

LỜI CẢM ƠN

Trước hết em xin chân thành cảm ơn thầy Ngô Trường Giang là giáo viên hướng dẫn em trong quá trình làm đồ án. Thầy đã giúp em rất nhiều và đã cung cấp cho em nhiều tài liệu quan trọng phục vụ cho quá trình tìm hiểu về đề tài “Tìm hiểu kỹ thuật cắt tia xương của ảnh”.

Em xin chân thành cảm ơn các thầy cô trong bộ môn công nghệ thông tin đã chỉ bảo em trong quá trình học và rèn luyện trong 4 năm học vừa qua.

Em xin chân thành cảm ơn ban giám hiệu trường Đại Học Dân Lập Hải Phòng đã tạo điều kiện cho em có kiến thức, thư viện của trường là nơi mà sinh viên trong trường có thể thu thập tài liệu trợ giúp cho bài giảng trên lớp. Đồng thời các thầy cô trong trường giảng dạy cho sinh viên kinh nghiệm cuộc sống. Với kiến thức và kinh nghiệm đó sẽ giúp em cho công việc và cuộc sống sau này.

Cuối cùng em xin bày tỏ lòng biết ơn tới những người thân trong gia đình và các bạn bè đã chia sẻ và động viên em trong suốt quá trình học tập cho đến nay.

Em xin chân thành cảm ơn!

Hải Phòng, ngày tháng năm 2010

Sinh viên

Nguyễn Thị Hoa

MỤC LỤC

LỜI CẢM ƠN	1
MỤC LỤC.....	2
DANH MỤC HÌNH VẼ.....	4
LỜI MỞ ĐẦU	5
CHƯƠNG 1: TỔNG QUAN VỀ XỬ LÝ ẢNH.....	6
1.1 Xử lý ảnh là gì?.....	6
1.2 Các vấn đề cơ bản trong xử lý ảnh.....	7
1.2.1 Một số khái niệm cơ bản.....	7
1.2.2 Thu nhận ảnh.....	7
1.2.3 Nâng cao chất lượng ảnh.....	10
1.2.4 Trích chọn đặc điểm.....	11
1.2.5 Nhận dạng	12
1.2.6 Nén ảnh	14
CHƯƠNG 2: XƯƠNG VÀ CÁC KỸ THUẬT TÌM XƯƠNG	15
2.1 Giới thiệu	15
2.2 Tìm xương dựa trên làm mảnh ảnh.....	16
2.2.1 Sơ lược về thuật toán làm mảnh.....	16
2.2.2 Một số thuật toán làm mảnh.....	17
2.3 Tìm xương không dựa trên làm mảnh ảnh.....	18
2.3.1 Khái quát về lược đồ Voronoi.....	19
2.3.2 Trục trung vị Voronoi rời rạc.....	19
2.3.3 Xương Voronoi rời rạc.....	20
2.3.4 Thuật toán tìm xương.....	21
CHƯƠNG 3: KỸ THUẬT CẮT TỈA XƯƠNG CỦA ẢNH.....	26
3.1 Giới thiệu	26
3.2 Ý tưởng chính của phương pháp.....	29
3.3 Cắt tỉa xương với DCE	33

3.3.1	Rời rạc hóa đường cong	33
3.3.2	Cắt tia xương với DCE.....	34
CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM		38
4.1	Môi trường cài đặt.....	38
4.2	Chương trình	38
KẾT LUẬN		40
TÀI LIỆU THAM KHẢO.....		41

DANH MỤC HÌNH VẼ

Hình 1. 1. Quá trình xử lý ảnh.....	6
Hình 1. 2. Các bước cơ bản trong một hệ thống xử lý ảnh	6
Hình 1. 3. Quá trình hiển thị và chỉnh sửa, lưu trữ ảnh thông qua DIB.	9
Hình 1. 4. Sự chuyển đổi giữa các mô hình biểu diễn ảnh.....	9
Hình 1. 5. Ảnh thu nhận và ảnh mong muốn.....	10
Hình 2. 1. Ví dụ về ảnh và xương	15
Hình 2. 2. Xương Voronoi rời rạc.....	21
Hình 2. 3. Minh họa thuật toán trộn hai sơ đồ Voronoi.....	22
Hình 2. 4. Minh họa thuật toán thêm một điểm biên vào sơ đồ Voronoi	23
Hình 3.1. Minh họa xương của ảnh.....	26
Hình 3.2. Minh họa hạn chế 1.	28
Hình 3.3. So sánh kết quả của [7] (a) và của phương pháp đề xuất (b).....	28
Hình 3.4. Minh họa hạn chế 3.	29
Hình 3.5. Cắt tỉa xương với phân chia đường biên.....	30
Hình 3.6. Trình tự bộ xương của lá.....	32
Hình 3.7 Minh họa cắt tỉa xương với DCE	35
Hình 3.8. Loại bỏ đỉnh lỗi không quan trọng tạo ra xương với hình ảnh tối ưu. ...	37
Hình 4.1. Ảnh đầu vào	38
Hình 4.2. Xương của ảnh	39
Hình 4.3. Ảnh sau khi cắt tỉa xương	39

LỜI MỞ ĐẦU

Xương được coi như hình dạng cơ bản của đối tượng với số ít các điểm ảnh cơ bản và nó là cách biểu diễn đối tượng một cách cô đọng. Nó thường được ứng dụng trong rất nhiều lĩnh vực như đồ họa máy tính, tra cứu ảnh, nhận dạng ký tự. Các thuật toán tìm xương thường gặp phải vấn đề tạo ra xương có gai nên làm ảnh hưởng tới độ chính xác. Đề tài trình bày kỹ thuật cắt tia xương của ảnh để làm mịn xương.

Đồ án bao gồm các chương:

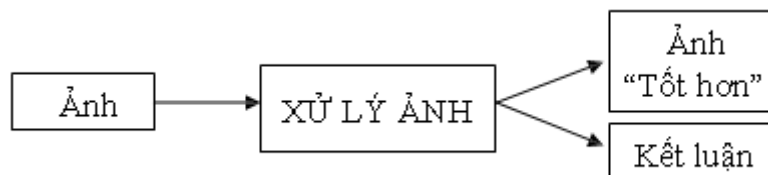
- Chương 1: Tổng quan về xử lý ảnh.
- Chương 2: Xương và các kỹ thuật tìm xương
- Chương 3: Kỹ thuật cắt tia xương của ảnh.
- Chương 4: Kết quả thực nghiệm.

CHƯƠNG 1: TỔNG QUAN VỀ XỬ LÝ ẢNH

1.1 Xử lý ảnh là gì?

Con người thu nhận thông tin qua các giác quan, trong đó thị giác đóng vai trò quan trọng nhất. Những năm trở lại đây với sự phát triển của phần cứng máy tính, xử lý ảnh và đồ họa đã phát triển một cách mạnh mẽ và có nhiều ứng dụng trong cuộc sống. Xử lý ảnh và đồ họa đóng một vai trò quan trọng trong tương tác người máy.

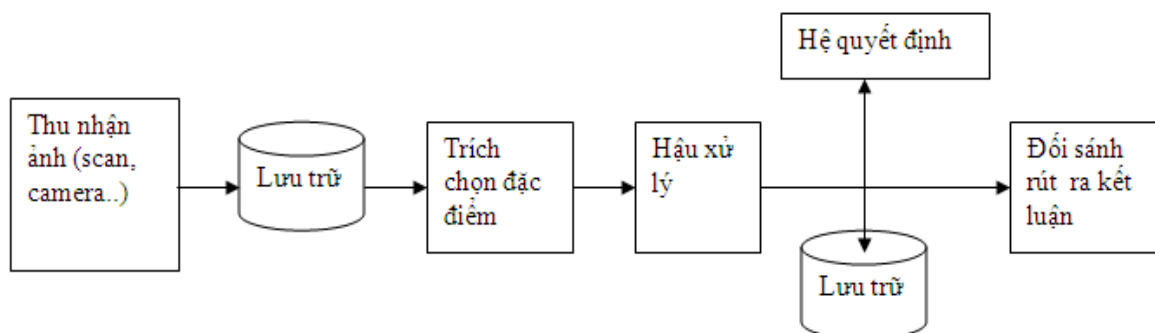
Quá trình xử lý ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.



Hình 1. 1. Quá trình xử lý ảnh

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

Sơ đồ tổng quát của một hệ thống xử lý ảnh:



Hình 1. 2. Các bước cơ bản trong một hệ thống xử lý ảnh

1.2 Các vấn đề cơ bản trong xử lý ảnh

1.2.1 Một số khái niệm cơ bản

* Ảnh và điểm ảnh:

Điểm ảnh được xem như là dấu hiệu hay cường độ sáng tại 1 tọa độ trong không gian của đối tượng và ảnh được xem như là 1 tập hợp các điểm ảnh.

* Mức xám, màu

Là số các giá trị có thể có của các điểm ảnh của ảnh

1.2.2 Thu nhận ảnh

1.2.2.1 Thu nhận, các thiết bị thu nhận ảnh

Các thiết bị thu nhận ảnh bao gồm camera, scanner các thiết bị thu nhận này có thể cho ảnh đen trắng

Các thiết bị thu nhận ảnh có 2 loại chính ứng với 2 loại ảnh thông dụng Raster, Vector.

Các thiết bị thu nhận ảnh thông thường Raster là camera các thiết bị thu nhận ảnh thông thường Vector là sensor hoặc bàn số hoá Digitalizer hoặc được chuyển đổi từ ảnh Raster.

Nhìn chung các hệ thống thu nhận ảnh thực hiện 1 quá trình

- Cảm biến: biến đổi năng lượng quang học thành năng lượng điện
- Tổng hợp năng lượng điện thành ảnh

1.2.2.2 Biểu diễn ảnh

Ảnh trên máy tính là kết quả thu nhận theo các phương pháp số hoá được nhúng trong các thiết bị kỹ thuật khác nhau. Quá trình lưu trữ ảnh nhằm 2 mục đích:

- Tiết kiệm bộ nhớ

- Giảm thời gian xử lý

Việc lưu trữ thông tin trong bộ nhớ có ảnh hưởng rất lớn đến việc hiển thị, in ấn và xử lý ảnh được xem như là 1 tập hợp các điểm với cùng kích thước nếu sử dụng càng nhiều điểm ảnh thì bức ảnh càng đẹp, càng mịn và càng thể hiện rõ hơn chi tiết của ảnh người ta gọi đặc điểm này là độ phân giải.

