

## MỤC LỤC

<b>MỤC LỤC</b> .....	<b>1</b>
<b>MỞ ĐẦU</b> .....	<b>3</b>
<b>DANH MỤC HÌNH VẼ</b> .....	<b>4</b>
<b>CHƯƠNG 1: TỔNG QUAN VỀ XỬ LÝ ẢNH</b> .....	<b>5</b>
1.1 Các khái niệm cơ bản trong xử lý ảnh.....	5
1.1.1 Xử lý ảnh là gì.....	5
1.1.2 Ảnh và điểm ảnh .....	5
1.1.3 Quan hệ giữa các điểm ảnh.....	5
1.1.4 Mức xám của ảnh.....	6
1.1.5 Độ phân giải .....	7
1.2 Các phép toán cơ bản trên ảnh nhị phân.....	8
1.2.1 Các phép toán logic.....	8
1.2.2 Các phép toán hình thái học.....	8
1.3 Các giai đoạn chính trong xử lý ảnh.....	15
1.4 Một số ứng dụng cơ bản.....	16
<b>CHƯƠNG 2: XƯƠNG VÀ CÁC KỸ THUẬT TÌM XƯƠNG</b> .....	<b>18</b>
2.1 Khái niệm xương .....	18
2.2 Các hướng tiếp cận trong việc tìm xương .....	18
2.2.1 Tìm xương dựa trên làm mảnh.....	18
2.2.2 Tìm xương không dựa trên làm mảnh.....	19

<b>CHƯƠNG 3: CẮT TIA XƯƠNG VỚI DSE.....</b>	<b>26</b>
3.1 Giới thiệu .....	26
3.2 Phương pháp DCE .....	27
3.2.1 Giới thiệu.....	27
3.2.2 Ý tưởng chính.....	29
3.2.3 Rời rạc hóa đường cong với DCE.....	31
3.2.4 Cắt tia xương với DCE.....	33
3.3 Phương pháp DSE .....	36
3.3.1 Ý tưởng chính.....	36
3.3.2 Các định nghĩa.....	37
3.3.3 Thuật toán DSE .....	39
<b>CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM .....</b>	<b>41</b>
4.1 Môi trường cài đặt .....	41
4.2 Chương trình thực nghiệm.....	41
4.2.1 Giao diện chương trình .....	41
4.2.2 So sánh kết quả tìm xương với các phương pháp DCE.....	41
4.2.3 Hiệu quả của việc sử dụng ngưỡng (threshold) .....	43
<b>KẾT LUẬN.....</b>	<b>45</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>46</b>

## MỞ ĐẦU

Xương có thể xem như việc biểu diễn hình dạng một cách cô đọng trong đó hình dạng có thể khôi phục lại hoàn toàn từ xương. Xương được sử dụng rộng rãi để phân tích hình dạng và nhận dạng đối tượng như tra cứu ảnh và đồ họa máy tính, nhận dạng kí tự, xử lý ảnh và phân tích các hình ảnh sinh học. Các thuật toán tìm xương đã được đưa ra nhưng đều gặp phải những hạn chế tương tự nhau đó là có độ nhạy cảm cao đối với nhiễu đường biên, những biến đổi nhỏ trên đường biên của đối tượng có thể làm thay đổi đáng kể xương nhận được, ảnh hưởng tới độ chính xác của xương. Đồ án trình bày kỹ thuật cắt tia xương bằng phương pháp DSE (Discrete Skeleton Evolution) đã giải quyết những hạn chế nêu trên.

Đồ án gồm bốn chương: Chương 1 giới thiệu tổng quan về xử lý ảnh. Chương 2 giới thiệu khái niệm về xương và một số phương pháp tìm xương. Chương 3 trình bày về phương pháp DSE (Discrete Skeleton Evolution). Chương 4 giới thiệu môi trường cài đặt và kết quả thu được.

## DANH MỤC HÌNH VẼ

Hình 1.1. Hình minh họa các phép toán trên ảnh nhị phân.....	8
Hình 1.2. Hiệu quả của thao tác nhị phân đơn giản trên một ảnh nhỏ.....	9
Hình 1.3. A dẫn bởi B .....	10
Hình 1.4. Dẫn mất điểm ảnh .....	11
Hình 1.5. Dẫn ảnh sử dụng phần tử cấu trúc.....	11
Hình 1.6. Phép co nhị phân. ....	12
Hình 1.7. Sử dụng phép toán mở .....	13
Hình 1.8. Phép đóng.....	14
Hình 1.9. Phép đóng với độ sâu lớn.....	14
Hình 1.10. Các giai đoạn chính trong xử lý ảnh .....	15
Hình 2.1. Trục trung vị.....	20
Hình 2.2. Xương Voronoi rời rạc ảnh hưởng của các hàm hiệu chỉnh khác nhau .....	22
Hình 2.3. Minh họa thuật toán trộn hai sơ đồ Voronoi.....	23
Hình 2.4. Minh họa thuật toán thêm một điểm biên vào sơ đồ Voronoi .....	24
Hình 3.1. Minh họa hạn chế 1 .....	28
Hình 3.2. Minh họa hạn chế 2 .....	28
Hình 3.3. Minh họa hạn chế 3.....	29
Hình 3.4. Cắt tỉa xương với phân chia đường biên.....	30
Hình 3.5. Trình tự xương của lá.....	31
Hình 3.6. Minh họa cắt tỉa xương với DCE .....	33
Hình 3.7. Các đỉnh lồi như nhau có thể sinh ra các nhánh xương khác nhau với mức quan trọng khác nhau.....	35
Hình 3.8. Loại bỏ đỉnh lồi không quan trọng tạo ra hình ảnh xương tối ưu.....	35
Hình 3.9. Quá trình tiến hóa bộ xương thu được trong vòng lặp cắt tỉa xương của một con chim.....	36
Hình 3.10. Các điểm xương cuối và các điểm giao nhau. ....	38
Hình 3.11. Khôi phục lại hình dạng gốc từ xương.....	38
Hình 4.1. Giao diện chương trình .....	41
Hình 4.2. Xương thu được bằng phương pháp DCE và DSE.....	42
Hình 4.3. Xương thu được bằng phương pháp DSE theo các ngưỡng khác nhau, t là giá trị ngưỡng.....	44

## CHƯƠNG 1: TỔNG QUAN VỀ XỬ LÝ ẢNH

### 1.1 Các khái niệm cơ bản trong xử lý ảnh

#### 1.1.1 Xử lý ảnh là gì

Xử lý ảnh số bao gồm các phương pháp và kỹ thuật biến đổi, để truyền tải hoặc mã hóa các ảnh tự nhiên.

Xử lý ảnh là lĩnh vực nghiên cứu, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới tuân thủ tính chất và đặc trưng riêng của xử lý.

Có 2 mục đích chính của xử lý ảnh:

- Cải thiện chất lượng phục vụ cho quan sát.
- Chuẩn bị các điều kiện cho việc trích chọn các đặc trưng phục vụ cho việc nhận dạng và ra quyết định.

#### 1.1.2 Ảnh và điểm ảnh

Ảnh tự nhiên là ảnh liên tục về không gian và độ sáng. Để xử lý bằng máy tính (số), ảnh cần phải được số hóa. Số hóa là sự biến đổi gần đúng một ảnh liên tục thành một tập điểm phù hợp với ảnh thật về trí (không gian) và độ sáng (mức xám). Khoảng cách giữa các điểm ảnh được thiết lập sao cho mắt người không phân biệt được ranh giới giữa chúng. Mỗi một điểm như vậy gọi là điểm ảnh (PEL: Picture Element) hay gọi tắt là Pixel. Trong khuôn khổ ảnh hai chiều, mỗi pixel tương ứng với cặp tọa độ  $(x, y)$ .

Điểm ảnh (pixel) là một phần tử của ảnh số tại tọa độ  $(x, y)$  với độ xám hoặc màu nhất định. Kích thước và khoảng cách giữa các điểm ảnh được chọn thích hợp sao cho mắt người cảm nhận được sự liên tục về không gian và mức xám của ảnh số gần như ảnh thật. Mỗi phần tử trong ma trận được gọi là phần tử ảnh.

Ảnh được xem như tập hợp các điểm ảnh.

#### 1.1.3 Quan hệ giữa các điểm ảnh

##### 1.1.3.1 Các lân cận của điểm ảnh

Giả sử một ảnh số được biểu diễn bằng hàm  $f(x, y)$ ,  $p$  và  $q$  là cặp điểm ảnh có quan hệ với nhau, điểm ảnh  $p$  có tọa độ  $(x, y)$ . Định nghĩa các lân cận của điểm ảnh.

Lân cận 4 của  $p$  kí hiệu  $N_4(p)$ :

$$- N_4(p) = \{(x-1, y); (x, y-1); (x, y+1); (x+1, y)\}$$

Lân cận chéo của  $p$  kí hiệu  $N_p(p)$ :

$$- N_p(p) = \{(x+1, y+1); (x+1, y-1); (x-1, y+1); (x-1, y-1)\}$$

Lân cận 8 của  $p$  kí hiệu  $N_8(p)$ :

$$- N_8(p) = N_4(p) + N_p(p)$$

### 1.1.3.2 Các mối liên kết điểm ảnh

Các mối liên kết được sử dụng để xác định giới hạn của đối tượng hoặc xác định vùng trong một ảnh. Một liên kết được đặc trưng bởi tính liên hệ giữa các điểm và mức xám của chúng. Có ba loại liên kết:

- Liên kết 4: Hai điểm ảnh  $p$  và  $q$  được gọi là liên kết 4 nếu  $q$  thuộc  $N_4(p)$
- Liên kết 8: Hai điểm ảnh  $p$  và  $q$  được gọi là liên kết 8 nếu  $q$  thuộc  $N_8(p)$
- Liên kết  $m$  (liên kết hỗn hợp): Hai điểm ảnh  $p$  và  $q$  được gọi là liên kết hỗn hợp nếu  $q$  thuộc  $N_4(p)$  hoặc  $q$  thuộc  $N_8(p)$

### 1.1.3.3 Đo khoảng cách giữa các điểm ảnh

Khoảng cách  $D(p, q)$  giữa hai điểm ảnh  $p$  có tọa độ  $(x, y)$ ,  $q$  có tọa độ  $(s, t)$  là hàm khoảng cách (Distance) nếu:

- $D(p, q) \geq 0$  (Với  $D(p, q)=0$  khi và chỉ khi  $p=q$ )
- $D(p, q) = D(q, p)$
- $D(p, z) \leq D(p, q) + D(q, z)$ ;  $z$  là một điểm ảnh khác

Khoảng cách Euclide giữa hai điểm ảnh  $p(x, y)$  và  $q(s, t)$  được định nghĩa như sau:  $D_e(p, q) = [(x - s)^2 + (y - t)^2]^{1/2}$

### 1.1.4 Mức xám của ảnh

Mức xám của điểm ảnh là cường độ sáng của nó được gán bằng giá trị số tại điểm đó.

Các thang giá trị mức xám thông thường là: 16, 32, 64, 128, 256 (Mức 256 là mức phổ dụng nhất vì máy tính dùng 1 byte (8 bit) để biểu diễn mức xám. Mức xám dùng 1 byte biểu diễn:  $2^8=256$ , tức là từ 0 đến 255)

Ảnh đen trắng là ảnh có hai màu đen và trắng. Nếu phân mức đen trắng thành L mức, sử dụng số bit B để mã hóa mức đen trắng (hay mức xám) thì L được xác định:  $L=2^B$ .

- Nếu  $L=2$ ,  $B=1$  nghĩa là chỉ có 2 mức 0 và 1. Ảnh dùng hai mức 0 và 1 để biểu diễn mức xám gọi là ảnh nhị phân. Mức 1 ứng với màu sáng còn mức 0 ứng với màu tối.
- Nếu L lớn hơn 2 đó là ảnh đa cấp xám. Như vậy ảnh nhị phân mỗi điểm ảnh được mã hóa trên 1 bit, còn ảnh 256 mức mỗi điểm ảnh được mã hóa trên 8 bit. Ảnh đen trắng nếu dùng 8 bit (1 byte) để biểu diễn mức xám số mỗi mức xám được biểu diễn dưới dạng một số nguyên nằm trong khoảng từ 0 đến 255, mức 0 biểu diễn cho cường độ đen nhất và mức 255 biểu diễn cho cường độ sáng nhất.

Ảnh màu: là ảnh tổ hợp từ 3 màu cơ bản đỏ (Red), lục (Green), lam (Blue). Để biểu diễn cho một điểm ảnh màu dùng 3 byte để mô tả 24 bit màu  $2^{8*3}=2^{24} \approx 16,7$  triệu màu.

### 1.1.5 Độ phân giải

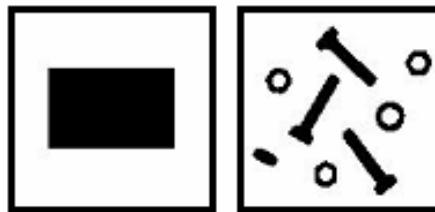
Độ phân giải (Resolution) của ảnh là mật độ điểm ảnh được ấn định trên ảnh số khi hiển thị. Như vậy khoảng cách giữa các điểm ảnh được chọn sao cho mắt người vẫn thấy được sự liên tục của ảnh. Việc lựa chọn khoảng cách thích hợp tạo nên một mật độ phân bố, đó chính là độ phân giải và được phân bố theo trục x và y trong không gian hai chiều.

## 1.2 Các phép toán cơ bản trên ảnh nhị phân

### 1.2.1 Các phép toán logic

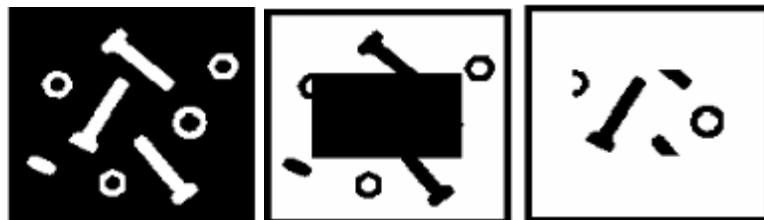
P	Q	P AND Q	P OR Q	NOT P
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Hình 1.1 dưới đây minh họa các phép toán với giá trị nhị phân “1” có màu đen, còn giá trị nhị phân “0” có màu trắng.



