trình bày một ứng dụng quan trọng của Morphology là làm mảnh và phát hiện biên.

★ Cài đặt thử nghiệm chương trình :

- Minh hoạ một cách chi tiết một số thao tác hình thái học như: Dẫn, co, đóng, mở, làm mảnh, nhận dạng biên.

Đồ án có thể là tài liệu tham khảo cho những người bắt đầu tìm hiểu về xử lý ảnh nói chung và các thao tác hình thái học (Morphology) nói riêng, giúp họ có được những khái niệm, đánh giá sơ đẳng, thuận tiện cho việc nghiên cứu sau này.

Vì thời gian có hạn nên Đồ án chỉ đề cập đến một số thao tác, khái niệm cơ bản, ứng dụng chủ yếu trong ảnh nhị phân và ảnh xám do đó không thể tránh khỏi những sơ suất và thiếu sót.

Rất mong nhận được sự thông cảm và góp ý thêm.

Em xin chân thành cảm ơn !



## TÀI LIỆU THAM KHẢO

1. Dougherty, E.R. 2004. *An Introduction to Morphological Image Processing*. Bellingham, WA: SPIE Press.
2. Bạch Hưng Khang, Lương Chi Mai, Ngô Quốc Tạo, Đỗ Năng Toàn, et al., *An Examination of Techniques for Raster to Vector Process and Its Implementation Mapscan Package Software, International Symposium, AMPST 96, University of Bradford, UK, 26-27 March, 1996.*
3. Toumasset J.J., *Traitement de l'Image par Exemple, Symbex, Chapitre 5, Images Binaires Operateurs Morphologiques, pp.117-139, 1990.*
4. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing,"
5. *Second Edition, Prentice Hall*
6. [www.ieeexplore.ieee.org](http://www.ieeexplore.ieee.org)
7. [www.ultra.obuda.kando.hu](http://www.ultra.obuda.kando.hu)
8. [www.img.cs.titech.ac.jp](http://www.img.cs.titech.ac.jp)
9. [www.diendantinhoc.com](http://www.diendantinhoc.com)
10. [www.ddth.com](http://www.ddth.com)
11. [www.itechpro.com](http://www.itechpro.com)
12. <http://ultra.obuda.kando.hu>
13. <http://mydebianblog.blogspot.com>