Việc lựa chọn độ phân giải thích hợp tùy thuộc vào nhu cầu sử dụng và đặc trưng của mỗi ảnh cụ thể, trên cơ sở đó các ảnh thường được biểu diễn theo 2 mô hình cơ bản.

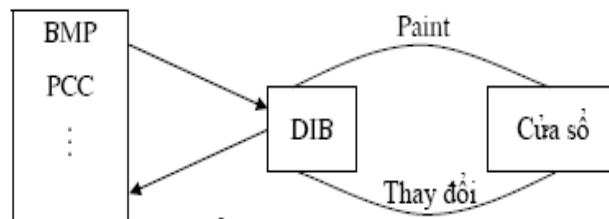
Mô hình Raster

Đây là cách biểu diễn ảnh thông dụng nhất hiện nay, ảnh được biểu diễn dưới dạng ma trận các điểm (điểm ảnh). Thường thu nhận qua các thiết bị như camera, scanner. Tùy theo yêu cầu thực tế mà mỗi điểm ảnh được biểu diễn qua 1 hay nhiều bit

Mô hình Raster thuận lợi cho hiển thị và in ấn. Ngày nay công nghệ phần cứng cung cấp những thiết bị thu nhận ảnh Raster phù hợp với tốc độ nhanh và chất lượng cao cho cả đầu vào và đầu ra. Một thuận lợi cho việc hiển thị trong môi trường Windows là Microsoft đưa ra khuôn dạng ảnh DIB (Device Independent Bitmap) làm trung gian. Hình 1. 4 thể hình quy trình chung để hiển thị ảnh Raster thông qua DIB.

Một trong những hướng nghiên cứu cơ bản trên mô hình biểu diễn này là kỹ thuật nén ảnh các kỹ thuật nén ảnh lại chia ra theo 2 khuynh hướng là nén bảo toàn và không bảo toàn thông tin nén bảo toàn có khả năng phục hồi hoàn toàn dữ liệu ban đầu còn nếu không bảo toàn chỉ có khả năng phục hồi độ sai số cho phép nào đó. Theo cách tiếp cận này người ta đã đề ra nhiều quy cách khác nhau như BMP, TIF, GIF, PCX...

Hiện nay trên thế giới có trên 50 khuôn dạng ảnh thông dụng bao gồm cả trong đó các kỹ thuật nén có khả năng phục hồi dữ liệu 100% và nén có khả năng phục hồi với độ sai số nhận được.



Hình 1. 3. Quá trình hiển thị và chỉnh sửa, lưu trữ ảnh thông qua DIB.

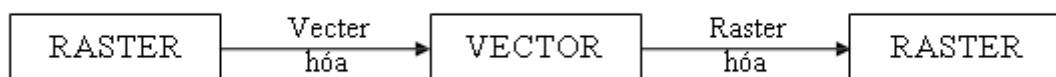
Mô hình Vector

Biểu diễn ảnh ngoài mục đích tiết kiệm không gian lưu trữ dễ dàng cho hiển thị và in ấn còn đảm bảo dễ dàng trong lựa chọn sao chép di chuyển tìm kiếm. Theo những yêu cầu này kỹ thuật biểu diễn vector tỏ ra ưu việt hơn.

Trong mô hình vector người ta sử dụng hướng giữa các vector của điểm ảnh lân cận để mã hoá và tái tạo hình ảnh ban đầu ảnh vector được thu nhận trực tiếp từ các thiết bị số hoá như Digital hoặc được chuyển đổi từ ảnh Raster thông qua các chương trình số hoá

Công nghệ phần cứng cung cấp những thiết bị xử lý với tốc độ nhanh và chất lượng cho cả đầu vào và ra nhưng lại chỉ hỗ trợ cho ảnh Raster.

Do vậy, những nghiên cứu về biểu diễn vectơ đều tập trung từ chuyển đổi từ ảnh Raster.

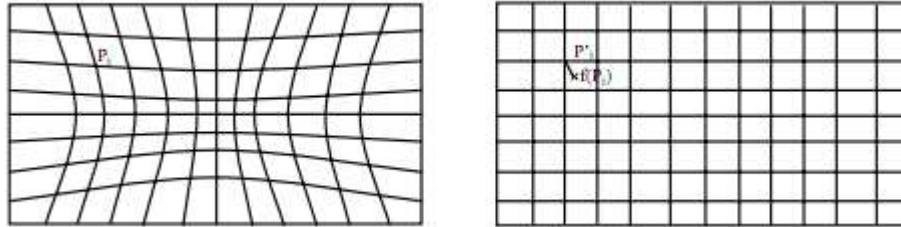


Hình 1. 4. Sự chuyển đổi giữa các mô hình biểu diễn ảnh

1.2.3 Nâng cao chất lượng ảnh

1.2.3.1 Nắn chỉnh biến dạng

Ảnh thu nhận thường bị biến dạng do các thiết bị quang học và điện tử.



Ảnh thu nhận

Ảnh mong muốn

Hình 1. 5. Ảnh thu nhận và ảnh mong muốn

Để khắc phục người ta sử dụng các phép chiếu, các phép chiếu thường được xây dựng trên tập các điểm điều khiển.

Giả sử (P_i, P'_i) $i = 1, n$ có n các tập điều khiển

Tìm hàm $f: P_i \rightarrow f(P_i)$ sao cho:

$$\sum_{i=1}^n \|f(P_i) - P'_i\|^2 \rightarrow \min$$

Giả sử ảnh bị biến đổi chỉ bao gồm: Tịnh tiến, quay, tỷ lệ, biến dạng bậc nhất tuyến tính. Khi đó hàm f có dạng:

$$f(x, y) = (a_1x + b_1y + c_1, a_2x + b_2y + c_2)$$

Ta có

$$\phi = \sum_{i=1}^n \|f(P_i) - P'_i\|^2 = \sum_{i=1}^n \left[(a_1x_1 + b_1y_1 + c_1 - x'_1)^2 + (a_2x_1 + b_2y_1 + c_2 - y'_1)^2 \right]$$

Để cho $\phi \rightarrow \min$

$$\begin{cases} \frac{\partial \phi}{\partial a_1} = 0 \\ \frac{\partial \phi}{\partial b_1} = 0 \\ \frac{\partial \phi}{\partial c_1} = 0 \end{cases} \Leftrightarrow \begin{cases} \sum_{i=1}^n a_1 x_1^2 + \sum_{i=1}^n b_1 x_1 y_1 + \sum_{i=1}^n c_1 x_1 = \sum_{i=1}^n x_1 x_1' \\ \sum_{i=1}^n a_1 x_1 y_1 + \sum_{i=1}^n b_1 y_1^2 + \sum_{i=1}^n c_1 y_1 = \sum_{i=1}^n y_1 x_1' \\ \sum_{i=1}^n a_1 x_1 + \sum_{i=1}^n b_1 y_1 + n c_1 = \sum_{i=1}^n x_1' \end{cases}$$

Giải hệ phương trình tuyến tính tìm được a_1, b_1, c_1

Tương tự tìm được a_2, b_2, c_2

→ Xác định được hàm f

1.2.3.2 Khử nhiễu

Có 2 loại nhiễu cơ bản trong quá trình thu nhận ảnh:

- Nhiễu hệ thống: là nhiễu có quy luật có thể khử bằng các phép biến đổi
- Nhiễu ngẫu nhiên: vết bản không rõ nguyên nhân → khắc phục bằng các phép lọc

1.2.3.3 Chỉnh mức xám

Nhằm khắc phục tính không đồng đều của hệ thống gây ra. Thông thường có 2 hướng tiếp cận:

- Giảm số mức xám: Thực hiện bằng cách nhóm các mức xám gần nhau thành một bó. Trường hợp chỉ có 2 mức xám thì chính là chuyển về ảnh đen trắng. Ứng dụng: In ảnh màu ra máy in đen trắng.
- Tăng số mức xám: Thực hiện nội suy ra các mức xám trung gian bằng kỹ thuật nội suy. Kỹ thuật này nhằm tăng cường độ mịn cho ảnh

1.2.4 Trích chọn đặc điểm

Các đặc điểm của đối tượng được trích chọn tùy theo mục đích nhận dạng trong quá trình xử lý ảnh. Có thể nêu ra một số đặc điểm của ảnh sau đây:

Đặc điểm không gian: Phân bố mức xám, phân bố xác suất, biên độ, điểm uốn v. v. .

Đặc điểm biến đổi: Các đặc điểm loại này được trích chọn bằng việc thực hiện lọc vùng (zonal filtering). Các bộ vùng được gọi là “mặt nạ đặc điểm” (feature mask) thường là các khe hẹp với hình dạng khác nhau (chữ nhật, tam giác, cung tròn v. v. .)

Đặc điểm biên và đường biên: Đặc trưng cho đường biên của đối tượng và do vậy rất hữu ích trong việc trích chọn các thuộc tính bất biến được dùng khi nhận dạng đối tượng. Các đặc điểm này có thể được trích chọn nhờ toán tử gradient, toán tử la bàn, toán tử Laplace, toán tử “chéo không” (zero crossing) v. v. .

Việc trích chọn hiệu quả các đặc điểm giúp cho việc nhận dạng các đối tượng ảnh chính xác, với tốc độ tính toán cao và dung lượng nhớ lưu trữ giảm xuống.

1.2.5 Nhận dạng

Nhận dạng tự động (automatic recognition), mô tả đối tượng, phân loại và phân nhóm các mẫu là những vấn đề quan trọng trong thị giác máy, được ứng dụng trong nhiều ngành khoa học khác nhau. Tuy nhiên, một câu hỏi đặt ra là: mẫu (pattern) là gì? Watanabe, một trong những người đi đầu trong lĩnh vực này đã định nghĩa: “Ngược lại với hỗn loạn (chaos), mẫu là một thực thể (entity), được xác định một cách ang áng (vaguely defined) và có thể gán cho nó một tên gọi nào đó”. Ví dụ mẫu có thể là ảnh của vân tay, ảnh của một vật nào đó được chụp, một chữ viết, khuôn mặt người hoặc một ký đồ tín hiệu tiếng nói. Khi biết một mẫu nào đó, để nhận dạng hoặc phân loại mẫu đó có thể:

Hoặc phân loại có mẫu (supervised classification), chẳng hạn phân tích phân biệt (discriminant analysis), trong đó mẫu đầu vào được định danh như một thành phần của một lớp đã xác định.