(a)Ảnh a

(b)Ảnh b



(c)

(d)

(e)

Hình 1.1. Hình minh họa các phép toán trên ảnh nhị phân

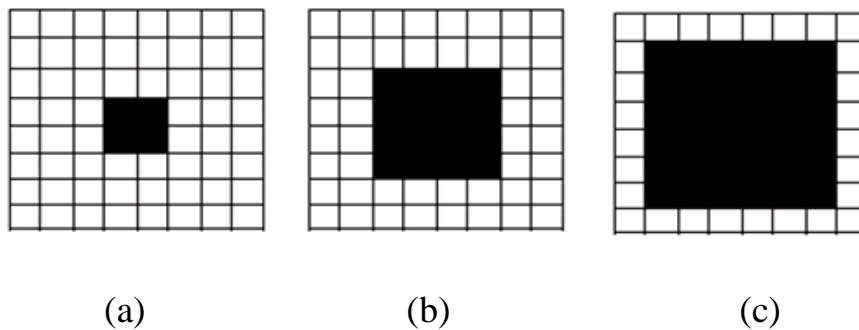
Trong hình 1.1: hình (a) và (b) là ảnh ban đầu; (c) phép NOT (b); (d) phép OR (a,b); (e) phép AND (a,b).

### 1.2.2 Các phép toán hình thái học

Hình thái (morphology) có nghĩa là “hình thức và cấu trúc của một đối tượng”, hoặc là cách sắp xếp mối quan hệ bên trong giữa các phần của đối tượng. Hình thái có liên quan đến hình dạng, và hình thái số là một cách để mô tả hoặc phân tích hình dạng của một đối tượng số.



Những thao tác hình thái nhị phân được xây dựng trên ảnh chỉ có 2 mức xám 0 và 1, “0” ứng với màu trắng, “1” ứng với màu đen. Trước hết, để bắt đầu, ta hãy xem hình 1.2a. Tập hợp các điểm ảnh đen tạo nên đối tượng ảnh hình vuông và trong hình 1.2b, đối tượng ảnh cũng là hình vuông nhưng là hình vuông lớn hơn so với hình 1.2a một điểm ảnh về mọi phía, nghĩa là thay mọi lân cận trắng của các điểm ảnh trong hình 1.2a thành các điểm ảnh đen. Đối tượng trong hình 1.2c cũng được thao tác tương tự, tức là hình 1.2b được tăng thêm một 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. Do vậy, đối tượng ảnh trong hình 1.2a 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 1.2. Hiệu quả của thao tác nhị phân đơn giản trên một ảnh nhỏ

Trong hình 1.2: hình (a) ảnh ban đầu; (b) ảnh dẫn 1 điểm ảnh; (c) ảnh dẫn 2 điểm ảnh so với ảnh ban đầu.

### 1.2.2.1 Phép dẫn nhị phân

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

$$(A)x = \{c \mid c = a + x, a \in A\} \quad (1.1)$$

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.

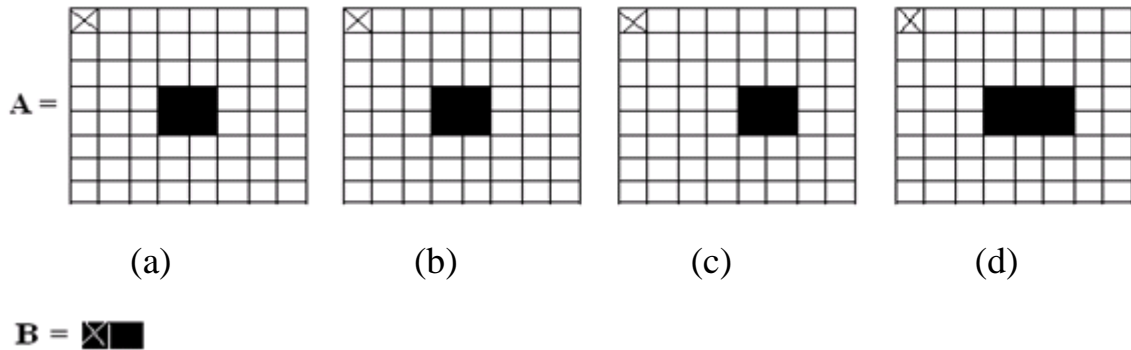
Bây giờ ta có thể định nghĩa phép dẫn (dilation) qua lý thuyết tập hợp như sau: Phép dẫn tập A bởi tập B, đó là tập:

$$A \oplus B = \{c \mid c = a + b, a \in A, b \in B\} \quad (1.2)$$

Để 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 trong hình 1.2a và  $B = \{(0,0), (0,1)\}$ .

Những phần tử trong tập  $C = A \oplus B$  được tính dựa trên công thức (1.1), có thể viết lại như sau:

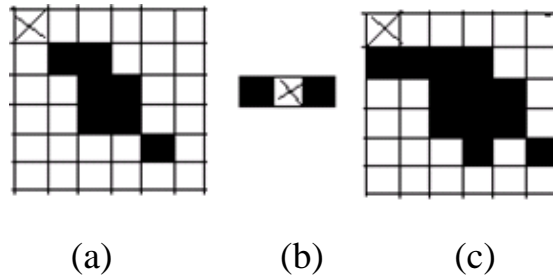
$$A \oplus B = (A + \{(0,0)\}) \cup (A + \{(0,1)\}) \quad (1.3)$$



Hình 1.3. A dẫn bởi B

Trong hình 1.3: (a) tập A ban đầu; (b) tập A cộng phần tử  $(0,0)$ ; (c) tập A cộng phần tử  $(0,1)$ ; (d) hợp của (b) và (c) (kết quả của phép dẫn).

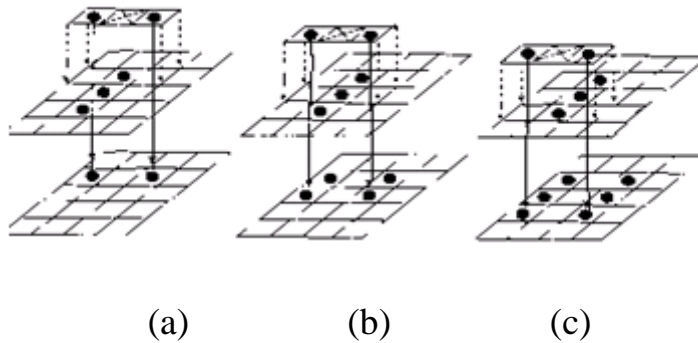
Nhận thấy rằng trong hình 1.4, có một số phần tử của đối tượng ban đầu sẽ không có.



Hình 1.4. Dẫn mắt điểm ảnh

Trong hình 1.4: (a) ảnh A1; (b) phần tử cấu trúc B1; (c) A1 được dẫn bởi B1.

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. Điều này được thể hiện khá rõ trong hình 1.5.



Hình 1.5. Dẫn ảnh sử dụng phần tử cấu trúc

Trong hình 1.5: (a) là 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.

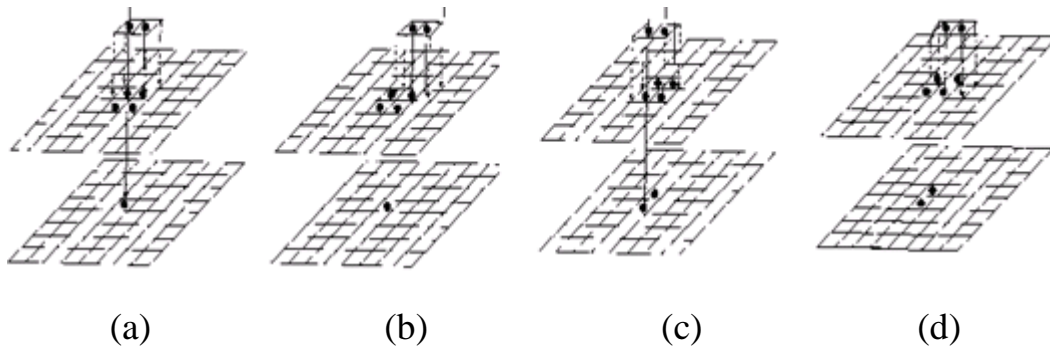
### 1.2.2.2 Phép co nhị phân

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. Trong trường hợp đơn giản nhất, một phép co nhị phân sẽ tách lớp điểm ảnh bao quanh đối tượng ảnh, chẳng hạn hình 1.2b là kết quả của phép co được áp dụng đối với hình 1.2c.

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

$$A \ominus B = \{c \mid (B)c \subseteq A\} \quad (1.4)$$

Đầu tiên, ta hãy xét một ví dụ đơn giản sau đây:



Hình 1.6. Phép co nhị phân.

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.

Phần tử cấu trúc dịch chuyển tới điểm ảnh tiếp theo trong ảnh, và có một điểm không phù hợp và kết quả là điểm trắng.

Ở lần dịch chuyển tiếp theo, các thành viên của cấu trúc lại phù hợp nên kết quả là điểm đen.

Tương tự được kết quả cuối cùng là điểm trắng.

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:

$$(B \ominus A)c = Bc \oplus \hat{A} \quad (1.5)$$

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  $\hat{A} = A$

Khi đó:

$$(B \ominus A)_c = B_c \oplus A \quad (1.6)$$

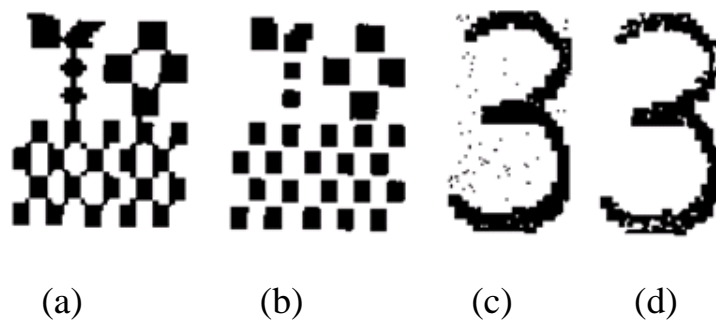
Hay, phần bù của phép co A bởi B được coi như phép dẫn nền của ảnh A (ta quy ước trong ảnh nhị phân rằng: đối tượng ảnh là những điểm đen quan sát, ảnh A là bao gồm cả điểm đen và nền).

### 1.2.2.3 Phép mở (Opening)

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 (dãn) và E là Erosion (co).

$$\text{Opening (I)} = D(E(I)) \quad (1.7)$$

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 1.7 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.



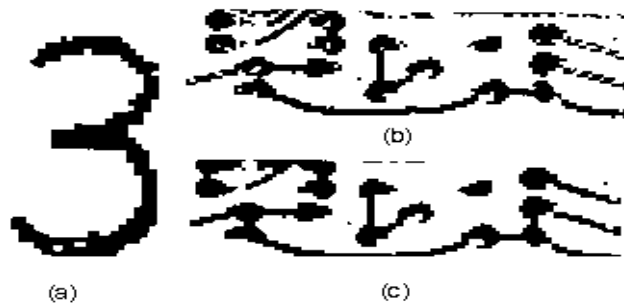
Hình 1.7. Sử dụng phép toán mở

Trong hình 1.7: (a) một ảnh có nhiều vật thể được liên kết; (b) các vật thể được cách ly bởi phép mở với cấu trúc đơn giản; (c) một ảnh có nhiễu; (d) ảnh nhiễu sau khi sử dụng phép mở, các điểm nhiễu.

### 1.2.2.4 Phép đóng (Closing)

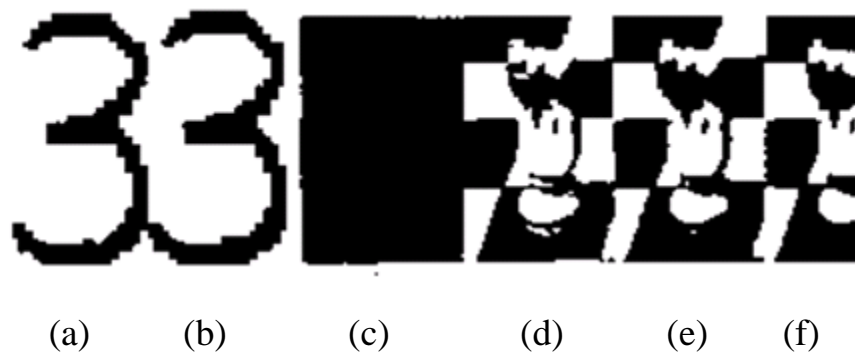
Tương tự phép mở ảnh nhưng trong phép đóng ảnh, thao tác dẫn ảnh được thực hiện trước, sau đó mới đến thao tác co ảnh và cùng làm việc trên cùng một phần tử cấu trúc.

$$\text{Close (I)} = E(D(I)) \quad (1.8)$$



Hình 1.8. Phép đóng

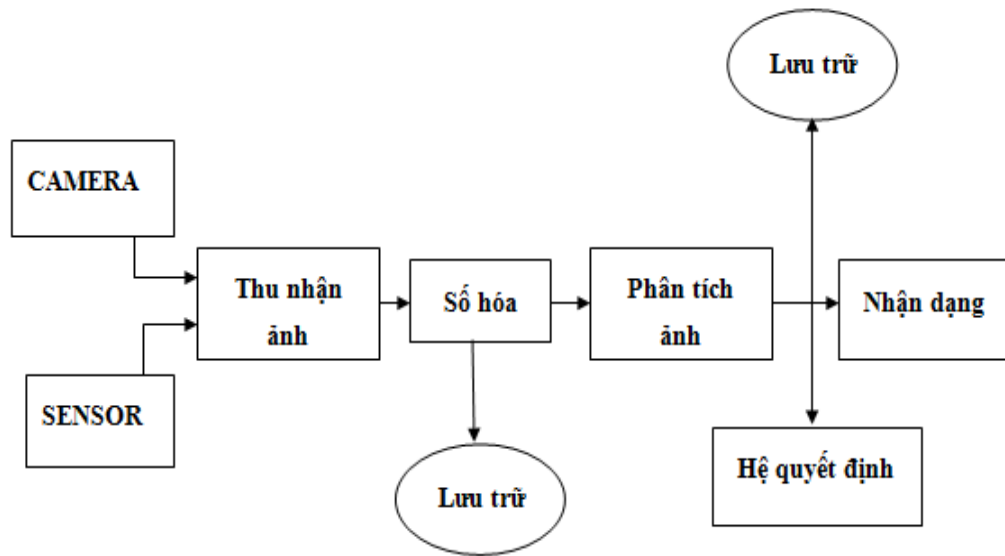
Trong hình 1.8: (a) kết quả đóng sử dụng cấu trúc đơn giản; (b) ảnh của một bảng mạch được phân ngưỡng và có các vết đứt; (c) ảnh tương tự sau khi đóng nhưng những nét đứt đã được nối liền.



Hình 1.9. Phép đóng với độ sâu lớn

Trong hình 1.9: (a) từ hình 1.8a, sử dụng phép đóng với độ sâu 2; (b) 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; (f) Sau khi thực hiện phép đóng với độ sâu 2.