Hoặc phân loại không có mẫu (unsupervised classification hay clustering) trong đó các mẫu được gán vào các lớp khác nhau dựa trên một tiêu chuẩn đồng dạng nào đó. Các lớp này cho đến thời điểm phân loại vẫn chưa biết hay chưa được định danh.

Hệ thống nhận dạng tự động bao gồm ba khâu tương ứng với ba giai đoạn chủ yếu sau đây:

- Thu nhận dữ liệu và tiền xử lý.
- Biểu diễn dữ liệu.
- Nhận dạng, ra quyết định.

Bốn cách tiếp cận khác nhau trong lý thuyết nhận dạng là:

- Đối sánh mẫu dựa trên các đặc trưng được trích chọn.
- Phân loại thống kê.
- Đối sánh cấu trúc.
- Phân loại dựa trên mạng nơ-ron nhân tạo.

Trong các ứng dụng rõ ràng là không thể chỉ dùng có một cách tiếp cận đơn lẻ để phân loại “tối ưu” do vậy cần sử dụng cùng một lúc nhiều phương pháp và cách tiếp cận khác nhau. Do vậy, các phương thức phân loại tổ hợp hay được sử dụng khi nhận dạng và nay đã có những kết quả có triển vọng dựa trên thiết kế các hệ thống lai (hybrid system) bao gồm nhiều mô hình kết hợp.

Việc giải quyết bài toán nhận dạng trong những ứng dụng mới, nảy sinh trong cuộc sống không chỉ tạo ra những thách thức về thuật giải, mà còn đặt ra những yêu cầu về tốc độ tính toán. Đặc điểm chung của tất cả những ứng dụng đó là những đặc điểm đặc trưng cần thiết thường là nhiều, không thể do chuyên gia đề xuất, mà phải được trích chọn dựa trên các thủ tục phân tích dữ liệu.

1.2.6 Nén ảnh

Nhằm giảm thiểu không gian lưu trữ. Thường được tiến hành theo cả hai cách khuynh hướng là nén có bảo toàn và không bảo toàn thông tin. Nén không bảo toàn thì thường có khả năng nén cao hơn nhưng khả năng phục hồi thì kém hơn. Trên cơ sở hai khuynh hướng, có 4 cách tiếp cận cơ bản trong nén ảnh:

- Nén ảnh thống kê: Kỹ thuật nén này dựa vào việc thống kê tần suất xuất hiện của giá trị các điểm ảnh, trên cơ sở đó mà có chiến lược mã hóa thích hợp. Một ví dụ điển hình cho kỹ thuật mã hóa này là *. TIF
- Nén ảnh không gian: Kỹ thuật này dựa vào vị trí không gian của các điểm ảnh để tiến hành mã hóa. Kỹ thuật lợi dụng sự giống nhau của các điểm ảnh trong các vùng gần nhau. Ví dụ cho kỹ thuật này là mã nén *. PCX
- Nén ảnh sử dụng phép biến đổi: Đây là kỹ thuật tiếp cận theo hướng nén không bảo toàn và do vậy, kỹ thuật thường nén hiệu quả hơn. *. JPG chính là tiếp cận theo kỹ thuật nén này.
- Nén ảnh Fractal: Sử dụng tính chất Fractal của các đối tượng ảnh, thể hiện sự lặp lại của các chi tiết. Kỹ thuật nén sẽ tính toán để chỉ cần lưu trữ phần gốc ảnh và quy luật sinh ra ảnh theo nguyên lý Fractal.

CHƯƠNG 2: XƯƠNG VÀ CÁC KỸ THUẬT TÌM XƯƠNG

2.1 Giới thiệu

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

Một định nghĩa xúc tích về xương dựa trên tính continuum (tương tự như hiện tượng cháy đồng cỏ) được đưa ra bởi Blum (1976) như sau: Giả thiết rằng đối tượng là đồng nhất được phủ bởi cỏ khô và sau đó dựng lên một vòng biên lửa. Xương được định nghĩa như nơi gặp của các vết lửa và tại đó chúng được dập tắt.



(a) Ảnh gốc

(b) Ảnh xương

Hình 2. 1. Ví dụ về ảnh và xương

Kỹ thuật tìm xương luôn là chủ đề nghiên cứu trong xử lý ảnh những năm gần đây. Mặc dù có những nỗ lực cho việc phát triển các thuật toán tìm xương, nhưng các phương pháp được đưa ra đều bị mất mát thông tin. Có thể chia thành hai loại thuật toán tìm xương cơ bản:

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

2.2 Tìm xương dựa trên làm mảnh ảnh

2.2.1 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 vector hoá 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 của đối tượng sẽ được kiểm tra: nếu như chúng thoả mãn điều kiện xoá nào đó tùy thuộc vào mỗi thuật toán thì nó sẽ bị xoá đi. Quá trình cứ lặp lại cho đến khi không còn điểm biên nào được xoá. Đối tượng được bóc dần lớp biên cho đến khi nào bị thu mảnh lại chỉ còn các điểm biên.

Các thuật toán làm mảnh được phân loại dựa trên phương pháp xử lý các điểm là thuật toán làm mảnh song song và thuật toán làm mảnh tuần tự.

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 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 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 thoả 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 Một số thuật toán làm mảnh

Trong phần này điếm qua một số đặc điếm, ưu và khuyết điếm của các thuật toán đã được nghiên cứu.

- Thuật toán làm mảnh cổ điển là thuật toán song song, tạo ra xương 8 liên thông, tuy nhiên nó rất chậm, gây đứt nét, xoá hoàn toàn một số cấu hình nhỏ.
- Thuật toán làm mảnh của Toumazet bảo toàn tất cả các điếm cực không gây đứt nét đối tượng. Tuy nhiên, thuật toán có nhược điếm là rất chậm, rất nhạy cảm với nhiễu, xương chỉ là 4-liên thông và không làm mảnh được với một số cấu hình phức tạp
- Thuật toán làm mảnh của Y. Xia dựa trên đường biên của đối tượng, có thể cài đặt theo cả phương pháp song song và tuần tự. Tốc độ của thuật

toán rất nhanh. Nó có nhược điểm là gây đứt nét, xương tạo ra là xương giả (có độ dày là 2 phần tử ảnh).

- Thuật toán làm mảnh của N. J. Naccache và R. Shinghal. Thuật toán có ưu điểm là nhanh, xương tạo ra có khả năng khôi phục ảnh ban đầu của đối tượng. Nhược điểm chính của thuật toán là rất nhạy với nhiễu, xương nhận được phản ánh cấu trúc của đối tượng thấp.
- Thuật toán làm mảnh của H. E. Lu P. S. P Wang tương đối nhanh, giữ được tính liên thông của ảnh, nhưng lại có nhược điểm là xương tạo ra là xương 4-liên thông và xoá mất một số cấu hình nhỏ.
- Thuật toán làm mảnh của P. S. P Wang và Y. Y. Zhang dựa trên đường biên của đối tượng, có thể cài đặt theo phương pháp song song hoặc tuần tự, xương là 8-liên thông, ít chịu ảnh hưởng của nhiễu. Nhược điểm chính của thuật toán là tốc độ chậm.
- Thuật toán làm mảnh song song thuần tuý nhanh nhất trong các thuật toán trên, bảo toàn tính liên thông, ít chịu ảnh hưởng của nhiễu. Nhược điểm là xoá hoàn toàn một số cấu hình nhỏ, xương tạo ra là xương 4-liên thông.

2.3 Tìm xương không dựa trên làm mảnh ảnh

Để tách được xương của đối tượng có thể sử dụng đường biên của đối tượng. Với điểm p bất kỳ trên đối tượng, ta bao nó bởi một đường biên. Nếu như có nhiều điểm biên có cùng khoảng cách ngắn nhất tới p thì p nằm trên trục trung vị. Tập 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 ảnh đã được tính toán và các điểm ảnh có giá trị lớn nhất được xem là nằm trên xương của đối tượng.

2.3.1 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 hai phần tử của tập Ω gồm n điểm trong mặt phẳng. Tập các điểm trong mặt phẳng gần P_i hơn P_j là nửa mặt phẳng $H(P_i, P_j)$ chứa điểm P_i và bị giới hạn bởi đường trung trực của đoạn thẳng $P_i P_j$. Do đó, tập các điểm gần P_i 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 Ω là hợp của tất cả các $V(P_i)$

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

Định nghĩa 2.2 [Đa giác Voronoi tổng quát]

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

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

2.3.2 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 $map(x, y)$ với hàm khoảng cách $d(\cdot, \cdot)$ như sau:

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

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 $\text{DM}(S)$, được gọi là 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 $\text{EDM}(S)$ 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:

$$\hat{B}(S) = \{B_1(S), \dots, B_n(S)\} \quad (2. 6)$$

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 :

$$\text{DVMA}(\hat{B}(S)) = \text{Vor}(\hat{B}(S)) \cap S \quad (2. 7)$$

2.3.3 Xương Voronoi rời rạc

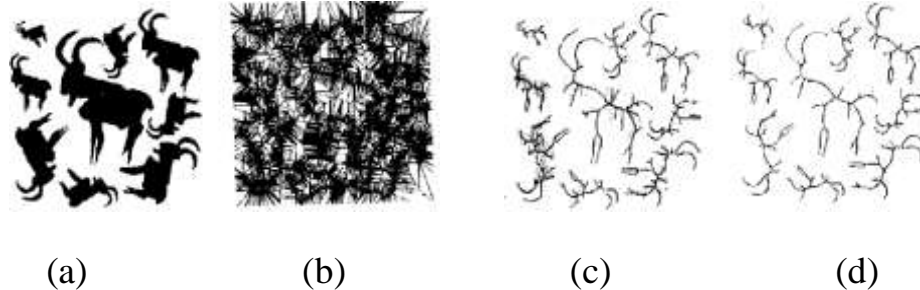
Định nghĩa 2. 6 [Xương Voronoi rời rạc - DiscreteVoronoi Skeleton]

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

$$\text{SkeDVMA}(\hat{B}(S), T) = \{(x, y) \mid (x, y) \in \text{DVMA}(\hat{B}(S)), \Psi(x, y) > T\} \quad (2. 8)$$

Ψ : là hàm hiệu chỉnh.

Để thấy nếu ngưỡng T càng lớn thì càng thì số lượng điểm tham gia trong xương Voronoi càng ít (Hình 2. 2).



Hình 2. 2. Xương Voronoi rời rạc.