### 1.3 Các giai đoạn chính trong xử lý ảnh



Hình 1.10. Các giai đoạn chính trong xử lý ảnh

Trước hết là quá trình thu nhận ảnh. Ảnh thu nhận qua camera. Thường ảnh thu nhận qua camera là tín hiệu tương tự (loại camera ống kính CCIR), nhưng cũng có thể là loại tín hiệu số hóa (loại CCD- Charge Coupled Device).

Ảnh cũng có thể thu nhận từ vệ tinh qua các bộ cảm ứng (sensor), hay ảnh, tranh được quét trên scanner. Tiếp theo là quá trình số hóa (Digitalizer) để biến đổi tín hiệu tương tự sang tín hiệu rời rạc (lấy mẫu) và số hóa bằng lượng hóa, trước khi chuyển sang giai đoạn xử lý, phân tích hay lưu trữ lại.

Quá trình phân tích ảnh thực chất bao gồm nhiều công đoạn nhỏ. Trước hết là công việc tăng cường ảnh (Image Enhancement) để nâng cao chất lượng ảnh. Do những nguyên nhân khác nhau: có thể do chất lượng thiết bị thu nhận ảnh, do nguồn sáng hay do nhiễu, ảnh có thể bị suy biến. Do vậy cần phải tăng cường và khôi phục (Image Restoration) lại ảnh để làm nổi bật một số đặc tính chính của ảnh, hay làm cho ảnh gần giống nhất với trạng thái gốc – trạng thái trước khi ảnh bị biến dạng. Giai đoạn tiếp theo là phát hiện các đặc tính như biên (Edge Detection), phân vùng ảnh (Image Segmentation), trích chọn các đặc tính (Feature Extraction), v. v. . .

Cuối cùng, tùy theo mục đích của ứng dụng, sẽ là giai đoạn nhận dạng, phân lớp hay các quyết định khác.

#### **1.4 Một số ứng dụng cơ bản**

Kỹ thuật xử lý ảnh trước đây chủ yếu được sử dụng để nâng cao chất lượng hình ảnh, chính xác hơn là tạo cảm giác về sự gia tăng chất lượng ảnh quang học trong mắt người quan sát. Thời gian gần đây, phạm vi ứng dụng xử lý ảnh mở rộng không ngừng, có thể nói hiện không có lĩnh vực khoa học nào không sử dụng các thành tựu của công nghệ xử lý ảnh số.

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. Hình ảnh các cơ quan chức năng của con người sau đó có thể được xử lý tiếp để nâng cao độ tương phản, lọc, tách các thành phần cần thiết (chụp cắt lớp) hoặc tạo ra hình ảnh trong không gian ba chiều (siêu âm 3 chiều).

Trong lĩnh vực địa chất, hình ảnh nhận được từ vệ tinh có thể được phân tích để xác định cấu trúc bề mặt trái đất. Kỹ thuật làm nổi đường biên (image enhancement) và khôi phục hình ảnh (image restoration) cho phép nâng cao chất lượng ảnh vệ tinh và tạo ra các bản đồ địa hình 3-D với độ chính xác cao.

Trong ngành khí tượng học, ảnh nhận được từ hệ thống vệ tinh theo dõi thời tiết cũng được xử lý, nâng cao chất lượng và ghép hình để tạo ra ảnh bề mặt trái đất trên một vùng rộng lớn, qua đó có thể thực hiện việc dự báo thời tiết một cách chính xác hơn. Dựa trên các kết quả phân tích ảnh vệ tinh tại các khu vực đông dân cư còn có thể dự đoán quá trình tăng trưởng dân số, tốc độ ô nhiễm môi trường cũng như các yếu tố ảnh hưởng tới môi trường sinh thái.

Xử lý ảnh được sử dụng nhiều trong các hệ thống quản lý chất lượng và số lượng hàng hóa trong các dây chuyền tự động, ví dụ như hệ thống phân tích ảnh để phát hiện bọt khí bên vật thể đúc bằng nhựa, phát hiện các linh kiện không đạt tiêu chuẩn (biến dạng) trong quá trình sản xuất hoặc hệ thống đếm sản phẩm thông qua hình ảnh nhận được từ camera quan sát.



Xử lý ảnh còn được sử dụng rộng rãi trong lĩnh vực hình sự và các hệ thống bảo mật hoặc kiểm soát truy cập: quá trình xử lý ảnh với mục đích nhận dạng vân tay hay khuôn mặt cho phép phát hiện nhanh các đối tượng nghi vấn cũng như nâng cao hiệu quả hệ thống bảo mật cá nhân cũng như kiểm soát ra vào. Ngoài ra, có thể kể đến các ứng dụng quan trọng khác của kỹ thuật xử lý ảnh tĩnh cũng như ảnh động trong đời sống như tự động nhận dạng, nhận dạng mục tiêu quân sự, máy nhìn công nghiệp trong các hệ thống điều khiển tự động, nén ảnh tĩnh, ảnh động để lưu và truyền trong mạng viễn thông v. v...

## CHƯƠNG 2: XƯƠNG VÀ CÁC KỸ THUẬT TÌM XƯƠNG

### 2.1 Khái niệm xương

Xương được coi như hình dạng cơ bản của một đối tượng, với số ít điểm các điểm ảnh cơ bản. Ta có thể lấy được thông tin về hình dạng nguyên bản của một đối tượng thông qua xương.

### 2.2 Các hướng tiếp cận trong việc tìm xương

Các kỹ thuật tìm xương luôn là chủ đề nghiên cứu trong xử lý ảnh. Do đó tính phức tạp của nó, mặc dù có những nỗ lực cho việc phát triển các thuật toán tìm xương nhưng các phương pháp đưa ra đều bị mất mát thông tin. Có thể chia thuật toán tìm xương thành hai loại cơ bản:

- Các thuật toán tìm xương dựa trên làm mảnh
- Các thuật toán tìm xương không dựa trên làm mảnh

#### 2.2.1 Tìm xương dựa trên làm mảnh

##### 2.2.1.1 Sơ lược về thuật toán làm mảnh

Thuật toán làm mảnh ảnh số nhị phân là một trong các thuật toán quan trọng trong xử lý ảnh và nhận dạng. Xương chứa những thông tin bất biến về cấu trúc của ảnh, giúp cho quá trình nhận dạng hoặc vecto hóa sau này.

Thuật toán làm mảnh là quá trình lặp duyệt và kiểm tra tất cả các điểm thuộc đối tượng. Trong mỗi lần lặp tất cả các điểm đối tượng sẽ được kiểm tra: tùy thuộc vào mỗi thuật toán, nếu như chúng thỏa mãn điều kiện xóa nào đó thì nó sẽ bị xóa đi. Quá trình cứ lặp lại cho đến khi không còn điểm biên nào được xóa. Đối tượng được bóc dần lớp biên cho đến khi nào bị thu mảnh lại chỉ còn các điểm biên.

##### 2.2.1.2 Tìm xương dựa trên làm mảnh

Thuật toán làm mảnh song song là thuật toán mà trong đó các điểm được xử lý theo phương pháp song song, tức là được xử lý cùng một lúc. Giá trị của mỗi điểm sau một lần lặp chỉ phụ thuộc vào giá trị của các láng giềng bên cạnh (thường là 8 – láng giềng) mà giá trị của các điểm này đã được xác định trong một lần lặp trước đó. Trong

máy có nhiều bộ vi xử lý mỗi vi xử lý sẽ xử lý một vùng của đối tượng, nó có quyền đọc từ các điểm ở vùng khác nhau nhưng chỉ được ghi trên vùng của nó xử lý.

Trong thuật toán làm mảnh tuần tự các điểm thuộc đối tượng sẽ được kiểm tra theo một thứ tự nào đó (chẳng hạn các điểm được xét từ trái qua phải, từ trên xuống dưới). Giá trị của điểm sau mỗi lần lặp không những phụ thuộc vào giá trị của các láng giềng bên cạnh mà còn phụ thuộc vào các điểm đã được xét trước đó trong chính lần lặp đang xét.

Chất lượng của thuật toán làm mảnh được đánh giá theo các tiêu chuẩn được liệt kê dưới đây nhưng không nhất thiết phải thỏa mãn đồng thời tất cả các tiêu chuẩn:

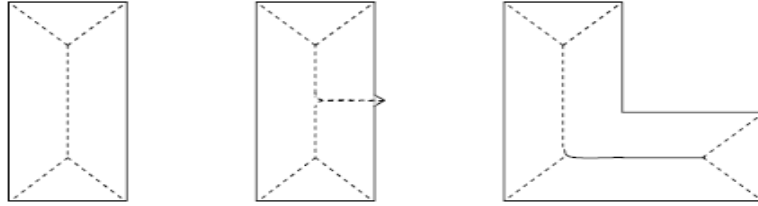
- Bảo toàn tính liên thông của đối tượng và phần bù của đối tượng
- Sự tương hợp giữa xương và cấu trúc của ảnh đối tượng
- Bảo toàn các thành phần liên thông
- Bảo toàn các điểm cực
- Xương chỉ gồm các điểm biên, càng mảnh càng tốt
- Bền vững đối với nhiễu
- Xương cho phép khôi phục ảnh ban đầu của đối tượng
- Xương thu được ở chính giữa đường nét của đối tượng được làm mảnh
- Xương nhận được bất biến với phép quay

### 2.2.2 Tìm xương không dựa trên làm mảnh

Để tách được xương của đối tượng có thể sử dụng đường biên của đối tượng. Với bất cứ một điểm  $p$  nào đó trên đối tượng, đều có thể bao nó bởi một đường biên. Nếu như có nhiều hơn một điểm biên có khoảng cách ngắn nhất thì  $p$  nằm trên trục trung vị. Tất cả các điểm như vậy lập thành trục trung vị hay xương của đối tượng. Việc xác định xương được tiến hành thông qua hai bước:

- Bước thứ nhất, tính khoảng cách từ mỗi điểm ảnh của đối tượng đến điểm biên gần nhất. Như vậy cần phải tính toán khoảng cách tới tất cả các điểm biên của ảnh.

- Bước thứ hai, khoảng cách của ảnh đã được tính toán và các điểm ảnh có giá trị lớn nhất được xem là nằm trên xương của đối tượng.



Hình 2.1. Trục trung vị

Hầu hết các nhà nghiên cứu đều cho rằng thay đổi trục trung vị thường không mang lại một xương chuẩn, và thời gian tính toán quá dài, tuy nhiên nó là mẫu cơ bản của phần lớn các phương pháp làm mảnh.

Phương pháp thay đổi trục trung vị được coi là một phương pháp làm mảnh không lặp, ngoài ra còn có một vài thuật toán duyệt các điểm biên 2 bên mẫu, tính điểm trung tâm các đường nối giữa các điểm biên đó và xương thu được là tập hợp các điểm trung tâm đó (line following) hoặc các phương thức sử dụng chuỗi Fourier (Fourier transform) cũng được coi là làm mảnh không lặp.

### 2.2.2.1 Khái quát về lược đồ Voronoi

Lược đồ Voronoi là một công cụ hiệu quả trong hình học tính toán. Cho hai điểm  $P_i, P_j$  là 2 phần tử của tập  $\Omega$  gồm  $n$  điểm trong mặt phẳng. Tập các điểm trong mặt phẳng gần hơn là nửa mặt phẳng  $H(P_i, P_j)$  chứa điểm và bị giới hạn bởi đường trung trực của đoạn thẳng. Do đó, tập các điểm gần hơn bất kỳ điểm  $P_j$  nào có thể thu được bằng cách giao  $n-1$  các nửa mặt phẳng  $H(P_i, P_j)$ :

$$V(P_i) = \bigcap_{j \neq i} H(P_i, P_j) \quad (i=1, \dots, n) \quad (2.1)$$

**Định nghĩa 2.1** [Đa giác/ Sơ đồ Voronoi]

Sơ đồ Voronoi của  $\Omega$  là tập hợp tất cả các  $V(P_i)$

$$\text{Vor}(\Omega) = \{V(P_i) \mid P_i \in \Omega \text{ (là một đa giác)}\} \quad (2.2)$$

**Định nghĩa 2.2** [Đa giác Voronoi tổng quát]

Cho tập các điểm  $\Omega$ , đa giác Voronoi của tập con  $U$  của  $\Omega$  được định nghĩa như sau:

$$V(U) = \{P \mid \exists v \in U, \forall w \in \Omega \setminus U: d(P, w) = d(P, v)\} \quad (2.3)$$

**2.2.2.2 Trục trung vị Voronoi rời rạc****Định nghĩa 2.3** [Bản đồ khoảng cách – Distance Map]

Cho đối tượng  $S$ , đối với mỗi  $(x, y) \in S$ , ta tính giá trị khoảng cách  $\text{map}(x, y)$  với hàm khoảng cách  $d(\cdot, \cdot)$  như sau:

$$\forall (x, y) \in S: \text{map}(x, y) = \min d[(x, y), (x_i, y_i)] \quad (2.4)$$

Trong đó  $(x_i, y_i) \in B(S)$  – tập các điểm biên của  $S$

Tập tất cả các  $\text{map}(x, y)$ , kí hiệu là  $DM(S)$ , được gọi bản đồ khoảng cách của  $S$ .

Chú ý: Nếu hàm khoảng cách  $d(\cdot, \cdot)$  là khoảng cách Euclide, thì phương trình (2.4) chính là khoảng cách ngắn nhất từ một điểm bên trong đối tượng tới biên. Do đó, bản đồ khoảng cách được gọi là bản đồ khoảng cách Euclide EDM( $S$ ) của  $S$ . Định nghĩa trên được dùng cho cả hình rời rạc lẫn liên tục.