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

- (a) Ảnh nhị phân.
- (b) Sơ đồ Voronoi.
- (c) Hiệu chỉnh bởi hàm Potential, $T=9.0$.
- (d) Hiệu chỉnh bởi hàm Potential, $T=18.0$.

2.3.4 Thuật toán tìm xương

Trong mục này sẽ trình bày ý tưởng cơ bản của thuật toán tìm xương và mô tả bằng ngôn ngữ tựa Pascal.

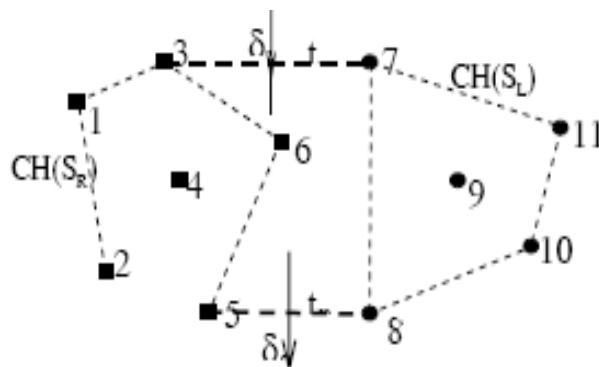
Tăng trưởng: Việc tính toán sơ đồ Voronoi được bắt đầu từ một điểm sinh trong mặt phẳng. Sau đó điểm sinh thứ hai được thêm vào và quá trình tính toán tiếp tục với đa giác Voronoi đã tìm được với điểm vừa được thêm vào đó. Cứ như thế, quá trình tính toán sơ đồ Voronoi được thực hiện cho đến khi không còn điểm sinh nào được thêm vào. Nhược điểm của chiến lược này là mỗi khi một 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 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 hai ý tưởng ở trên. Tuy nhiên, nó sẽ mang 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 hai phần (bên trái: 1-6, bên phải: 7-11) bởi đường gấp khúc δ , 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 δ sẽ là một bộ phận của đường trung trực nối hai điểm mà một điểm thuộc $Vor(S_L)$ và một điểm thuộc $Vor(S_R)$. Trước khi xây dựng δ , ta tìm ra phần tử đầu và cuối của nó. Nhìn vào hình trên, ta nhận thấy rằng cạnh δ_1 và δ_5 là các tia. Để nhận thấy rằng việc tìm ra các cạnh đầu và cuối của δ trở thành việc tìm cạnh vào t_α và cạnh ra t_ω .

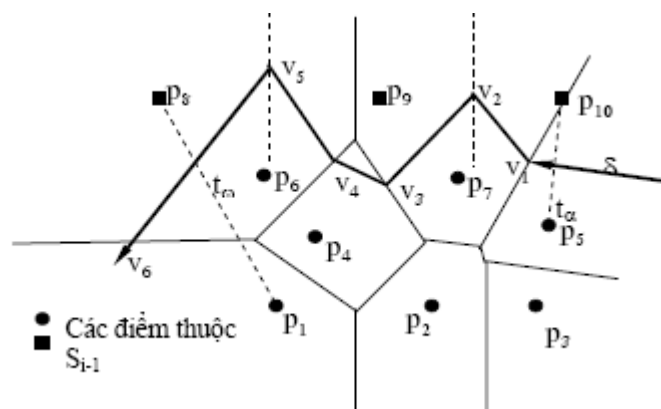


Hình 2. 3. Minh họa thuật toán trộn hai sơ đồ Voronoi

Sau khi đã tìm được t_α và t_ω , các điểm cuối của t_α được sử dụng để xây dựng phần tử đầu tiên của δ (δ_1 trong hình trên). Sau đó thuật toán tìm điểm

giao của δ với $\text{Vor}(S_L)$ và $\text{Vor}(S_R)$. Trong ví dụ trên, δ đầu tiên giao với $V(3)$. Kể từ đây, các điểm nằm trên phần kéo dài δ sẽ gần điểm 6 hơn điểm 3. Do đó, phần tử tiếp theo δ_2 của δ sẽ thuộc vào đường trung trực của điểm 6 và điểm 7. Sau đó điểm giao tiếp theo của δ sẽ thuộc về $\text{Vor}(S_L)$; δ bây giờ sẽ đi vào $V(9)$ và δ_2 sẽ được thay thế bởi δ_3 . Quá trình này sẽ kết thúc khi δ gặp phần tử cuối δ_5 .

Trên đây chỉ là minh họa cho thuật 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 vào ngay 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 sơ đồ Voronoi $\text{Vor}(L_i)$ 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, $0 \leq i \leq i_{MAX}$, $Vor(S_i)$ sơ đồ Vorronoi 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 ;

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

End

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

Begin

Increment i ; GetScanLine L_i ;

$Vor(L_i) :=$ các đường trung trực sinh bởi các điểm sinh thuộc L_i

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

End

End.

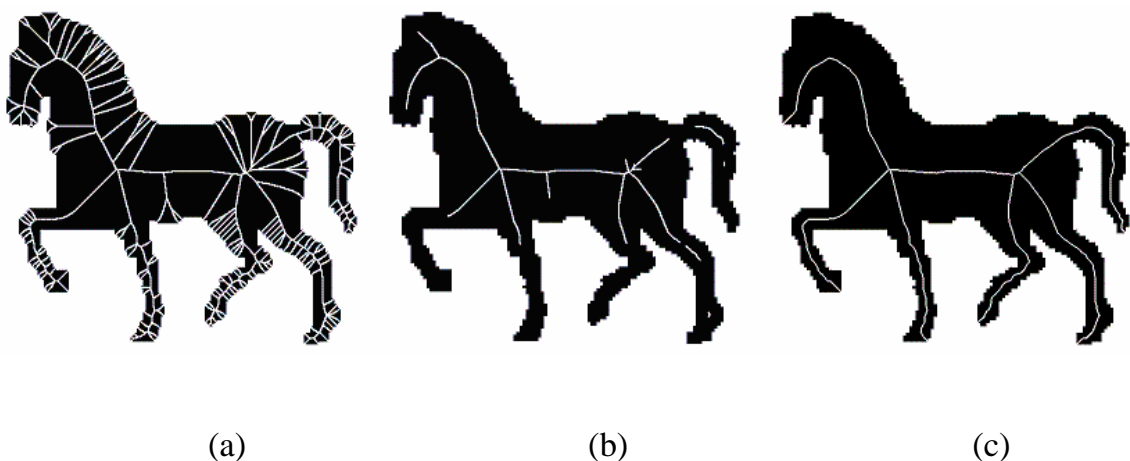
Giả sử xét trên hệ tọa độ thực. Ảnh vào được quét từ dưới lên. Tọa độ y (biến i) tương ứng với từng dòng quét được tăng dần theo từng dòng. Trong thủ tục trên, hàm quan trọng nhất là hàm VoroLink, hàm này thực hiện việc trộn sơ đồ Voronoi của L_{i-1} dòng đã được quét trước đó với sơ đồ Voronoi của

dòng hiện tại thứ i . Trong vòng lặp trên, hàm VoroPreScan là một biến thể của hàm 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 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: KỸ THUẬT CẮT TỈA XƯƠNG CỦA ẢNH

3.1 Giới thiệu

Xương được coi như hình dạng cơ bản của một đối tượng, với số ít các điểm ảnh cơ bản. Ta có thể lấy được các thông tin về hình dạng nguyên bản của một đối tượng thông qua xương. Vì thế nó được áp dụng rộng rãi trong các ngành như đồ họa máy tính, tra cứu ảnh, nhận dạng ký tự [1]. Do tầm quan trọng của xương rất nhiều thuật toán phát hiện xương được đề xuất. Nhiều nhà nghiên cứu đã có nhiều cố gắng để nhận ra hình dạng cơ bản của cấu trúc xương được biểu diễn bởi đồ thị hoặc cây như [2], [3], [4], [29], [30], [31], [36]. Tuy nhiên những phương pháp trên chỉ có thể ứng dụng cho các đối tượng có hình dạng đơn giản và do đó không thể áp dụng cho đối tượng có hình dạng phức tạp hơn giống như trong tập dữ liệu MPEG-7 [37]. Yếu tố quan trọng nhất của xương là sự nhạy cảm của xương tới đường biên của đối tượng, một ít nhiễu hoặc vài sự thay đổi của đường biên dẫn đến tạo ra những nhánh xương thừa cái mà có thể làm ảnh hưởng nghiêm trọng tới hình dạng cơ bản của đồ thị xương. Ví dụ như xương trong hình 3.1(a) có nhiều nhánh xương thừa được phát sinh ra bởi nhiễu đường biên.



Hình 3.1. Minh họa xương của ảnh.

Bộ xương trong (a) có nhiều nhánh thừa, để loại bỏ chúng phương pháp cắt tỉa xương được áp dụng. Hình (b) minh họa bài toán cắt tỉa thực tế (nó

được tạo ra bởi một phương pháp trong [7]). Đặc biệt, quan sát thấy cắt tia xương có thể làm thay đổi hình học cơ bản của xương ban đầu. Hình (c) minh họa kết quả từ phương pháp đề xuất là đảm bảo được dạng hình học nguyên bản của xương ban đầu.

Các thuật toán phát hiện xương có thể phân thành 4 loại:

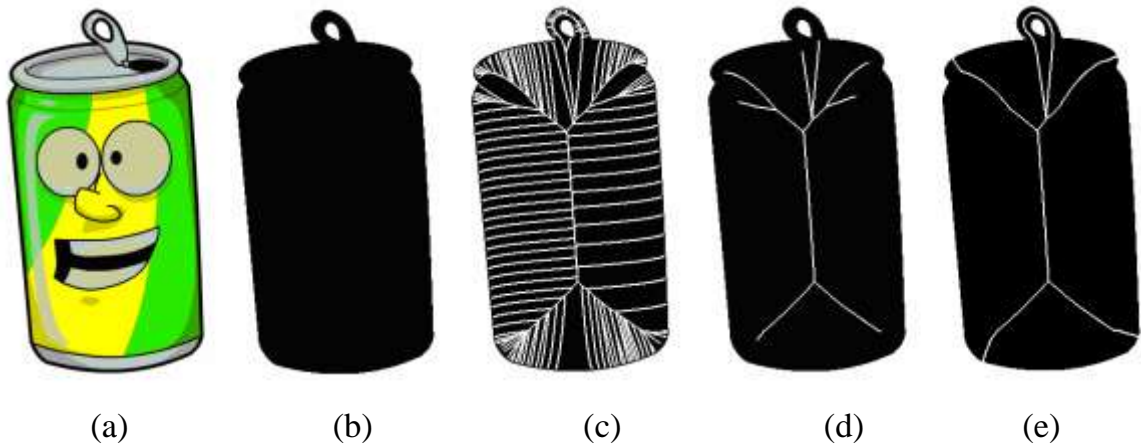
- Thuật toán làm mảnh ảnh [8], [9], [10], [34].
- Giải thuật miền rời rạc dựa trên lược đồ Voronoi [5], [12], [27], [28].
- Tìm đỉnh trong bản đồ khoảng cách của các điểm biên [7], [10], [11], [13], [19], [33], [35].
- Dựa trên phép toán hình thái học [22], [24], [25], [26].