**Định nghĩa 2.4** [Tập các điểm biên sinh]

Cho  $\text{map}(x, y)$  là khoảng cách ngắn nhất từ  $(x, y)$  đến biên (theo định nghĩa 2.3). Ta định nghĩa:  $\text{map}^{-1}(x, y) = \{p \mid p \in B(S), d(p, (x, y)) = \text{map}(x, y)\}$

Khi đó tập các điểm biên sinh  $\hat{B}(S)$  được định nghĩa bởi:

$$\hat{B}(S) = \cup \text{map}^{-1}(x, y), (x, y) \in S \quad (2.5)$$

Do  $S$  có thể chứa các đường biên rời nhau, nên  $\hat{B}(S)$  bao gồm nhiều tập con, mỗi tập mô tả một đường biên phân biệt:

$$\hat{B}(S) = \{B_1(S), \dots, B_{1n}(S)\} \quad (2.6)$$

**Định nghĩa 2.5** [Trục trung vị Voronoi rời rạc (DVMA)]

Trục trung vị Voronoi rời rạc được định nghĩa là kết quả của sơ đồ Voronoi bậc nhất rời rạc của tập các điểm biên sinh giao với hình sinh S:

$$DVMA(\hat{B}(S)) = \text{Vor}(\hat{B}(S)) \cap S \quad (2.7)$$

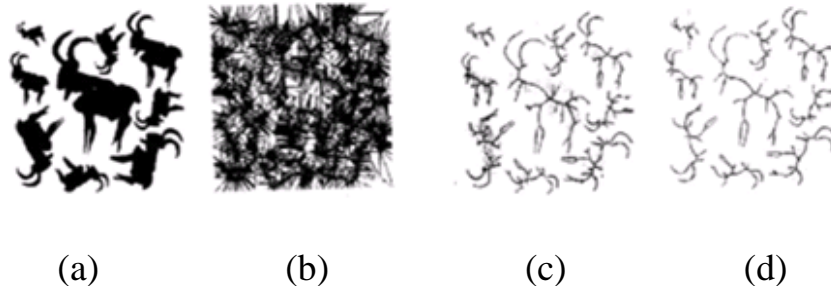
**2.2.2.3 Xương Voronoi rời rạc****Định nghĩa 2.6** [Xương Voronoi rời rạc – Discrete Voronoi Skeleton]

Xương Voronoi rời rạc theo ngưỡng T, kí hiệu là SkeDVMA( $\hat{B}(S), T$ ) (hoặc Ske( $\hat{B}(S), T$ )) là một tập con của trục trung vị Voronoi:

$$\text{SkeDVMA}(\hat{B}(S), T) = \{ (x, y) \mid (x, y) \in DVMA(\hat{B}(S)), \Psi(x, y) > T \} \quad (2.8)$$

$\Psi$ : là hàm hiệu chỉnh

Để thấy ngưỡng T càng lớn thì số lượng điểm tham gia trong xương Voronoi càng ít (Hình 2.2).



Hình 2.2. Xương Voronoi rời rạc ảnh hưởng của các hàm hiệu chỉnh khác nhau

Trong hình 2.2: (a) ảnh nhị phân; (b) Sơ đồ Voronoi; (c) hiệu chỉnh bởi hàm Potential,  $T=9.0$ ; (d) hiệu chỉnh bởi hàm Potential,  $T=18.0$

**2.2.2.4 Thuật toán tìm xương**

Thuật toán tìm xương dựa trên một số ý tưởng sau:

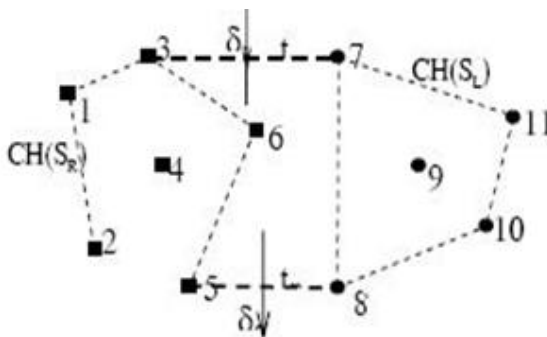
Tăng trưởng: Việc tính toán sơ đồ Voronoi được bắt đầu từ một điểm sinh trong mặt phẳng. Sau đó điểm sinh thứ hai được thêm vào và quá trình tính toán tiếp tục với đa giác Voronoi đã tìm được với điểm vừa được thêm vào đó. Cứ như thế, quá trình tính toán sơ đồ Voronoi được thực hiện cho đến khi không còn điểm sinh nào được

thêm vào. Nhược điểm của chiến lược này là mỗi khi điểm mới được thêm vào, nó có thể gây ra sự phân vùng toàn bộ các đa giác Voronoi đã được tính.

Chia để trị: Tập các điểm biên đầu tiên được chia thành hai tập điểm có kích cỡ bằng nhau. Sau đó thuật toán tính toán sơ đồ Voronoi cho cả hai tập con điểm biên đó. Cuối cùng, người ta thực hiện việc ghép cả hai sơ đồ Voronoi trên để thu hút được kết quả mong muốn. Tuy nhiên, việc chia tập các điểm biên thành hai phần không phải được thực hiện một lần, mà được lặp lại nhiều lần cho đến khi việc tính toán sơ đồ Voronoi trở nên đơn giản. Vì thế, việc tính sơ đồ Voronoi trở thành vấn đề làm thế nào để trộn hai sơ đồ Voronoi lại với nhau.

Thuật toán sẽ trình bày ở đây là sự kết hợp của 2 ý tưởng trên. Tuy nhiên, nó sẽ mang lại nhiều đáng đáp của thuật toán chia để trị.

Hình 2.3 minh họa ý tưởng của thuật toán này. Mười một điểm biên được chia thành 2 phần (bên trái: 1-6, bên phải: 7-11) bởi đường gấp khúc  $\delta$ , và hai sơ đồ Voronoi tương ứng  $Vor(S_L)$  và  $Vor(S_R)$ . Để thu được sơ đồ Voronoi  $Vor(S_L \cup S_R)$ , ta thực hiện việc trộn hai sơ đồ trên và xác định lại một số đa giác sẽ bị sửa đổi do ảnh hưởng của các điểm bên cạnh thuộc sơ đồ kia. Mỗi phần tử của  $\delta$  sẽ là một bộ phận của đường trung trực nối hai điểm mà một điểm thuộc  $Vor(S_L)$  và một thuộc  $Vor(S_R)$ . Trước khi xây dựng  $\delta$ , ta tìm ra phần tử đầu và cuối của nó. Nhìn vào hình 2.3, ta nhận thấy rằng cạnh  $\delta_1$  và  $\delta_5$  là các tia. Để nhận thấy rằng việc tìm các cạnh đầu và cuối của  $\delta$  trở thành việc tìm cạnh vào  $t_\alpha$  và cạnh ra  $t_\omega$

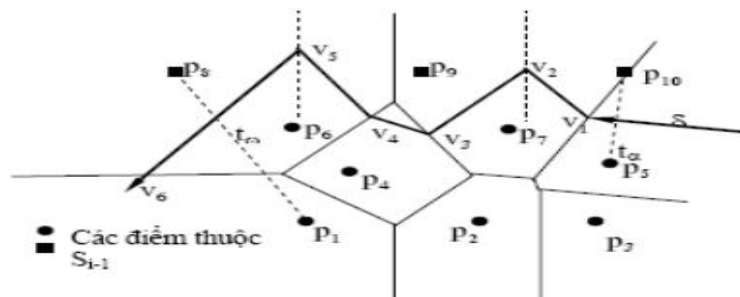


Hình 2.3. Minh họa thuật toán trộn hai sơ đồ Voronoi

Sau khi tìm được  $t_\alpha$  và  $t_\omega$ , các điểm cuối của  $t_\alpha$  được sử dụng để xây dựng phần tử đầu tiên  $\delta$  ( $\delta_1$  trong hình trên). Sau đó thuật toán tìm điểm giao của  $\delta$  với  $\text{Vor}(S_L)$  và  $\text{Vor}(S_R)$ . Trong ví dụ trên,  $\delta$  đầu tiên giao với  $V(3)$ .

Kể từ đây, các điểm nằm trên phần kéo dài  $\delta$  sẽ gần điểm 6 hơn điểm 3. Do đó, phần tử tiếp theo  $\delta_2$  của  $\delta$  sẽ thuộc vào đường trung trực của điểm 6 và điểm 7. Sau đó điểm giao tiếp theo của  $\delta$  sẽ thuộc vào  $\text{Vor}(S_L)$ ;  $\delta$  sẽ đi vào  $V(9)$  và  $\delta_2$  sẽ được thay thế bởi  $\delta_3$ . Quá trình này sẽ kết thúc khi  $\delta$  gặp phần tử cuối  $\delta_5$ .

Trên đây chỉ là minh họa cho thuật toán trộn hai sơ đồ Voronoi trong chiến lược chia để trị. Tuy nhiên, trong thuật toán sẽ trình bày ở đây thì sự thực hiện có khác một chút. Tập các điểm ảnh không phải được đưa ngay vào từ đầu mà sẽ được quét vào từng dòng một. Giả sử tại bước thứ  $i$ , ta đã thu được một sơ đồ Voronoi gồm  $i-1$  hàng các điểm sinh  $\text{Vor}(S_{i-1})$ . Tiếp theo, ta quét lấy một hàng  $L_i$  các điểm ảnh từ tập các điểm biên còn lại. Thực hiện việc tính toán sơ đồ Voronoi  $\text{Vor}(S_i)$  cho hàng này, sau đó trộn  $\text{Vor}(S_{i-1})$  với  $\text{Vor}(L_i)$ . Kết quả ta sẽ được một sơ đồ mới, và lại thực hiện việc quét hàng  $L_{i+1}$  các điểm sinh còn lại v. v. . Quá trình này sẽ kết thúc khi không còn điểm biên nào để thêm vào sơ đồ Voronoi. Do  $\text{Vor}(L_i)$  sẽ có dạng răng lược (nếu  $L_i$  có  $k$  điểm thì  $\text{Vor}(L_i)$  sẽ gồm  $k-1$  đường thẳng đứng), nên việc trộn  $\text{Vor}(S_{i-1})$  với  $\text{Vor}(L_i)$  có phần đơn giản hơn.



Hình 2.4. Minh họa thuật toán thêm một điểm biên vào sơ đồ Voronoi

Giải thuật trên có thể được mô tả bằng ngôn ngữ tựa Pascal như sau:

Procedure VORONOI

(\* $S_i$ : Tập các điểm của  $i$  dòng quét đầu tiên,  $0 \leq i \leq iMAX$ ,  $\text{Vor}(S_i)$  sơ đồ Voronoi của  $S_i$  \*)



Begin

$i:=0; S_i:=\text{rỗng};$

While ( $i < i_{\max} \wedge S_i \subset \text{straight\_line}$ ) do

Begin

(Khởi tạo sơ đồ Voronoi cho đến khi nó chứa ít nhất một đỉnh)

Increment  $i$ ; GetScanLine  $L_i$ ;

$\text{Vor}(S_i) = \text{VoroPreScan}(\text{Vor}(S_{i-1}), L_i)$ ;

End

While ( $i < i_{\max}$ ) do

Begin

Increment  $i$ ; GetScanLine  $L_i$ ;

$\text{Vor}(L_i) :=$  các đường trung trực sinh bởi các điểm sinh thuộc  $L_i$

$\text{Vor}(S_i) := \text{VoroLink}(\text{Vor}(S_{i-1}), \text{Vor}(L_i))$ ;

End

End.

Giả sử xét trên hệ trục tọa độ thực. Ảnh vào được quét từ dưới lên. Tọa độ  $y$  (biến  $i$ ) tương ứng với từng dòng quét được tăng dần theo từng dòng. Trong thủ tục trên, hàm quan trọng nhất là hàm  $\text{VoroLink}$ , hàm này thực hiện việc trộn sơ đồ Voronoi của  $L_{i-1}$  dòng đã được quét trước đó với sơ đồ Voronoi của dòng hiện tại thứ  $i$ . Trong vòng lặp trên, hàm  $\text{VoroPreScan}$  là một biến cụ thể của hàm  $\text{VoroLink}$ , có nhiệm vụ khởi tạo sơ đồ Voronoi và thoát khỏi vòng lặp ngay khi nó thành lập được sơ đồ Voronoi chứa ít nhất một đỉnh. Hàm  $\text{VoroLink}$  thực hiện việc trộn hai sơ đồ Voronoi  $\text{Vor}(S_{i-1})$  và  $\text{Vor}(L_i)$  với nhau để thành  $\text{Vor}(S_i)$ .

## CHƯƠNG 3: CẮT TỈA XƯƠNG VỚI DSE

### 3.1 Giới thiệu

Xương hay còn gọi là trục trung vị, được sử dụng rộng rãi để phân tích hình dạng và nhận dạng đối tượng như tra cứu ảnh và đồ họa máy tính, nhận dạng kí tự, xử lý ảnh và phân tích các hình ảnh sinh học.

Các phương pháp dựa vào việc phân tích hình dạng dựa trên cây xương hay đồ thị xương chỉ khả năng áp dụng cho các đối tượng đơn giản và hình dạng đặc biệt mà không thể áp dụng cho các hình dạng phức tạp như các hình dạng trong tập dữ liệu MPEG-7. Có hai yếu tố chính hạn chế việc thực hiện của xương dựa vào đối sánh hình dạng:

- Sự nhạy cảm của xương với sự biến đổi đường biên của đối tượng: một ít nhiễu hoặc biến đổi của đường biên thường tạo ra nhánh xương thừa điều này có thể làm biến đổi hình dạng đúng của xương.
- Chi phí thời gian cho việc tìm xương và đối sánh các cây xương hay các đồ thị xương không thể đáp ứng yêu cầu thu hồi hình dạng nhanh.

Phương pháp tìm xương được phân làm 4 loại: phương pháp làm mảnh, thuật toán miền rời rạc dựa trên đồ thị Voronoi, các thuật toán dựa trên biến đổi khoảng cách và phép toán hình thái học.

Tất cả các xương thu được không khuất phục được sự nhạy cảm của xương và nhiều trong số chúng cũng bao gồm phương pháp cắt tỉa cùng với tìm xương. Một phần thiết yếu của thuật toán tìm xương là các thuật toán cắt tỉa xương. Thuật toán này thường xuất hiện trong một loạt tính toán phụ thuộc vào các công thức. Có 2 cách chính của các phương pháp cắt tỉa:

- Dựa vào độ đo có ý nghĩa được giao tới các điểm xương.
- Dựa vào làm mịn đường biên trước khi trích chọn xương.

Việc làm mịn đường cong vẫn có một vài vấn đề đáng kể đó là làm thay đổi vị trí của xương và khó khăn trong việc phân biệt nhiễu từ thông tin hình dạng ở tần số thấp trên đường biên.

Một tiến bộ lớn đã từng được làm trong phương pháp cắt tia xương đó là dựa vào độ đo có ý nghĩa tới các điểm xương. Khi định nghĩa độ đo có ý nghĩa cho các điểm xương sẽ loại bỏ các điểm xương có ý nghĩa thấp.

Skaked và Bruckstein đưa ra các phân tích hoàn chỉnh và so sánh các phương pháp cắt tia để độ đo có ý nghĩa phổ biến của các điểm xương cùng với việc lan truyền vận tốc, độ dày tối đa, hàm bán kính chiều dài trục, chiều dài của đường biên được mở rộng.

Ogniewicz et al đưa ra một vài độ đo có ý nghĩa cho việc cắt tia phần thừa của xương mà không cần ngắt kết nối các xương.

Siddiqi et al. nối thông lượng đo lường với phương pháp làm mảnh để trích xuất mảnh mẽ và chính xác kết nối xương.

Tuy nhiên lỗi trong tính toán thông lượng bị giới hạn bởi độ phân giải điểm ảnh và cũng tỉ lệ thuận với đường biên của tiến hóa đường biên trước đó. Điều này làm cho vị trí chính xác của các điểm cuối trở nên khó khăn.