Có 2 phương pháp cắt tia xương chính.

- Dựa trên phương pháp đo lường tới các điểm xương [5], [6], [7], [20], [28].
- Dựa trên làm mịn đường biên trước khi phát hiện xương [20], [38], [39]. Khi làm mịn vẫn có một vài vấn đề quan trọng cần lưu ý đó là những thông tin về đường biên của đối tượng có thể bị thay đổi khi có nhiễu [20].

Những phương pháp cắt tia xương được giới thiệu ở trên có một vài hạn chế sau đây:

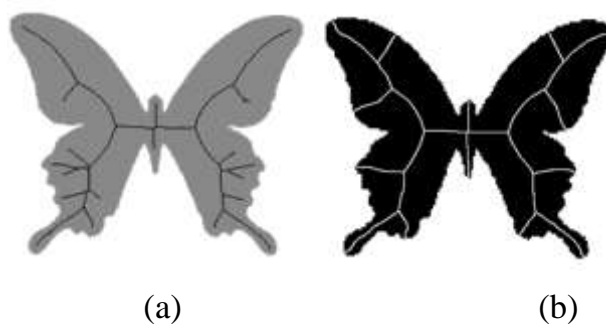
Hạn chế đầu tiên là nhiều phương pháp không đảm bảo được hình học cơ bản của vật thể đối với những hình dạng phức tạp (ví dụ một hình có lỗ). Điều này được minh họa trong hình 3.2, xương thu được bởi phương pháp [7] (d) vi phạm hình học của xương đầu vào (c).



Hình 3.2. Minh họa hạn chế 1.

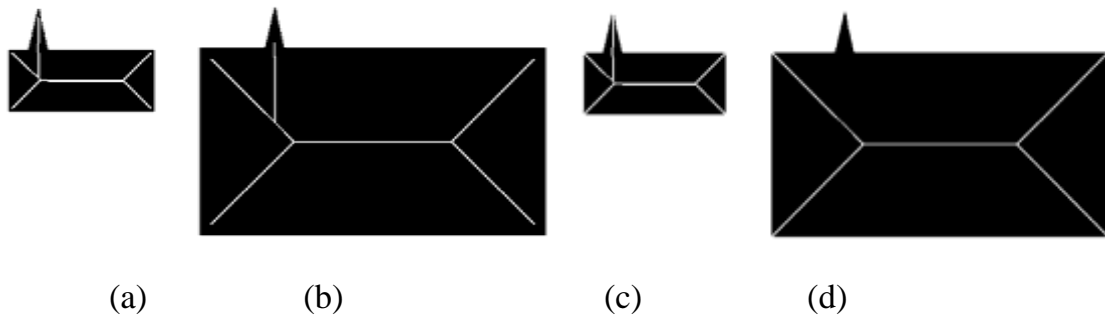
- (a) Đôi tượng đầu vào.
- (b) Mặt nạ đôi tượng nhị phân.
- (c) Bộ xương ban đầu.
- (d) Cắt tia xương của đôi tượng bằng phương pháp [7].
- (e) Cắt tia xương theo phương pháp đề xuất.

Hạn chế thứ 2 là những nhánh chính của bộ xương bị ngắt 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 3.3. So sánh kết quả của [7] (a) và của phương pháp đề xuất (b).

Hạn chế thứ 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. Điều này được minh họa trong hình 3.4.



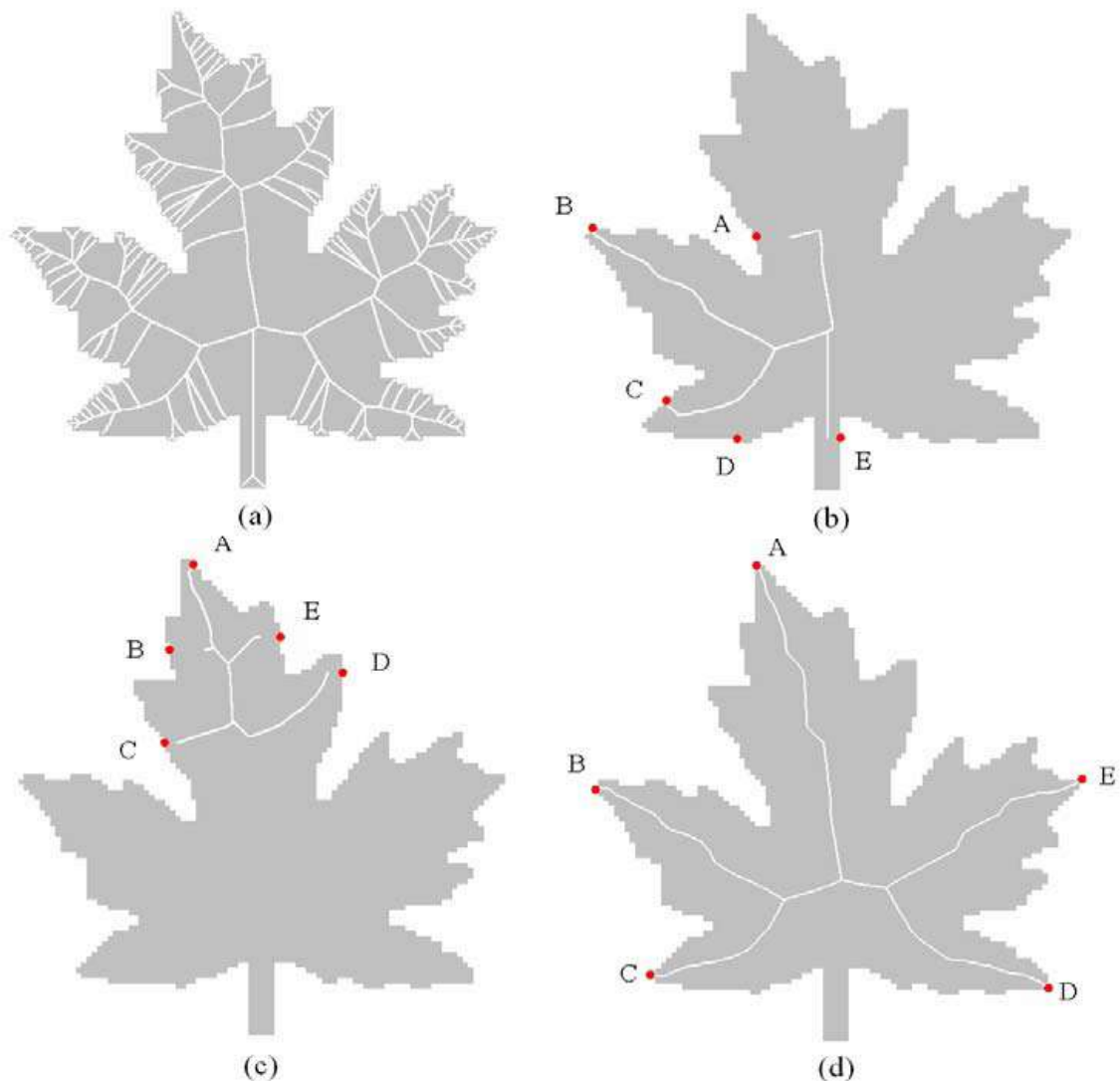
Hình 3.4. Minh họa hạn chế 3.

Với cùng một nguồn đầu vào kết quả của phương pháp [7] (b) đưa ra không chú ý hình dạng toàn cục của vật thể do nhiều. 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 trong [40].

3.2 Ý tưởng chính của phương pháp

Xiang Bai, Longin Jan Latecki, Wen-Yu Liu, đề xuất một phương pháp hoàn toàn loại bỏ 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 mà phương pháp đề xuất không bao gồm các 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 của phương pháp là di chuyển tất cả điểm xương của điểm tăng trưởng nằm trên cùng đoạn đường biên. Công việc này được thực hiện cho bất kì phần nào trong đoạn đường biên nhưng một vài phần cho kết quả tốt hơn những phần khác. Hình 5 minh họa 3 phương pháp cắt tia xương khác nhau (b, c, d) với cùng xương đầu vào là (a).



Hình 3.5. Cắt tia xương với phân chia đường biên.

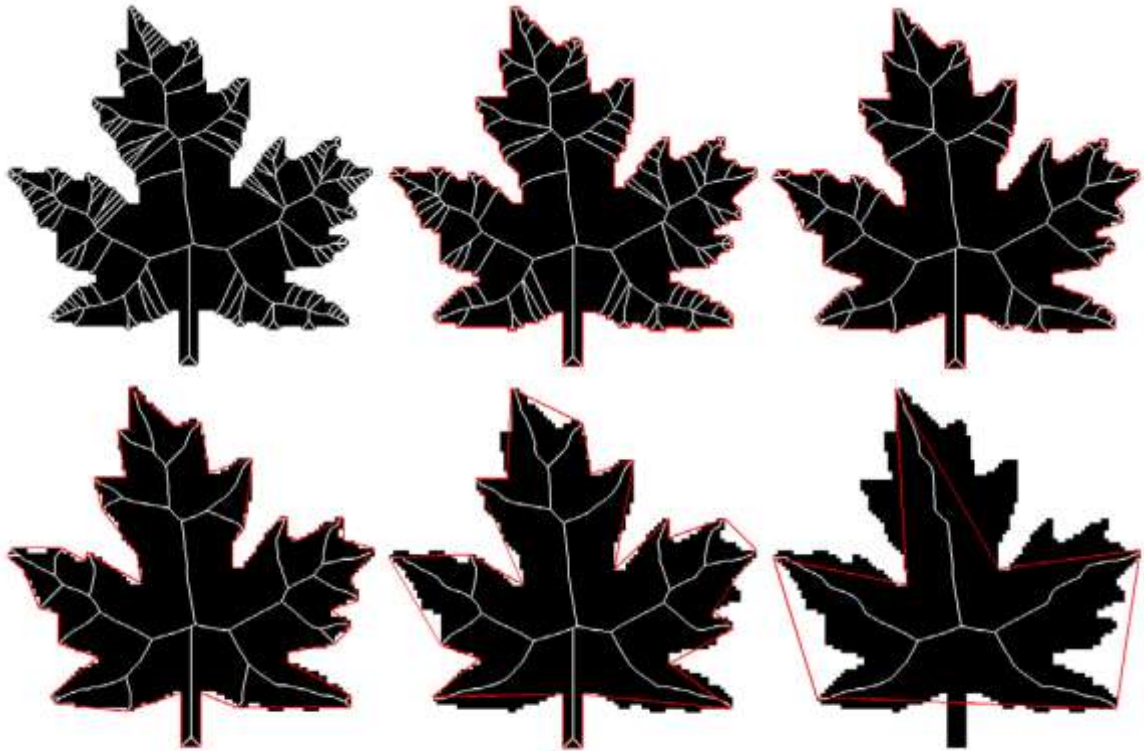
Cắt tia xương (a) với sự chú ý phân chia đường biên gây ra bởi 5 điểm ngẫu nhiên trên đường biên trong (b) và (c). 5 điểm trong (d) được lựa chọn bởi DCE.

Xương cắt tia dựa trên 3 phương pháp phân chia đường biên khác nhau với điểm kết thúc được đánh dấu bằng dấu chấm. Ví dụ loại bỏ tất cả điểm xương của điểm tăng trưởng trong đoạn đường 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) và (c). Từ đó đặt ra câu hỏi là làm thế nào để tìm ra các đoạn phân chia đường biên tốt nhất. Tác giả có được sự

phân chia như vậy nhờ quá trình DCE (Discrete Curve Evolution) [15], [16], [17] 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. 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. Khi ấy tồn tại một tập hợp con của các điểm lấy mẫu nằm trên đường biên của đối tượng liên tục. Số điểm như vậy phụ thuộc vào độ lệch chuẩn của lỗi lấy mẫu. Câu hỏi đặt ra là làm thế nào để xác định các điểm nằm trên đường biên của đối tượng gốc (hoặc rất gần) hoặc các điểm nhiễu (nằm xa đường biên gốc). Quá trình DCE được chứng minh bằng thực nghiệm và lý thuyết để loại bỏ các điểm nhiễu [15], [16], [17]. Quá trình này 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 xác định bởi những đỉnh liên tục của đa giác đơn giản.

Một cấu trúc xương tuần tự được khắc phục bằng phương pháp đề xuất minh họa trong hình 3.6, nơi mà các đường biên đa giác (màu đỏ) được đơn giản bởi DCE. Vì DCE có thể làm giảm điểm biên nhiễu mà không thay thế các điểm biên chính nên tính chính xác của xương được bảo đảm. Tính liên tục trong đó hàm ý sự ổn định trong sự hiện diện của nhiễu của phương pháp cắt tia đề xuất theo tính liên tục của DCE. Điều này có nghĩa rằng nếu một đường biên cho trước và các bản nhiễu của nó được đóng (đo bằng khoảng cách Hausdorff), các bộ xương cắt tia thu được cũng sẽ được đóng. Một bằng chứng về tính liên tục của DCE đối với khoảng cách Hausdorff của các đường cong đa giác được đưa ra trong [23].



Hình 3.6. Trình tự bộ xương của lá.

Trình tự bộ xương của lá thu được bằng cách cắt tỉa bộ xương đầu vào (phía trên bên trái) với sự chú ý đoạn đường biên thu được bởi DCE. Đường nét bên ngoài (màu đỏ) thể hiện đơn giản hóa đường biên với DCE.

Phương pháp cắt tỉa xương có thể được áp dụng với bất kỳ bộ xương đầu vào nào. Mỗi điểm xương là trung tâm của vòng tròn lớn nhất và những điểm đường biên tiếp tuyến với đường tròn đều được đưa ra. Cắt tỉa xương không phải thực hiện sau khi đã tính được bộ xương mà được thực hiện đồng thời với quá trình tăng trưởng xương. Để thực hiện ý tưởng này tác giả mở rộng thuật toán phát triển xương trong [7] dựa trên độ đo khoảng cách Eculidean. Trước tiên tác giả chọn một điểm hạt giống xương như là một điểm lớn nhất của khoảng cách Euclidean. Những điểm xương mới được thêm vào bằng cách kiểm tra liên thông 8, trong quá trình này những nhánh xương thừa được loại bỏ bởi DCE.

3.3 Cắt tia xương với DCE

3.3.1 Rời rạc hóa đường cong

DCE được giới thiệu trong [16], [17], [18]. Đường biên của đối tượng trong ảnh số bị thay đổi bởi nhiễu và các lỗi phân đoạn. DCE loại bỏ những thay đổi đó trong khi vẫn đảm bảo được hình dạng ban đầu của vật thể bởi đơn giản hóa hình dạng. Bất kỳ đường cong của ảnh số có thể được coi là một đa giác mà không bị mất thông tin, nhưng phải có số đỉnh lớn để nghiên cứu sự phát triển của hình dạng. Ý tưởng cơ bản của sự phát triển đề xuất là các đa giác đều đơn giản.

Trong mỗi bước tiến hóa một đoạn liên tiếp s_1, s_2 được thay thế bởi đoạn nối điểm cuối $s_1 U s_2$.

Phần chính của sự tiến hóa này là sự thay thế. Sự thay thế được thực hiện theo phép đo liên quan K đưa ra bởi:

$$K_{s_1, s_2} = \frac{\beta_{s_1, s_2} \cdot l_{s_1} \cdot l_{s_2}}{l_{s_1} + l_{s_2}}$$

s_1, s_2 là những cạnh đa giác liên quan tới đỉnh v , $\beta(s_1, s_2)$ góc quay tại đỉnh chung của đoạn s_1, s_2 , l độ dài bình thường với sự chú ý tổng độ dài của đường cong đa giác C .

Đầu vào là đường biên đa giác P với n đỉnh, DCE tạo ra một chuỗi các đa giác đơn giản $P=P^n, P^{n-1}, \dots, P^3$ như vậy $P^{n-(k+1)}$ thu được bằng cách loại bỏ đỉnh v từ P^{n-k} với K là nhỏ nhất.

Định nghĩa 1. Một tính chất quan trọng của DCE là phân chia trình tự với đa giác đầu vào P . $\{v_1, \dots, v_n\}$ là đỉnh của P , $\{u_1, \dots, u_m\} \subset \{v_1, \dots, v_n\}$ là đỉnh lỗi của P^{n-k} sao cho $m \leq n-k$. Trên cấp $n-k$ của phân chia hệ thống H_n .

$k(P)$, P bị phân thành các cung nhỏ m của P : $H_{n-k}(P) = \{[u_1, u_2], [u_2, u_3], \dots, [u_m, u_1]\}$.

Nếu đỉnh u_i được xoá trong bước tiến hóa tiếp theo, (ví dụ, $u_i \in P^{n-k} - P^{n-k+1}$), hoặc trở thành lõm (để xoá một trong những đỉnh ở bên cạnh), sau đó cung $[u_{i-1}, u_{i+1}]$ thay thế cung $[u_{i-1}, u_i]$, $[u_i, u_{i+1}]$ trong mức chia $H_{n-(k+1)}(P)$.

Nhận thấy DCE và phân chia trình tự có thể được định nghĩa cho một tập hữu hạn của đường cong đa giác. Trong mỗi bước DCE một vectơ đơn được loại bỏ từ một đa giác mà phép đo liên quan là nhỏ nhất. Phương pháp cắt tia đề xuất có thể được áp dụng cho mặt phẳng D , với đường biên ∂D bao gồm đa số các đa giác đóng đơn giản.

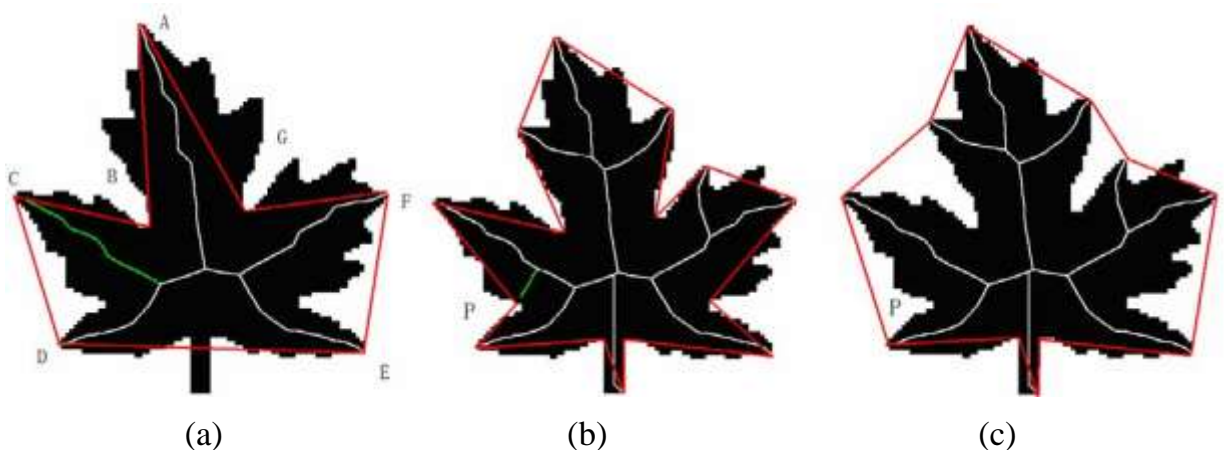
DCE có thể loại bỏ hiệu quả nhiều và từng phần không quan trọng của ảnh, nhưng một tham số dừng đúng cách là vẫn cần thiết. Nói cách khác, tìm kiếm k để đa giác đơn giản P^{n-k} miêu tả chi tiết những đường biên đầu vào. Để định lượng mức độ chi tiết, tác giả xác định khoảng cách trung bình P^{n-k} giữa điểm gốc của P và những đoạn dòng tương ứng của nó trong P^{n-k} .

Đưa ra giới hạn T , có thể dừng DCE nếu $D_{av}(P^{n-k}) > T$ cho một vài k . Cho một chuỗi các giá trị T , chúng ta có thể có được một trình tự của đơn giản hóa đường biên đa giác DCE, dẫn đến trình tự của những xương tương ứng. Nói chung, điều kiện dừng thích hợp phụ thuộc vào ứng dụng cụ thể. Một điều kiện dừng thích hợp cho tương tự hình dáng của DCE được đưa ra trong [18].