Torsello et al. đã khắc phục vấn đề này bằng cách lấy trong tập các biến thể của mật độ do biên độ cong và loại bỏ đường cong đóng góp lỗi.

Gần đây, Bai et al. đưa ra phương pháp cắt tia mới bằng cách chia đường biên thành các phân đoạn riêng biệt với các đỉnh từ DCE. Do DCE không thay đổi cấu trúc liên kết, cắt tia xương có cấu trúc của xương đầu vào. Mặc dù các đỉnh lỗi từ DCE có thể cắt tia xương để nhận được cấu trúc xương mới nhưng nó cũng tạo ra những nhánh xương không quan trọng do đó phương pháp đề xuất tiếp tục giải quyết những hạn chế của phương pháp DCE. Trong phương pháp này tác giả thực hiện loại bỏ những nhánh xương có liên quan thấp nhất tới việc khôi phục lại hình dạng. Phương pháp này cho phép vượt qua sự bất ổn của xương và thu được nhiều điểm xương ổn định.

## **3.2 Phương pháp DCE**

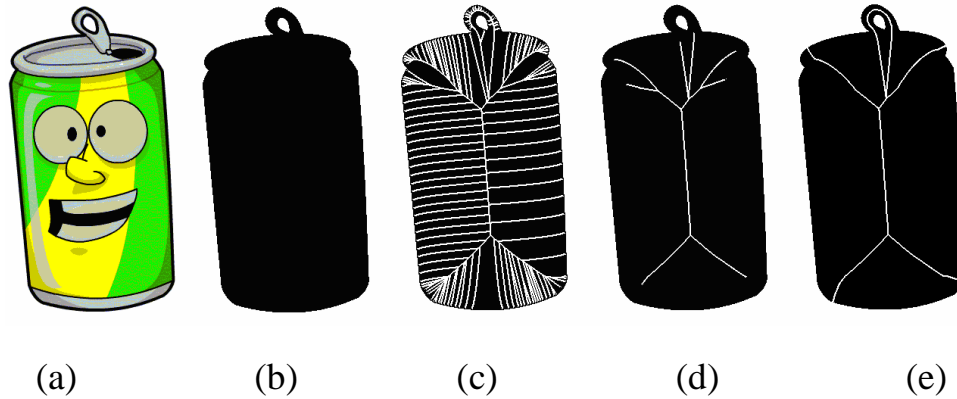
### **3.2.1 Giới thiệu**

Có hai phương pháp cắt tia xương chính:

- Dựa trên phương pháp đo lường tới các điểm xương.
- Dựa trên làm mịn đường biên trước khi phát hiện xương.

Những phương pháp cắt tia trên có những hạn chế sau:

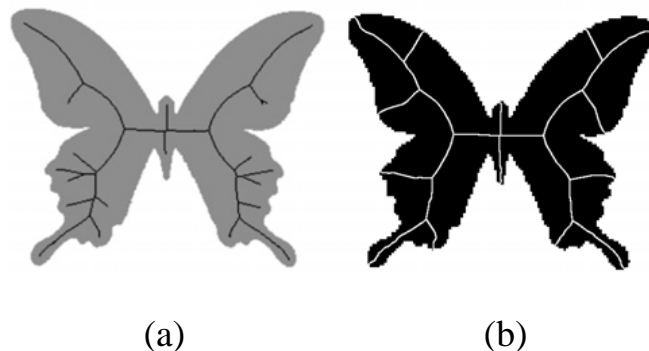
Hạn chế 1 là các phương pháp không đảm bảo được hình học cơ bản của vật thể đối với những hình dạng phức tạp (ví dụ hình có lỗ). Điều này được minh họa trong hình 3.2, xương thu được dựa vào độ đo có ý nghĩa (d) vi phạm hình học của xương đầu vào (c).



Hình 3.1. Minh họa hạn chế 1

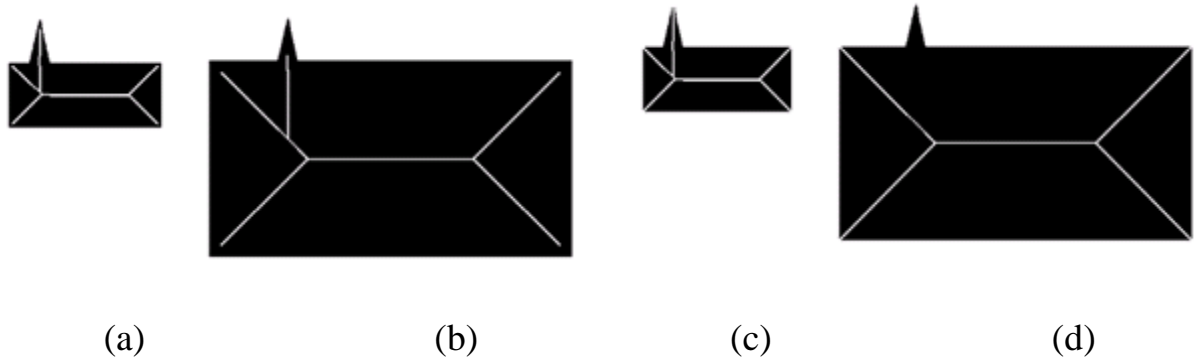
Trong hình 3.1: (a) đối tượng đầu vào; (b) mặt nạ đối tượng nhị phân; (c) bộ xương ban đầu; (d) cắt tia xương của đối tượng dựa vào độ đo có ý nghĩa; (e) cắt tia xương theo phương pháp đề xuất.

Hạn chế 2 là những nhánh xương chính của bộ xương bị nhấn đi và những nhánh xương ngắn không bị loại bỏ hoàn toàn. Điều này có thể làm mất nhiều thông tin hình dạng quan trọng và nó làm ảnh hưởng nghiêm trọng tới cấu trúc của xương.



Hình 3.2. Minh họa hạn chế 2

Hạn chế 3 là thường chỉ quan tâm đến những thông tin cục bộ của những điểm coi như là điểm xương và những thông tin toàn cục bị loại bỏ.



Hình 3.3. Minh họa hạn chế 3.

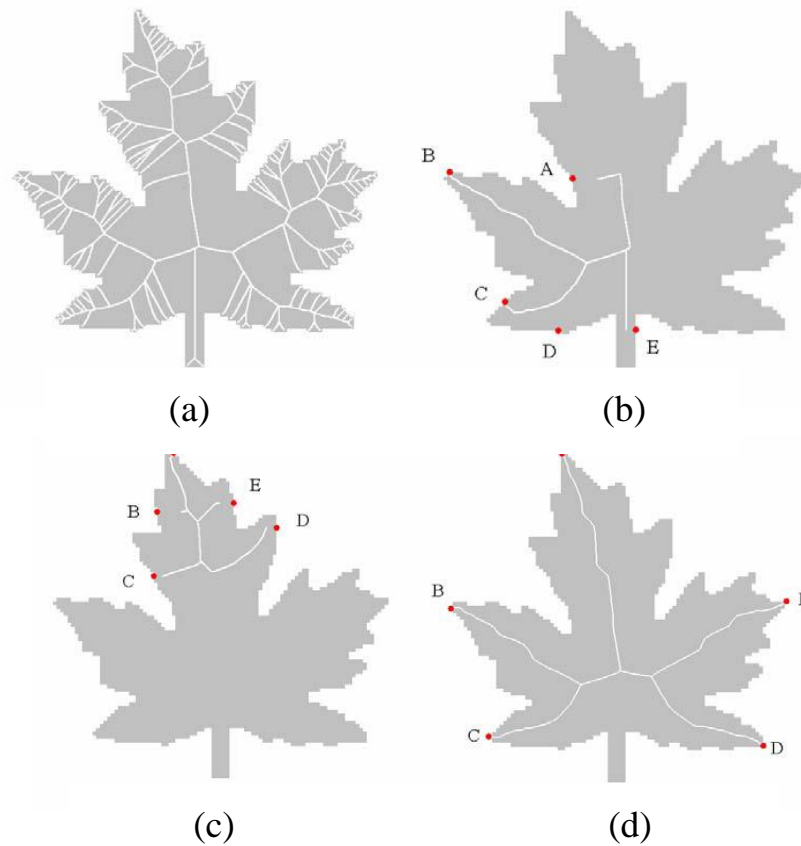
Với cùng một nguồn đầu vào kết quả ở (b) dựa vào độ đo có ý nghĩa, kết quả này không chú ý hình dạng toàn cục của vật thể do nhiễu. Phương pháp đề xuất đã duy trì được hình dạng nguyên bản của vật thể (d).

Hạn chế thứ 4 là kết quả của cắt tia xương có thể khác nhau đối với những điểm đầu vào nhọn.

### 3.2.2 Ý tưởng chính

Nhóm tác giả Xiang Baia, Login Jan Latec ki, Wen-Yu Liu đã đề xuất một phương pháp loại bỏ hoàn toàn những điểm lỗi ra mà không loại bỏ những điểm biên, vì vậy không loại bỏ những điểm xương chính. Những điểm sai hoặc thừa ra hoàn toàn bị loại bỏ trong khi những nhánh xương chính không bị ngắt đi vì thế không bao gồm những hạn chế nêu trên. Phương pháp này có thể cắt tia xương dựa trên việc phân chia đường biên thành những đoạn cong.

Ý tưởng chính là di chuyển tất cả điểm xương của các điểm tăng trưởng nằm trên cùng một đoạn đường biên. Hình 3.4 minh họa 3 phương pháp cắt tia xương khác nhau (b, c, d) với cùng xương đầu vào là (a).



Hình 3.4. Cắt tia xương với phân chia đường biên

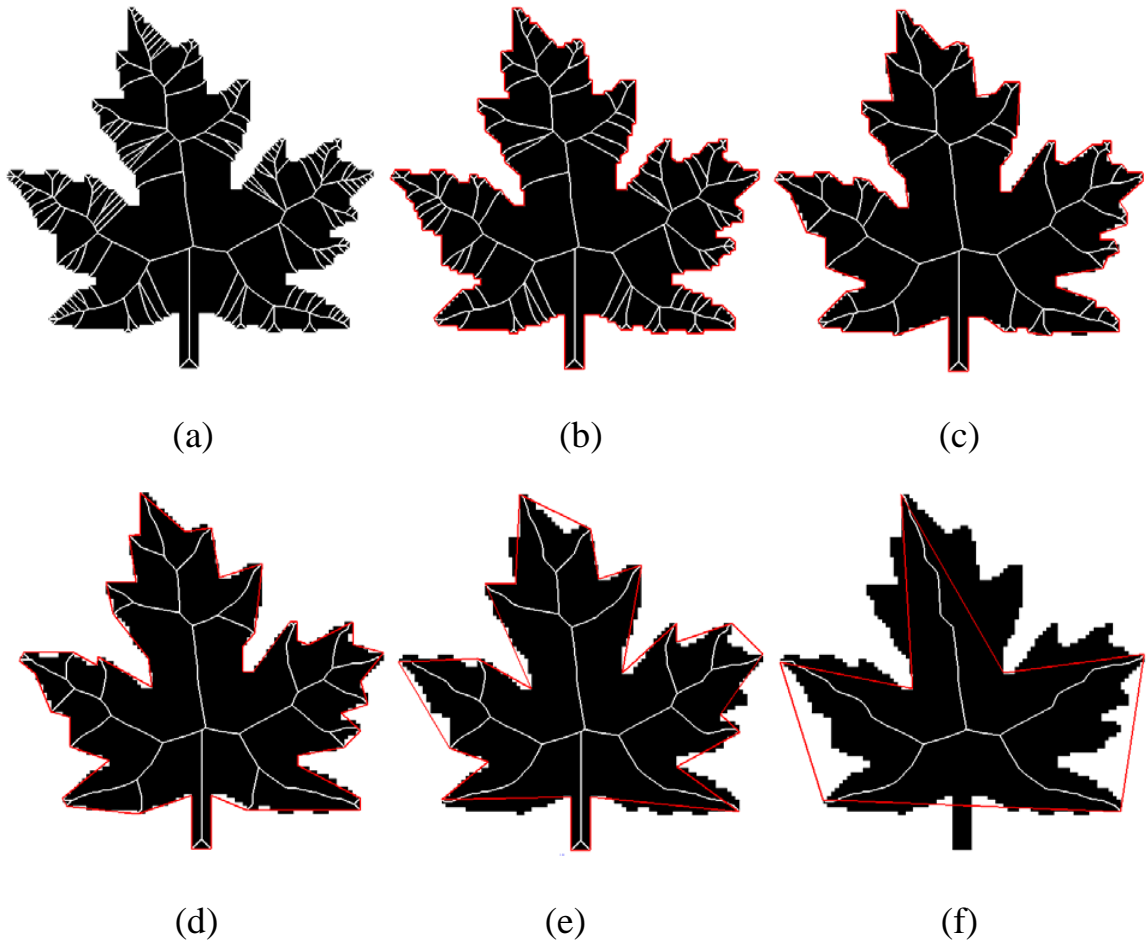
Cắt tia xương (a) với sự chú ý phân chia đường biên gây ra bởi 5 điểm ngẫu nhiên trên đường biên trong (b) và (c). 5 điểm trong (d) được tạo ra bởi DCE.

Xương cắt tia dựa trên 3 phương pháp phân chia đường biên khác nhau với điểm kết thúc được đánh dấu chấm. Ví dụ bỏ tất cả các điểm xương của điểm tăng trưởng trong đoạn biên CD trong (c) dẫn đến loại bỏ một phần dưới của xương, rõ ràng phân chia đường biên trong (d) cho kết quả cắt tia tốt hơn phân chia khác trong (b), (c). Từ đó đặt ra câu hỏi làm thế nào để tìm các đoạn phân chia tốt nhất. Tác giả có được sự phân chia như vậy nhờ quá trình DCE được giới thiệu ngắn gọn như sau: đầu tiên quan sát rằng đường biên của ảnh số có thể được biểu diễn như là một đa giác hữu hạn mà không bị mất thông tin do hình ảnh có độ phân giải hữu hạn. Tác giả giả định rằng các đỉnh của đa giác thu được từ lấy mẫu đường biên của đối tượng liên tục với một vài lỗi lấy mẫu. Số điểm như vậy phụ thuộc vào độ lệch chuẩn của lỗi lấy mẫu. Quá trình

DCE giúp loại bỏ các điểm nhiễu bằng loại bỏ đệ quy các đỉnh đa giác với sự đóng góp hình dạng nhỏ nhất (mà là nhiều khả năng kết quả từ nhiễu). Khi đó ta có được một tập hợp con các đỉnh tốt nhất tiêu biểu cho hình dạng đường biên. Tập hợp con này cũng có thể được xem như là sự chia ra của đường biên đa giác gốc thành những đoạn đường biên liên tục của đa giác đơn giản.