3.3.2 Cắt tia xương với DCE

Cho một bộ xương $S(D)$ của một mặt phẳng D và đưa ra một DCE đa giác đơn giản P^k , thể hiện cắt tia xương bằng cách di chuyển tất cả những điểm $s \in S(D)$, như vậy tạo ra những điểm tăng trưởng $\tan(s)$ của s chứa trong

cùng đoạn DCE mở. Mỗi điểm cắt tia s là kết quả từ phần đường biên cục bộ với sự chú ý phân chia DCE, và do đó, s có thể được coi như là điểm xương không quan trọng và có thể loại bỏ. Quá trình làm đơn giản đường biên với DCE đã hoàn thành cắt tia nhánh của xương. Đặc biệt, loại bỏ một đỉnh lồi v từ P^{n-k} tới $P^{n-(k+1)}$ bởi DCE, tức là hoàn thành loại bỏ những nhánh xương mà kết thúc tại v . Trong hình 3.5 minh họa việc sử dụng DCE thu được một hình đa giác với 7 đỉnh và xương của đối tượng được cắt tia dựa trên đa giác đó. Chỉ có 5 nhánh xương kết thúc tại 5 đỉnh lồi của đa giác đơn giản. Việc cắt tia xương được tính toán dựa trên sự chú ý đoạn DCE (A, C), (C, D), (D, E), (E, F), (F, A). Trong hình 3.5(a) nhánh xương màu xanh kết thúc tại C còn lại bởi vì nó tiếp tuyến tới hình tròn lớn nhất trên hai đoạn DCE khác nhau đó là 2 cung đường biên (B, C) và (C, D).



Hình 3.7 Minh họa cắt tia xương với DCE

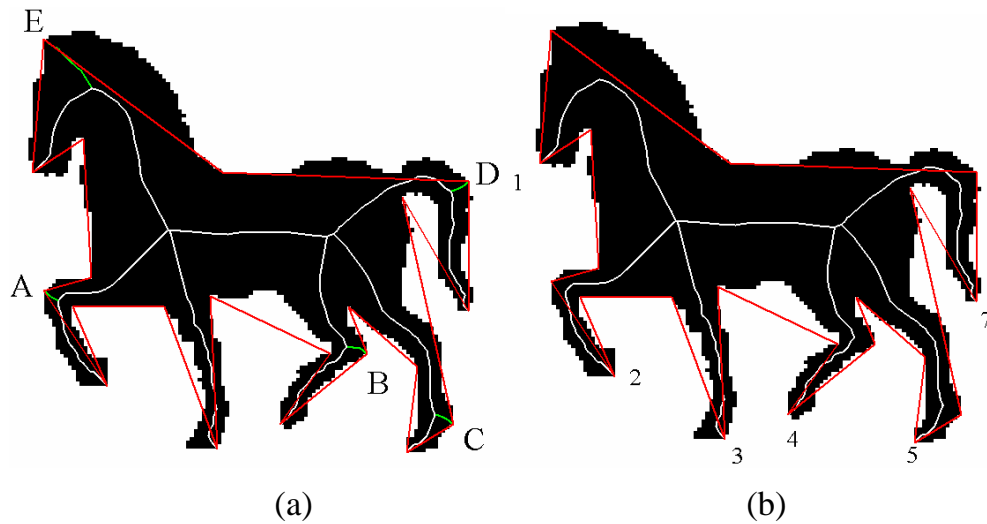
Hình 3.7(a) đưa ra một đa giác đơn giản với 7 đỉnh (màu đỏ) xương thu được dựa trên đa giác này. Nhánh xương màu xanh (kết thúc tại C) còn lại vì nó có những điểm tăng trưởng trên 2 cung khác nhau BC và CD của đường biên gốc. Nhánh xương màu xanh trong (b) không thuộc về xương được xác định bởi đa giác DCE khi nó kết thúc tại đỉnh lõm P. Trong (c) nó được loại bỏ bởi đơn giản hóa DCE.

Tác giả thực hiện phân tích đoạn DCE dựa trên các đỉnh lỗi bởi đơn giản hóa DCE. Khi một đỉnh lỗi trở thành một đỉnh lõm trong quá trình tiến hóa của DCE, thì những nhánh xương kết thúc tại đỉnh đó được loại bỏ. Cách tiếp cận này cho phép loại bỏ những nhánh nhỏ trong quá trình tiến hóa DCE. Hình 3.7 minh họa tại sao chỉ sử dụng những đỉnh lỗi để xác định các đoạn DCE. Nhánh xương màu xanh trong (a) sẽ trở thành một phần của bộ xương. Khi sử dụng đỉnh lõm của đa giác đơn giản (màu đỏ) trong (b) thì nhánh xương đó sẽ bị loại bỏ trong (c). Như vậy đoạn DCE được định nghĩa bằng cách chỉ sử dụng những đỉnh lỗi của đa giác đơn giản vì thế mà có thể cắt tia nhanh những nhánh không quan trọng.

Một thuộc tính quan trọng của DCE là gây ra phân chia đường biên và mỗi phân chia đó làm giảm các đỉnh của đường biên đa giác, kết quả có một nhánh xương kết thúc tại mỗi điểm phân chia. Theo kết quả ở trên, trong một 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 của nó) thì cung $[u_{i-1}, u_{i+1}]$ thay thế cung $[u_{i-1}, u_i], [u_i, u_{i+1}]$. Khi đó cắt tia xương sẽ loại bỏ toàn bộ nhánh xương kết thúc tại u_i .

Cho mỗi đỉnh lỗi v của đa giác, tác giả tính toán khoảng cách $D_1(v)$ giữa v và đỉnh lõm u gần nhất như đoạn vu là trong hình nếu như đỉnh u tồn tại. Sau đó loại bỏ đỉnh có giá trị thấp của phép đo liên quan mới $D_1(v)$.

Hình 3.8 minh họa hiệu quả của loại bỏ đỉnh lỗi v với phép đo liên quan $D_1(v)$. Có năm nhánh xương ngắn (màu xanh) kết thúc tại A, B, C, D, E của hình 3.8(a) được loại bỏ trong hình 3.8(b). Nó dẫn tới phân chia đường biên với 7 đỉnh lỗi được đánh số 1-7 trong hình 3.8(b).



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

Tóm lại, đỉnh V_f được sử dụng cho việc phân chia đường biên bởi DCE được tính toán như sau: $V_f = V_s - (V_{l\ddot{o}m} \cup V_1)$.

- V_s chỉ ra tất cả đỉnh của đa giác đơn giản P thu được bởi DCE.
- $V_{l\ddot{o}m}$ chỉ ra tất cả đỉnh lõm của V_s .
- V_1 chỉ ra đỉnh V_s với giá trị thấp của phép đo D_1 .

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

4.1 Môi trường cài đặt

Thực hiện cài đặt thuật toán cắt tỉa xương của ảnh trên môi trường cài đặt là Matlab (Matlab 7. 0).

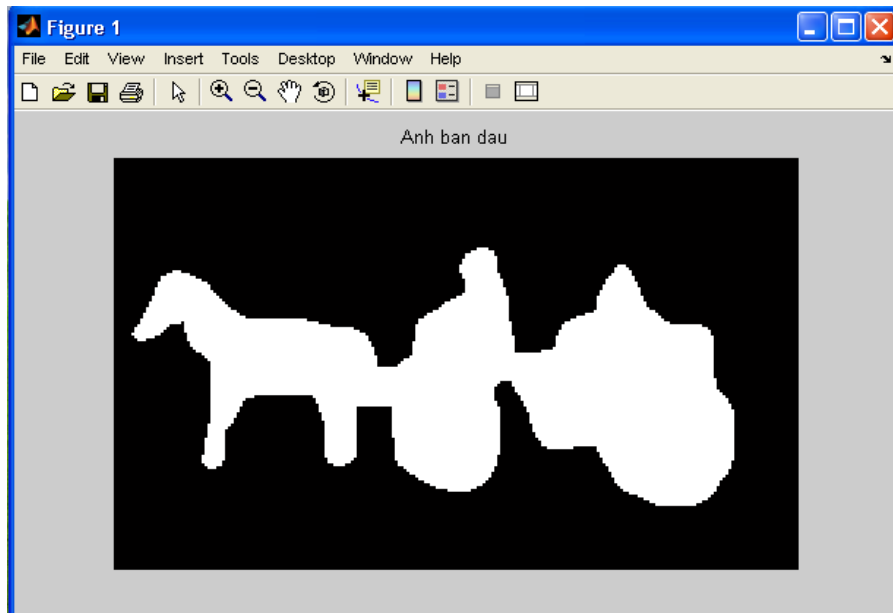
Yêu cầu về cấu hình máy tính:

- Bộ vi xử lý Pentium hoặc Pentium Pro.
- Window 95 hoặc NT trở lên.
- Dung lượng ổ cứng 25MB cho tới 1GB và tới 2.5GB nếu cài đặt Matlab cùng với Simulink.
- Bộ nhớ động (RAM) tối thiểu 16MB.

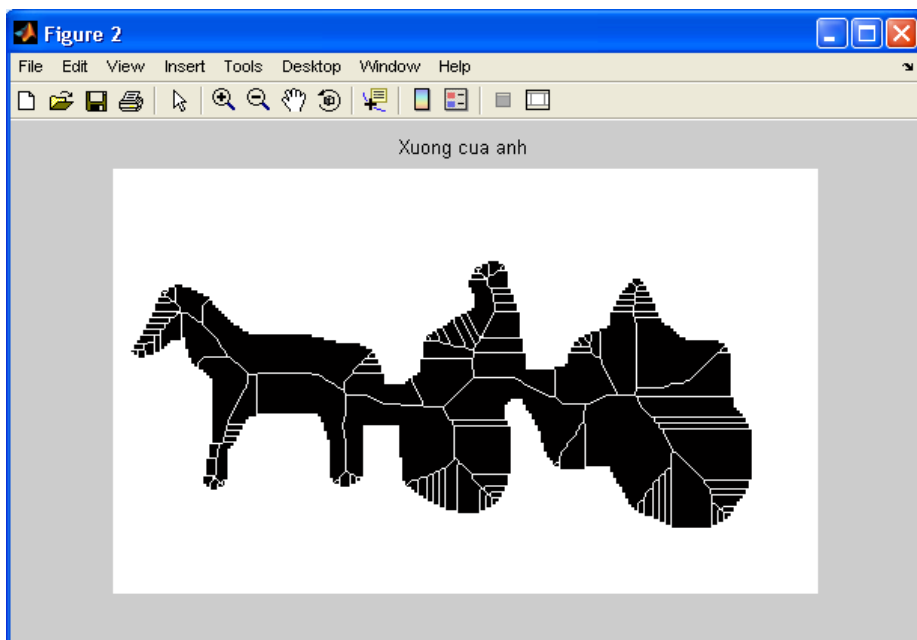
4.2 Chương trình

- Đầu vào là một ảnh thuộc tập dữ liệu MPEG-7 [37].
- Đầu ra là một ảnh sau khi tiến hành cắt tỉa xương.