### 3.2.3 Rời rạc hóa đường cong với DCE

DCE loại bỏ những biến dạng đó bằng việc đơn giản hóa hình dạng. Ví dụ như hình dưới đây minh họa một vài giai đoạn của DCE. Hình dạng của lá đơn giản hơn bởi DCE, trong khi vẫn bảo toàn toàn những phần trực quan chính.



Hình 3.5. Trình tự xương của lá

Để bất kỳ đường cong của ảnh số nào có thể coi như một đa giác mà không mất thông tin phải có số lượng đỉnh lớn để nghiên cứu các quá trình tiến hóa của các hình dạng đa giác.

Ý tưởng cơ bản cho sự phát triển của đa giác là:

Trong mỗi bước lặp, một cặp đường phân đoạn liên tiếp  $s_1, s_2$  được thay thế bằng một đường đơn nối các điểm cuối của  $s_1$  với  $s_2$ .

Phần chính của việc tiến hóa này là thứ tự của việc thay thế. Thay thế được thực hiện theo phép đo liên quan  $K$  được đưa ra bởi công thức:

$$K(s_1, s_2) = \frac{\beta_{s_1, s_2} l_{s_1} l(s_2)}{l_{s_1} + l(s_2)} \quad (3.1)$$

Trong đó:  $s_1, s_2$  là các cạnh của đa giác liên quan đến đỉnh  $v$ ,  $\beta(s_1, s_2)$  là góc quay tại đỉnh chung của đoạn  $s_1, s_2$ ,  $l$  là hàm tính độ dài trung bình đối với tổng độ dài của đường cong đa giác  $C$ .

Đầu vào là đường biên đa giác  $P$  với  $n$  đỉnh, DCE tạo ra chuỗi đa giác đơn giản  $P = P^n, P^{n-1}, \dots, P^3$  như vậy là  $P^{n-(k+1)}$  thu được bằng việc loại bỏ những đỉnh đơn giản  $v$  từ  $P^{n-k}$  của hình dạng mà đóng góp đo lường bởi  $K$  là nhỏ nhất.

Định nghĩa 4. Một thuộc tính quan trọng của DCE là giới thiệu việc phân chia thứ bậc của đa giác đầu vào  $P$ . Tập  $\{v_1, \dots, v_n\}$  là các đỉnh của  $P$  và tập  $\{u_1, \dots, u_m\} \subset \{v_1, \dots, v_n\}$  là các đỉnh lồi của  $P^{n-k}$  với  $m \leq n-k$ . Với mức  $n-k$  của việc phân chia hệ thống phân cấp  $H_{n-k}(P)$ ,  $P$  bị phân tách thành  $m$  đường cong phụ:  $H_{n-k}(P) = \{[u_1, u_2], [u_2, u_3], \dots, [u_m, u_1]\}$ . Các đường cong này là phân chia đường cong DCE (trên DCE mức  $n-k$ ).

Nếu đỉnh  $u_i$  được xóa trong bước kế tiếp, hoặc trở thành lõm (do xóa một trong các đỉnh gần nó), khi đó đường cong  $[u_{i-1}, u_{i+1}]$  thay cho đường cong  $[u_{i-1}, u_i], [u_i, u_{i+1}]$  trong mức chia  $H_{n-(k+1)}(P)$ .

DCE và phân chia hệ thống phân cấp có thể được định nghĩa cho một tập hữu hạn của các đường cong đa giác. Sự khác biệt duy nhất là trong mỗi bước DCE là một đỉnh đơn bị loại từ một trong các đa giác mà phép đo liên quan của đỉnh đơn đó là nhỏ nhất. Điều này giúp cho phương pháp cắt tia có thể áp dụng cho tập mặt phẳng  $D$  như



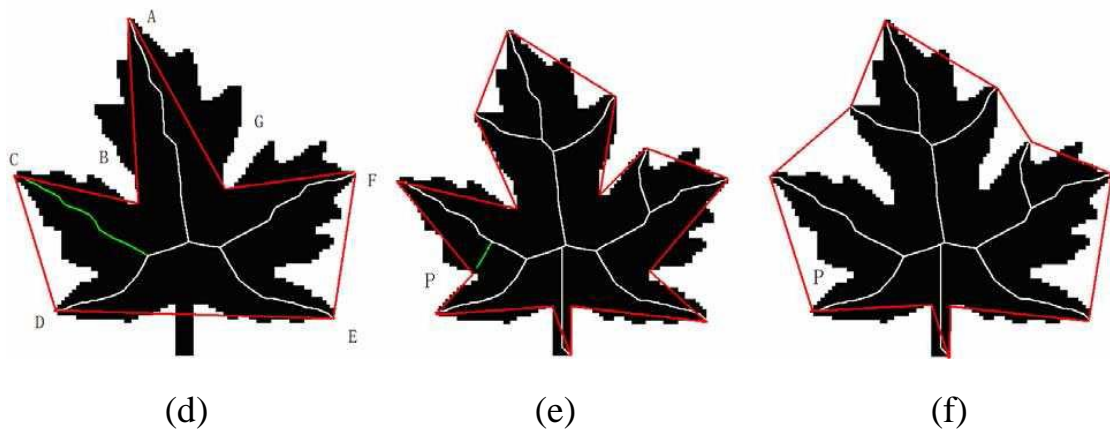
đường biên  $\partial D$  được bao gồm một số lượng hữu hạn của đường cong đóng đơn giản. Như vậy có thể áp dụng cho  $D$  có kết nối phức tạp.

Quy trình DCE có thể loại bỏ hiệu quả nhiễu và trực quan không quan trọng của hình ảnh, một tham số riêng dừng đúng lúc vẫn cần thiết. Nói cách khác, tìm kiếm  $k$  để đa giác đơn giản miêu tả chi tiết đường biên đầu vào. Để định lượng mức độ chi tiết tác giả định nghĩa khoảng cách trung bình  $D_{av}(P^{n-k})$  giữa các điểm góc của  $P$  và các đường phân đoạn tương ứng của nó trong  $P^{n-k}$ .

Tác giả đưa ra ngưỡng  $T$  để dừng DCE nếu  $D_{av}(P^{n-k}) > T$  cho một vài  $k$ . Đưa ra chuỗi giá trị của  $T$ , tác giả có thể thu được một trình tự đơn giản hóa của đường cong đa giác DCE dẫn tới trình tự đơn giản hóa của xương tương ứng. Nói chung, một điều kiện dừng thích hợp phụ thuộc vào ứng dụng cụ thể.

### 3.2.4 Cắt tia xương với DCE

Cho xương  $S(D)$  của mặt phẳng  $D$  và đưa ra một DCE đa giác đơn giản  $P^k$ . Thực hiện cắt tia xương bằng việc di chuyển tất cả các điểm xương  $s \in S(D)$  như là các điểm tăng trưởng  $Tan(s)$  của  $s$  được chứa trong cùng đoạn DCE mở. Mỗi điểm cắt tia xương  $s$  là kết quả từ phần đường biên cục bộ với liên quan phân vùng DCE, vì vậy  $s$  có thể coi như là điểm xương không quan trọng và có thể loại bỏ. Quá trình đơn giản hóa đường biên với DCE giúp hoàn thành cắt tia các nhánh của xương. Đặc biệt loại bỏ đỉnh lồi  $v$  từ  $P^{n-k}$  bởi DCE thu được  $P^{n-(k+1)}$  để hoàn thành loại bỏ những nhánh xương kết thúc tại  $v$ . Ví dụ minh họa cho việc đơn giản hóa đường biên với DCE.



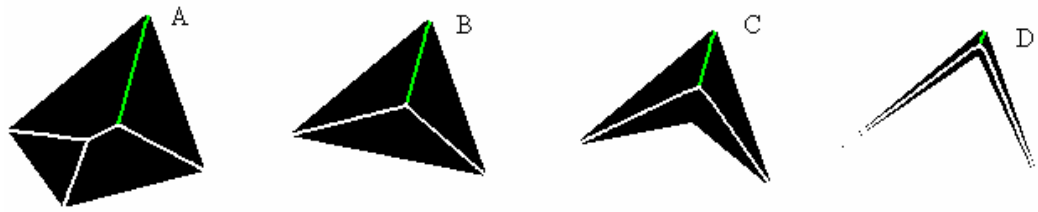
Hình 3.6. Minh họa cắt tia xương với DCE

Trong hình 3.6: hình (a) biểu diễn một hình đa giác với 7 đỉnh thu được bởi DCE và xương của đối tượng được cắt tỉa dựa trên đa giác đó, trong hình này chỉ có 5 nhánh xương kết thúc tại 5 đỉnh lồi của đa giác đơn giản. Cắt tỉa xương được tính toán với việc liên quan tới các đoạn DCE là các đoạn (A, C), (C, D), (D, E), (E, F), (F, A). Nhánh xương màu xanh trong hình (a) kết thúc tại C vẫn còn vì nó tiếp tuyến với đường tròn lớn nhất trên hai đoạn DCE khác nhau là cung (A, C) và (C, D). Hình (b) nhánh xương màu xanh kết thúc tại đỉnh P không thuộc về xương được xác định bởi DCE vì nhánh xương đó kết thúc tại đỉnh lõm P. Trong hình (c) nó được loại bỏ bởi đơn giản hóa DCE

Tác giả thực hiện phân tách đường biên thành các đoạn DCE dựa trên đỉnh lồi của đơn giản hóa DCE. Điều này không chỉ đưa ra đỉnh bị loại bỏ bởi DCE mà còn thay thế đỉnh lồi thành đỉnh lõm trong quá trình xử lý của DCE, nhánh xương cuối kết thúc tại đỉnh lõm bị loại bỏ. Điều này cho phép loại những nhánh nhỏ trong những giai đoạn trước đó của quá trình DCE. Hình 3.6(a) minh họa tại sao chỉ sử dụng các đỉnh lồi để định nghĩa đoạn DCE. Nhánh xương màu xanh trong hình 3.6(b) kết thúc tại đỉnh P là một phần của của xương nếu chỉ sử dụng đỉnh lõm của đa giác đơn giản (biểu diễn bởi màu đỏ) để định nghĩa các đoạn DCE. Nhưng nhánh xương kết thúc tại P bị loại bỏ trong hình 3.6(c) được định nghĩa bằng việc sử dụng các đỉnh lồi của đa giác đơn giản điều này cho phép cắt tỉa nhanh hơn những nhánh xương không liên quan.

Một thuộc tính rất quan trọng của DCE là gây ra phân chia đường biên và mỗi phân chia đó làm giảm các đỉnh của đa giác. Thực tế là có một nhánh xương kết thúc tại mỗi điểm phân chia. Theo kết quả trên, trong mỗi bước tiến hóa DCE nếu đỉnh  $u_i$  của đa giác bị xóa (tức là  $u_i \in P^{n-k} - P^{n-(k+1)}$  hoặc trở thành lõm (do việc xóa đi một trong những đỉnh bên cạnh nó) thì cung  $[u_{i-1}, u_{i+1}]$  thay thế cung  $[u_{i-1}, u_i]$ ,  $[u_i, u_{i+1}]$ . Vì vậy toàn bộ những nhánh xương kết thúc tại đỉnh  $u_i$  bị loại bỏ.

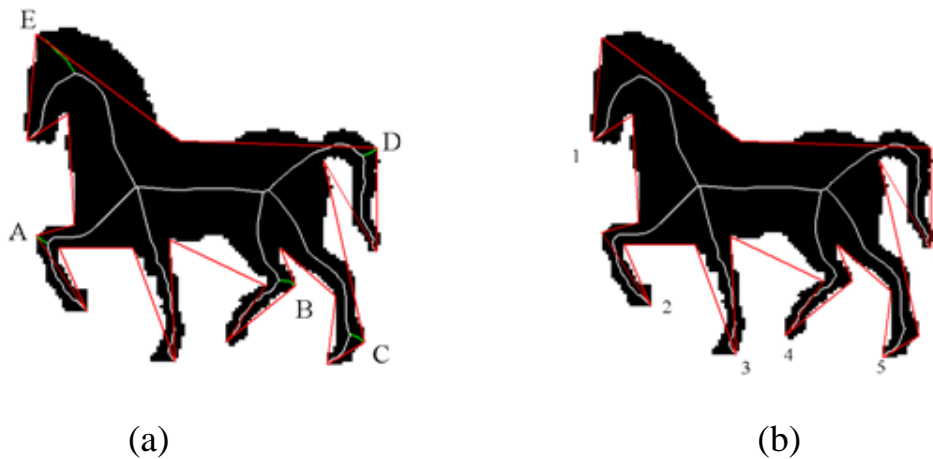
Mặc dù các đỉnh lồi từ DCE có thể cắt tỉa xương để nhận được cấu trúc xương mới nhưng nó cũng có thể tạo ra những nhánh xương không quan trọng Điều này được minh họa trong hình 3.7.



Hình 3.7. Các đỉnh lồi như nhau có thể sinh ra các nhánh xương khác nhau với mức quan trọng khác nhau

Đỉnh A, B, C và D có cùng DCE theo phép đo  $K$ . Tuy nhiên 4 nhánh xương màu xanh kết thúc tại đó có tầm quan trọng khác nhau. Nhánh xương kết thúc tại D có mức quan trọng thấp hơn và có thể bị loại bỏ. Do đó đỉnh lõm bên trong hình dạng với các đỉnh C và D, sự quan trọng của các nhánh xương kết thúc tại đỉnh lồi C và D bị giảm đánh kể.