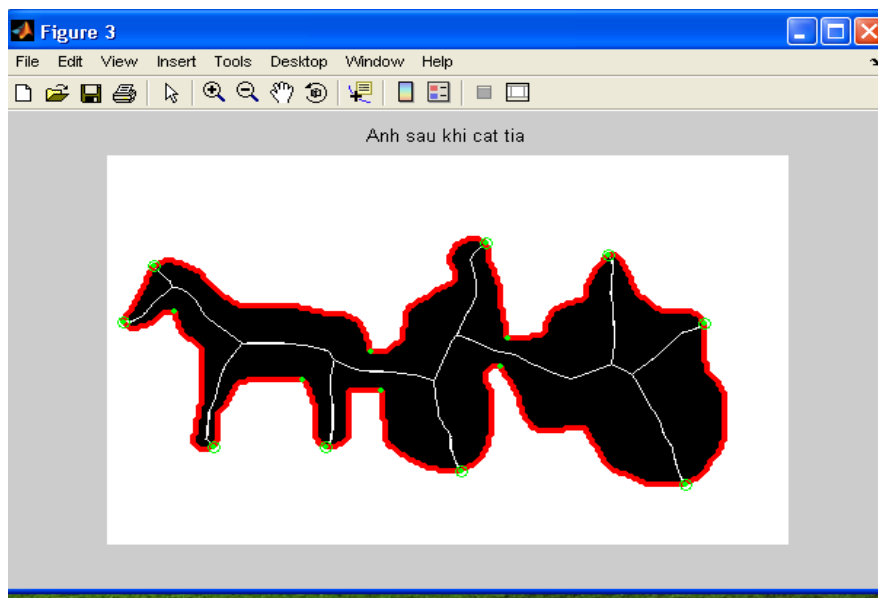
Giao diện chương trình



Hình 4.1. Ảnh đầu vào



Hình 4.2. Xương của ảnh



Hình 4.3. Ảnh sau khi cắt tia xương

KẾT LUẬN

Để hoàn thành đề tài đồ án tốt nghiệp “Kỹ thuật cắt tia xương của ảnh” em đã tìm hiểu về xử lý ảnh và bài báo “Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution” của tác giả Xiang Bai, Longin Jan Latecki, Wen-Yu Liu, từ đó em đã thu được một số thông tin như sau:

- Tổng quan về xử lý ảnh.
- Xương và các thuật toán tìm xương.
- Kỹ thuật cắt tia xương của ảnh.

Từ đó em xây dựng chương trình mô phỏng cắt tia xương của ảnh bằng ngôn ngữ matlab.

Tuy nhiên trong quá trình tìm hiểu bài báo do chưa có nhiều thời gian nên em chưa tìm hiểu hết được các mục tác giả đưa ra trong phần tài liệu tham khảo. Trong thời gian tới đây em sẽ cố gắng đọc các tài liệu đó để hiểu thêm về các thuật toán liên quan về xương trong xử lý ảnh.

TÀI LIỆU THAM KHẢO

- [1]. H. Blum. Biological Shape and Visual Science (Part I). *J. Theoretical Biology*, 38:205-287, 1973.
- [2]. K. Siddiqi, A. Shkoufandeh, S. Dickinson and S. Zucker. Shock Graphs and Shape Matching. In *ICCV*, 1998: 222-229.
- [3]. C. Di Ruberto. Recognition of shapes by attributed skeletal graphs. *Pattern Recognition*, 37: 21 –31, 2004.
- [4]. T. E. R. Hancock. A skeletal measure of 2D shape similarity. *Computer Vision and Image Understanding*, 95: 1 – 29, 2004.
- [5]. R. L. Ogniewicz, O. Kübler, Hierarchic Voronoi skeletons, *Pattern Recognition*, 28 (3): 343 –359, 1995.
- [6]. G. Malandain and S. Fernandez-Vidal. Euclidean skeletons. *Image and Vision Computing*, 16: 317– 327, 1998.
- [7]. W. -P. Choi, K. -M. Lam, and W. -C. Siu Extraction of the Euclidean skeleton based on a connectivity criterion. *Pattern Recognition*, 36: 721 – 729, 2003.
- [8]. C. Pudney. Distance-Ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images. *Computer Vision and Image Understanding*, 72 (3):404-413, 1998.
- [9]. W. Xie, R. P. Thompson, and R. Perucchio. A topology-preserving parallel 3D thinning algorithm for extracting the curve skeleton. *Pattern Recognition*, 36: 1529 – 1544, 2003.
- [10]. F. Leymarie and M. Levine. Simulating the grassfire transaction form using an active Contour model. *IEEE Trans. Pattern Analysis and Machine Intell*, 14 (1): 56 – 75, 1992.

[11]. P. Golland and E. Grimson. Fixed topology skeletons. In CVPR, Vol. 1, 2000, pp. 10-17.

[12]. N. Mayya and V. T. Rajan. Voronoi Diagrams of polygons: A framework for Shape Representation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 638 – 643.

[13]. Y. Ge, J. M. Fitzpatrick. On the Generation of Skeletons from Discrete Euclidean Distance Maps. *IEEE Trans. Pattern Analysis and Machine Intell.*, 18 (11):1055-1066, 1996.

[14]. Gold, C. M. , D. Thibault and Z. Liu. Map Generalization by Skeleton Retraction. *ICA Work- shop on Map Generalization*, Ottawa, August 1999. (pages?)

[15]. L. J. Latecki and R. Lakämper. Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution. *Computer Vision and Image Understanding (CVIU)*, vol. 73, pp. 441-454, 1999.

[16]. L. J. Latecki, R. Lakamper. Polygon evolution by vertex deletion. *Proc. of Int. Conf. on Scale- Space'99*, 1999, volume LNCS 1682.

[17]. L. J. Latecki, R. Lakamper, Shape similarity measure based on correspondence of visual parts, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. , 22 (10): 1185-1190, 2000.

[18]. L. J. Latecki, R. Lakamper. Application of planar shape comparison to object retrieval in image databases. *Pattern Recognition*, 35 (1): 15 – 29, 2002.

[19]. G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics and Image Processing*, 34 (3): 344-371, 1986.

[20]. D. Shaken and A. M. Bruckstein. Pruning Medial Axes. *Computer Vision and Image Understanding*, 69 (2): 156-169, 1998.

- [21]. K. Siddiqi, A. Tannenbaum, S. W. Zucker. Hyperbolic "Smoothing" of Shapes. In *ICCV*, 1998: 215-221.
- [22]. P. Dimitrov, J. N. Damon & K. Siddiqi. Flux Invariants for Shape. *Int. Conf. Computer Vision and Pattern Recognition*, 2003.
- [23]. L. J. Latecki, R. Ghadially, R. Lakämper, and U. Eckhardt. Continuity of the discrete curve evolution. *Journal of Electronic Imaging*, 9 (3), pp. 317-326, July 2000.
- [24]. P. Dimitrov, C. Phillips, and K. Siddiqi. Robust and Efficient Skeletal Graphs. In *CVPR*, 2000: 1417-1423.
- [25]. K. Siddiqi, S. Bouix, A. R. Tannenbaum, S. W. Zucker. Hamilton-Jacobi Skeletons. *International Journal of Computer Vision*, 48 (3): 215-231, 2002.
- [26]. A. Vasilevskiy and K. Siddiqi: Flux Maximizing Geometric Flows. *IEEE Trans. Pattern Analysis Machine Intell.* , 24 (12): 1565-1578, 2002.
- [27]. F. Y. L. Chin, J. Snoeyink, and C. An Wang. Finding the Medial Axis of a Simple Polygon in Linear Time. In *ISAAC*, 1995: 382-391.
- [28]. J. W. Brandt and V. R. Algazi. Continuous skeleton computation by Voronoi diagram. *Comput. Vision, Graphics, Image Process*, vol. 55 , pp. 329–338, 1992.
- [29]. S. C. Zhu and A. Yuille. FORMS: a Flexible Object Recognition and Modeling System. In *ICCV*, 1995.
- [30]. T. Liu, D. Geiger and R. V. Kohn. Representation and Self-Similarity of Shapes. In *ICCV*, Bombay, India, January 1998.
- [31]. C. Aslan, and S. Tari. An Axis Based Representation for Recognition. *ICCV* 2005.

- [32]. H. I. Choi, S. W. Choi, and H. P. Moon. Mathematical Theory of Medial Axis Transform. *Pacific Journal of Mathematics*, 181 (1): 57-88, 1997.
- [33]. C. Arcelli and G. Sanniti di Baja. Euclidean skeleton via center of maximal disk extraction. *Image and Vision Computing*, Vol. 11, pp. 163-173, 1993.
- [34]. C. Arcelli and G. Sanniti di Baja. A Width Independent Fast Thinning Algorithm. In *IEEE Trans. PAMI*, 7:463-474, 1985.
- [35]. R. Kimmel *et al.* Skeletonization via Distance Maps and Level Sets. *CVIU: Comp. Vision and Image Understanding*, 62 (3):382-391, 1995.
- [36]. T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* , vol. 26, no. 5, pp. 550-571, 2004.
- [37]. L. J. Latecki, R. Lakamper, and U. Eckhardt. Shape Descriptors for Non-rigid Shapes with a Single Closed Contour. *Proc. CVPR*, 2000.
- [38]. F. Mokhtarian and A. K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. PAMI*. 14: 789-805, 1992.
- [39]. S. M. Pizer, W. R. Oliver, and S. H. Bloomberg. Hierarchical shape description via the multi-resolution symmetric axis transform. *IEEE Trans. PAMI*. 9: 505-511, 1987.
- [40]. G. Borgefors, G. Ramella, and G. Sanniti di Baja. Hierarchical decomposition of multiscale skeletons. *IEEE Trans. PAMI*. 13 (11): 1296-1312, 2001.