Để vượt qua vấn đề này tác giả giới thiệu thêm đo lường thích hợp bằng việc tính toán khoảng cách cách  $D_l(v)$  giữa  $v$  và đỉnh lồi  $u$  gần nhất như đoạn  $vu$  bên trong hình dạng nếu như đỉnh  $u$  tồn tại



Hình 3.8. Loại bỏ đỉnh lồi không quan trọng tạo ra hình ảnh xương tối ưu

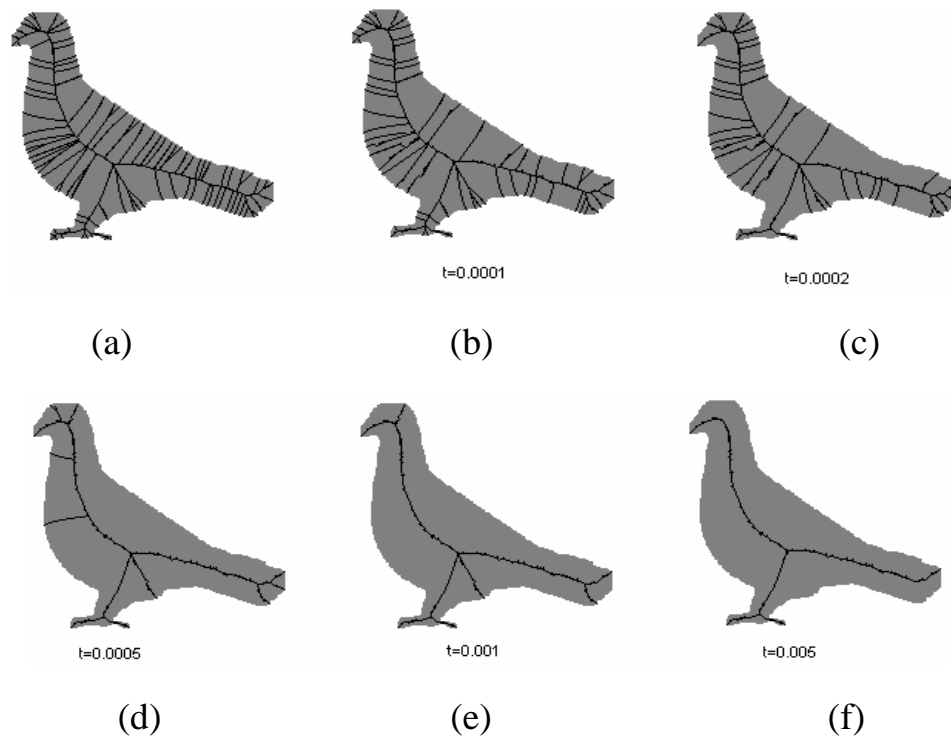
Hình 3.8 minh họa tác dụng của việc loại bỏ đỉnh lồi  $v$  với  $D_l(v)$  thấp. Có 5 nhánh xương ngắn (màu xanh) kết thúc tại A, B, C, D, E trong hình 3.8(a) 5 nhánh xương này bị loại bỏ trong hình 3.8(b). Việc phân chia chỉ với 7 đỉnh được đánh số từ 1 đến 7 trong hình 3.8(b).

Tóm lại, các đỉnh  $V_f$  được sử dụng cho việc phân chia đường biên gây ra bởi DCE được tính toán như:  $V_f = V_s - (V_{concave} \cup V_l)$ , đỉnh  $V_s$  biểu thị tất cả các đỉnh của đa giác đơn giản hóa  $P$  thu được bởi DCE,  $V_{concave}$  biểu thị tất cả các đỉnh lồi của  $V_s$ , và  $V_s$  với giá trị thấp của phép đo  $D_l$ .

### 3.3 Phương pháp DSE

#### 3.3.1 Ý tưởng chính

Việc thực hiện đối sánh xương phụ thuộc trực tiếp vào đặc tính của việc biểu diễn hình dạng. Do đó để cắt tia những xương um tùm trong các hình dạng xương là điều thường xuyên không thể tránh khỏi. Kết quả nghiên cứu của nhóm tác giả Xiang Baia và Login Jan Latec ki đã giới thiệu phương pháp mới cho việc cắt tia xương gọi là DSE (Discrete Skeleton Evolution). Ý tưởng của phương pháp DSE là loại bỏ những nhánh cuối của xương sẽ không làm thay đổi cấu trúc của hình dạng gốc, ý tưởng này kế thừa từ việc làm gàn đây của Bai et al. Nhóm tác giả thu được một cách tự nhiên cấu trúc phân cấp của những xương được đơn giản hóa như minh họa ở hình 3.9.



Hình 3.9. Quá trình tiến hóa bộ xương thu được trong vòng lặp cắt tia xương của một con chim

### 3.3.2 Các định nghĩa

Trước khi đưa ra phương pháp được đề xuất, tác giả đưa ra một số định nghĩa.

Theo định nghĩa của Blum về trục trung vị, xương  $S$  của tập  $D$  là các quỹ tích các tâm của các đường tròn lớn nhất. Một đường tròn lớn nhất của  $D$  là một đường tròn đóng chứa trong  $D$ . Đường tròn này tiếp tuyến trong tới đường biên  $\partial D$  và không chứa trong bất kỳ đường tròn nào khác trong  $D$ . Mỗi một đường tròn tối đa phải tiếp tuyến trong tới đường biên ít nhất hai điểm. Với rất nhiều điểm xương  $s \in S$  tác giả lưu trữ bán kính  $r(s)$  của đường tròn lớn nhất.

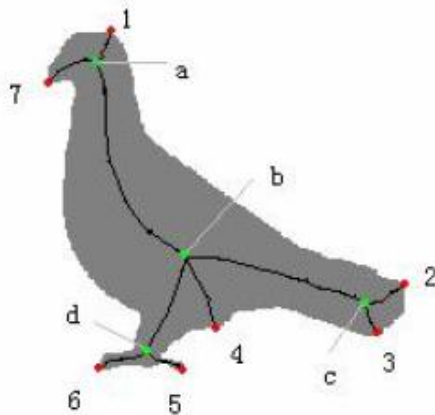
Xương  $S$  trong đồ thị hình học có thể bị phân hủy trong hữu hạn các điểm của các vòng cung được kết nối gọi là các nhánh xương bao gồm các điểm ở mức độ hai, và các nhánh gặp ở các khớp xương (hoặc các điểm phân nhánh) đó là các điểm ở mức độ ba hoặc cao hơn.

**Định nghĩa 1.** Điểm xương chỉ có một điểm tiếp giáp gọi là điểm cuối (hay điểm xương cuối); điểm xương có hơn hai điểm tiếp giáp là điểm giao nhau. Nếu một điểm xương không là điểm cuối hay điểm giao nhau thì nó là điểm kết nối. (Ở đây tác giả định nghĩa các đường cong của xương có độ rộng là một điểm ảnh)

**Định nghĩa 2.** Một nhánh xương cuối là một phần xương giữa điểm xương cuối và điểm giao nhau gần nhất. Cho  $l_i$  ( $i=1,2,3,\dots,N$ ) là các điểm cuối của xương  $S$ . Với mỗi điểm cuối  $l_i$ ,  $f(l_i)$  biểu thị điểm giao nhau gần nhất. Về mặt hình thức, một nhánh xương cuối  $P(l_i, f(l_i))$  là đường dẫn xương ngắn nhất giữa  $l_i$  và  $f(l_i)$ .

Ví dụ, trong hình 2 đường cong từ 1 đến a là nhánh xương cuối:  $P(1, f(1)) = P(1, a)$ . Đường cong từ a đến b không phải là nhánh xương cuối, nó là nhánh xương trong. Quan sát thấy rằng điểm a là điểm giao nhau gần nhất của hai điểm cuối (1 và 7).

Dựa vào định nghĩa của Blum về xương, điểm xương  $s$  phải là tâm của đường tròn lớn nhất chứa hình dạng  $D$ .

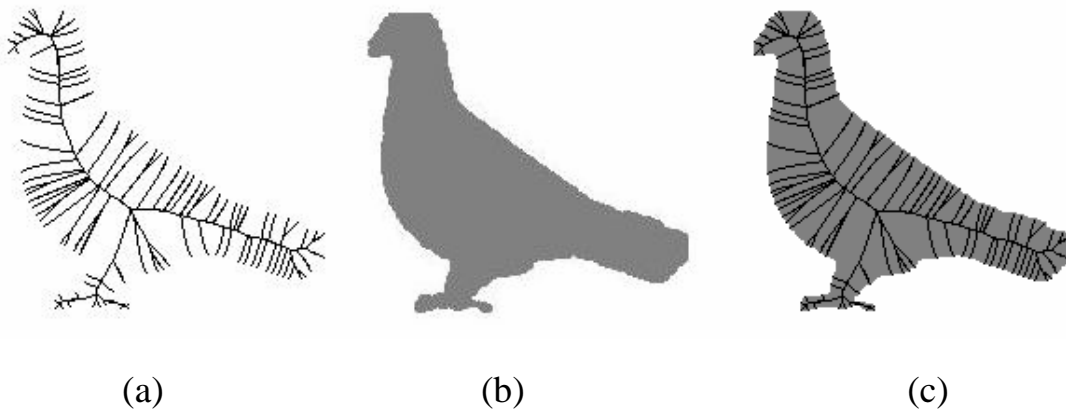


Hình 3.10. Các điểm xương cuối và các điểm giao nhau.

Trong hình 3.10: các điểm xương cuối (màu đỏ), các điểm giao nhau (màu xanh)

**Định nghĩa 3.** Tập  $r(s)$  biểu thị bán kính của đường tròn lớn nhất  $B(s, r(s))$  tâm tại điểm xương  $s$ . Xây dựng lại xương  $S$  được biểu thị bởi  $R(S)$  và được đưa bởi công thức

$$R(S) = \bigcup_{s \in S} B(s, r(s)) \quad (3.2)$$



Hình 3.11. Khôi phục lại hình dạng gốc từ xương.

Như minh họa ở hình 3.11, có thể khôi phục lại hình dạng gốc từ xương của chúng. Cắt tỉa có thể xem như đơn giản hóa xương và có thể chứa đựng đầy đủ thông tin của hình dạng xương.

Tác giả giới thiệu một thuật toán để cắt tia xương. Trong mỗi bước, một nhánh xương cuối có trọng số thấp nhất sẽ bị loại bỏ, có hai động cơ chính dẫn đến việc làm trên:

- Loại bỏ một nhánh xương cuối sẽ không làm thay đổi cấu trúc liên kết của bộ xương.
- Nhánh xương với đóng góp thấp tới việc khôi phục lại hình dạng gốc được loại bỏ đầu tiên.

Tác giả định nghĩa trọng số  $w_i$  cho mỗi nhánh xương cuối  $P(l_i, f(l_i))$  như sau:

$$w_i = 1 - \frac{A(R(S - P(l_i, f(l_i))))}{A(R(S))} \quad (3.3)$$

Hàm  $A(\ )$  là hàm diện tích. Trực giác cho cắt tia xương là nhánh xương cuối với trọng số nhỏ  $w_i$  có ảnh hưởng không đáng kể tới việc xây dựng lại hình dạng xương. Khi diện tích của vùng khôi phục lại không có nhánh xương này gần giống như diện tích của vùng xây dựng lại có nhánh xương đó. Vì vậy nhánh xương trên có thể bị loại bỏ. Đề xuất cắt tia xương dựa trên vòng lặp của việc loại bỏ những nhánh cuối tới khi đáp ứng được ngưỡng mong muốn.

### 3.3.3 Thuật toán DSE

Thuật toán cắt tia xương được cho như sau:

1. Tác giả khởi tạo các trọng số cho tất cả các nhánh xương cuối  $w_i^{(0)}$  ( $i = 1, 2, \dots, N^{(0)}$ ) dựa trên xương gốc  $S^{(0)}$ :

$$w_i^{(0)} = 1 - \frac{A(R(S^{(0)} - P(l_i^{(0)}, f(l_i^{(0)}))))}{A(R(S^{(0)}))} \quad (3.4)$$

2. Trong lần lặp thứ  $k$ , cho  $i = 1, 2, \dots, N^{(k)}$  tính toán trọng số cho mỗi nhánh xương cuối trong xương  $S^{(k)}$ :

$$w_i^k = 1 - \frac{A(R(S^{(k)} - P(l_i^k, f(l_i^{(k)}))))}{A(R(S^{(k)}))} \quad (3.5)$$

3. Chọn trọng số tối thiểu  $w_{\min}^{(k)}$ . Nếu  $w_{\min}^{(k)}$  nhỏ hơn ngưỡng  $t$ , thực hiện bước 4; ngược lại thì dừng và đưa ra  $S^{(k)}$  là kết quả cuối cùng.
4. Loại bỏ nhánh xương cuối  $P_{\min}^{(k)}$  với trọng số thấp nhất  $w_{\min}^{(k)}$  và có được xương mới:

$$S^{(k+1)} = S^{(k)} - P_{\min}^{(k)} \quad (3.6)$$

5. Đặt  $k=k+1$  và quay lại bước 2.



## CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

### 4.1 Môi trường cài đặt

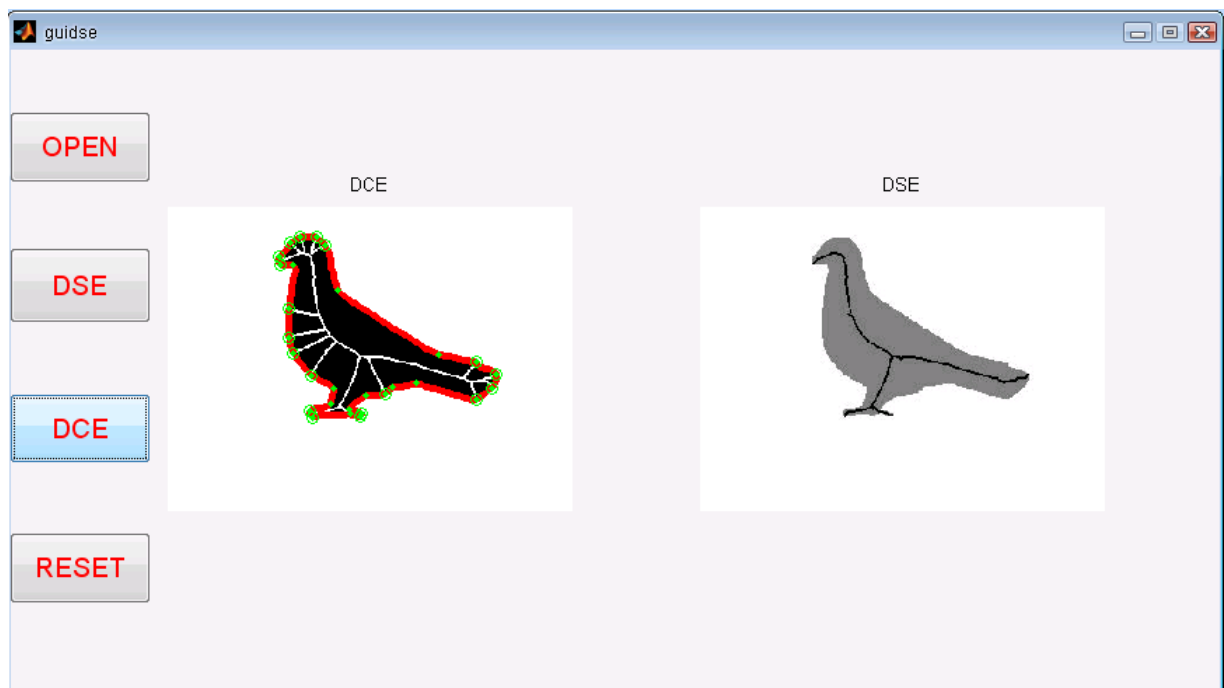
Chương trình được cài đặt trên Môi trường Windows XP, sử dụng ngôn ngữ Matlab với máy tính có cấu hình như sau:

- CPU: Intel® Pentium E5500(2.8 GHz)
- HDD: 160 GB
- Memory: 2GB

Tập dữ liệu được sử dụng trong thử nghiệm là tập dữ liệu thuộc: MPEG-7

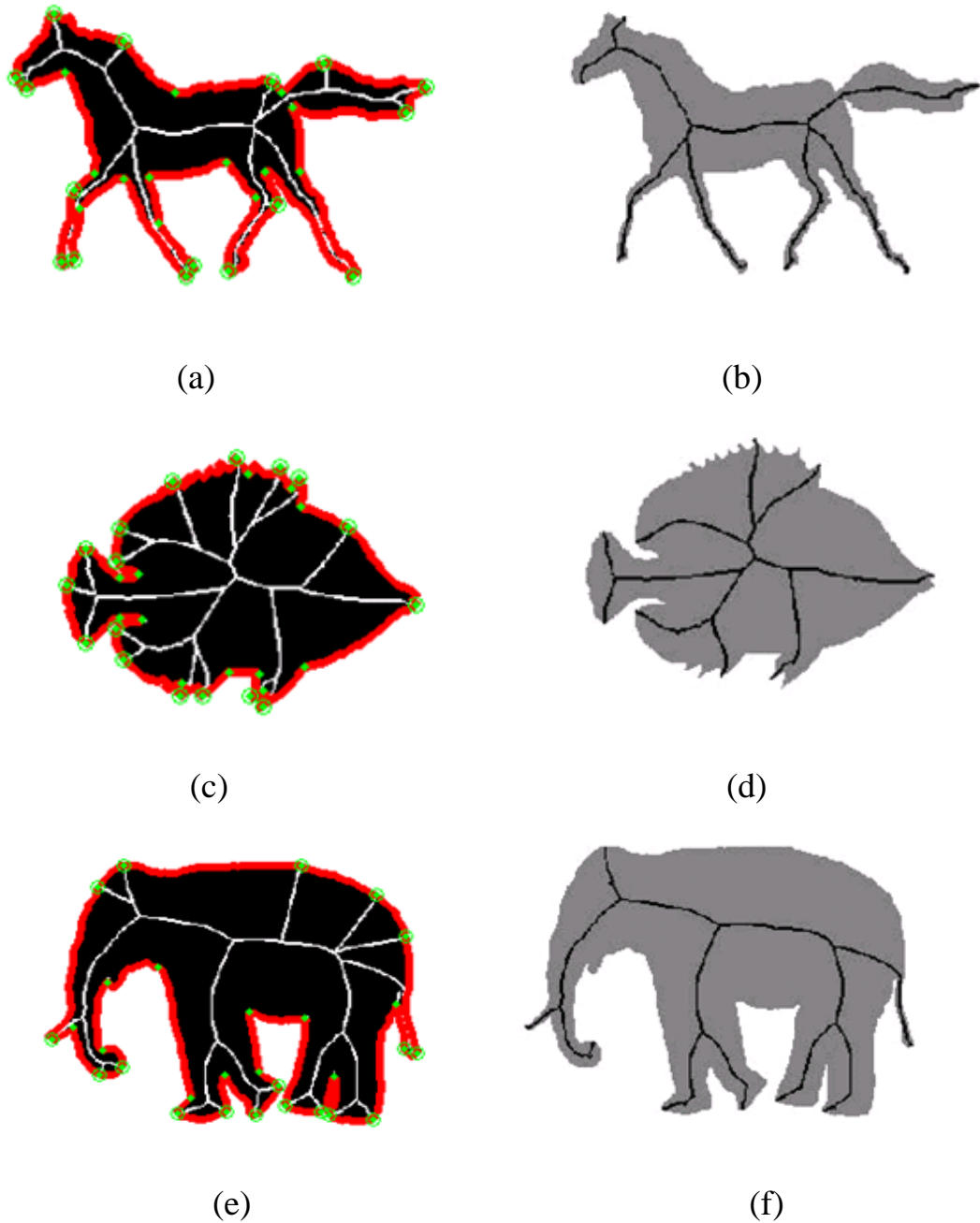
### 4.2 Chương trình thực nghiệm

#### 4.2.1 Giao diện chương trình



Hình 4.1. Giao diện chương trình

#### 4.2.2 So sánh kết quả tìm xương với các phương pháp DCE

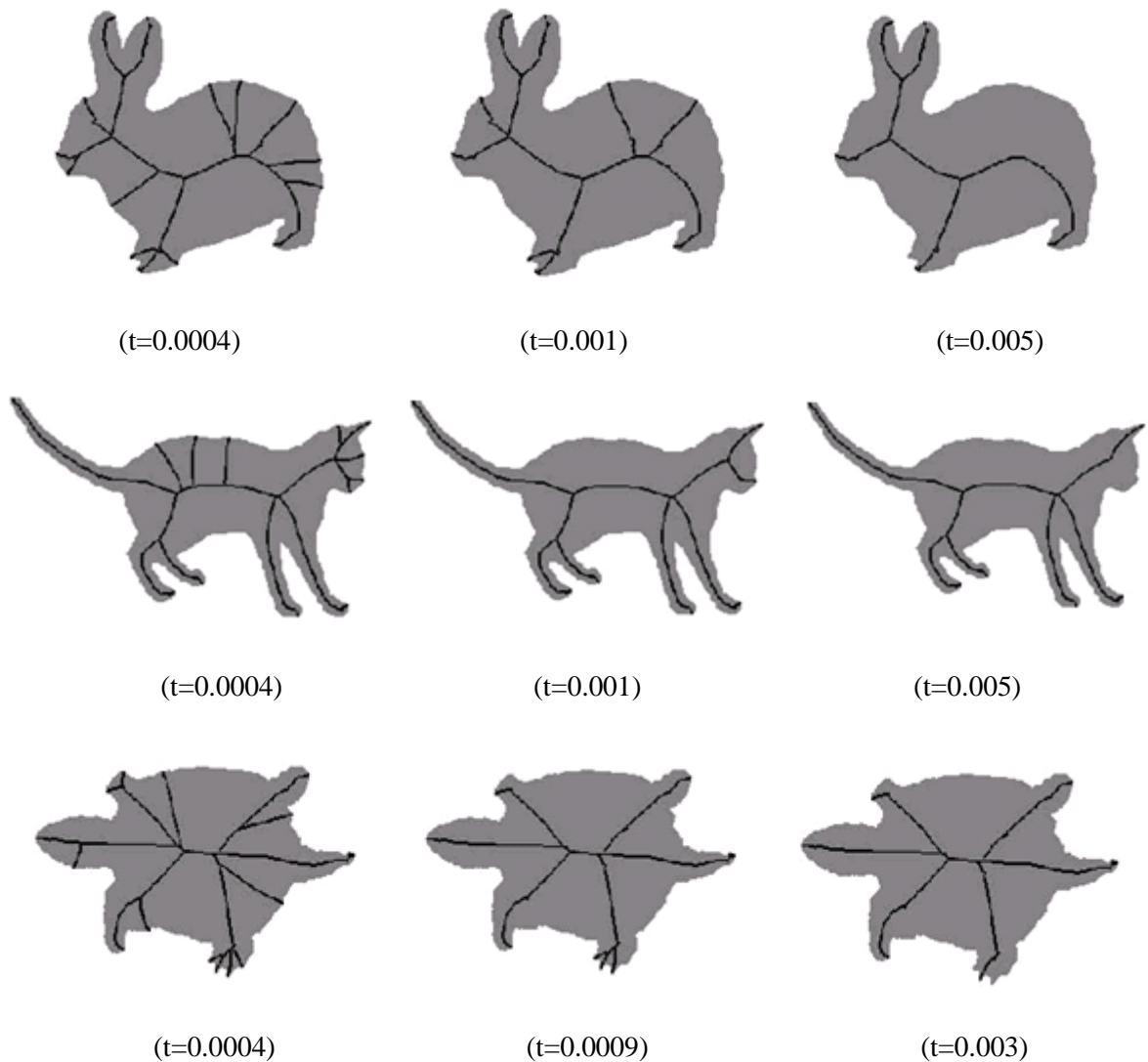


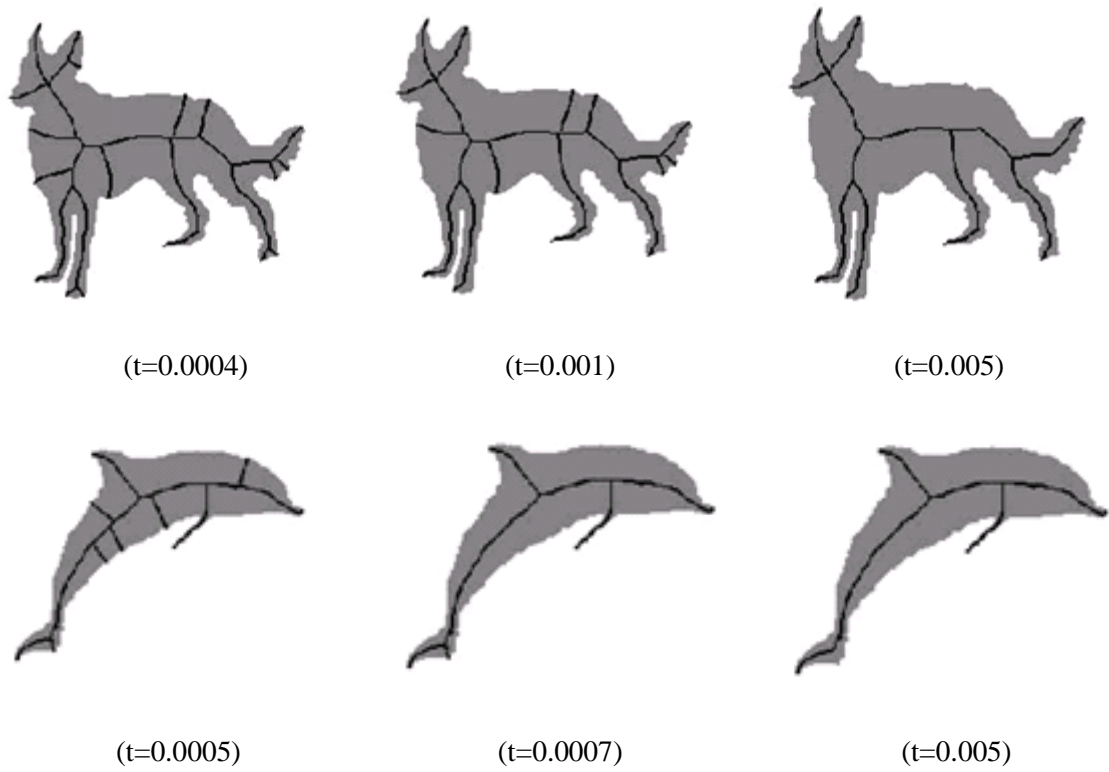
Hình 4.2. Xương thu được bằng phương pháp DCE và DSE

Trong hình 4.2: hình (a), (c), (e) là xương thu được theo phương pháp DCE với số đỉnh được đơn giản hóa bởi DCE là 29; hình (b), (d), (f) là xương thu được theo phương pháp DSE ngưỡng bằng 0,0003.

### 4.2.3 Hiệu quả của việc sử dụng ngưỡng (threshold)

Hiệu quả của các giá trị ngưỡng (threshold) khác nhau trên xương của một đối tượng được minh họa trong hình 4.3. Nhờ vào ngưỡng (threshold) phương pháp DSE có thể thực hiện tìm xương trên ảnh đầu vào với nhiều ngưỡng khác nhau để thu được nhiều hình dạng xương khác nhau. Khi giá trị ngưỡng (threshold) tăng, có ít hơn các nhánh trong xương. Tuy nhiên để thu được xương theo mong muốn thì việc chọn được ngưỡng phù hợp là rất cần thiết.





Hình 4.3. Xương thu được bằng phương pháp DSE theo các ngưỡng khác nhau,  $t$  là giá trị ngưỡng

## KẾT LUẬN

Đồ án “Tìm hiểu phương pháp DSE (Discrete Skeleton Evolution) cho bài toán tìm xương của ảnh” đã đạt được một số kết quả như sau:

**Về lý thuyết**, đồ án đã trình bày và hiểu được:

- Tổng quan về xử lý ảnh số.
- Một số hướng tiếp cận trong tìm xương của ảnh.
- Tìm hiểu thuật toán cắt tia xương của ảnh dựa vào DSE (Discrete Skeleton Evolution) do Xiang Baia, Login Jan Latec ki đề xuất.

**Về thực nghiệm**, đồ án đã tiến hành cài đặt thử nghiệm chương trình tìm xương và cắt tia xương dựa vào DSE (Discrete Skeleton Evolution) và so sánh với kết quả tìm xương với phương pháp DCE (Discrete Curve Evolution).

Tuy nhiên trong quá trình thực hiện, do năng lực còn nhiều hạn chế, nên đề tài mới chỉ dừng lại ở mức đọc, dịch hiểu và tìm hiểu tóm lược về phương pháp, chưa đánh giá tổng hợp được phương pháp. Nếu có điều kiện, em sẽ tìm đọc tài liệu để nghiên cứu nhằm tổng hợp nhiều phương pháp và đưa ra được những đánh giá kết luận dựa trên những gì đã tìm hiểu được.

Em rất mong nhận được sự đóng góp ý kiến của các Thầy Cô và các bạn để em có thêm kiến thức và kinh nghiệm tiếp tục hoàn thiện nội dung nghiên cứu trong đề tài.

Em xin chân thành Cảm ơn!

## TÀI LIỆU THAM KHẢO

### Tài liệu Tiếng Việt

[1]. Đỗ Năng Toàn, Phạm Việt Bình (2007), *Giáo trình xử lý ảnh*, Nhà xuất bản Đại học Thái Nguyên.

[2]. Nguyễn Thị Hoa (2010), *Tìm hiểu phương pháp DCE(Discrete Curve Evolution) cho bài toán tìm xương của ảnh*, Đồ án tốt nghiệp, Trường ĐHDL Hải Phòng.

[3]. Nguyễn Thị Lan (2011), *Tìm hiểu phương pháp BPR (Bending Potential Ratio) cho bài toán tìm xương của ảnh*, Đồ án tốt nghiệp, Trường ĐHDL Hải Phòng.

### Tài liệu Tiếng Anh

[4]. Xiang Baia, Login Jan Latec ki, Wen–Yu Liu (2007), *Skeleton Prunning by Contour Partitionning with Discrete Curve Evolution*, CVPR.

[5]. Xiang Baia, Login Jan Latec ki (2007), *Skeleton Prunning with Discrete Skeleton Evolution*, CVPR.