

ĐẠI HỌC THÁI NGUYÊN
KHOA CÔNG NGHỆ THÔNG TIN

GIÁO TRÌNH MÔN HỌC

XỬ LÝ ẢNH

Người soạn : TS. ĐỖ NĂNG TOÀN,
TS. PHẠM VIỆT BÌNH

Thái Nguyên, Tháng 11 năm 2007

LỜI NÓI ĐẦU

Khoảng hơn mười năm trở lại đây, phần cứng máy tính và các thiết bị liên quan đã có sự tiến bộ vượt bậc về tốc độ tính toán, dung lượng chứa, khả năng xử lý v.v.. và giá cả đã giảm đến mức máy tính và các thiết bị liên quan đến xử lý ảnh đã không còn là thiết bị chuyên dụng nữa. Khái niệm ảnh số đã trở nên thông dụng với hầu hết mọi người trong xã hội và việc thu nhận ảnh số bằng các thiết bị cá nhân hay chuyên dụng cùng với việc đưa vào máy tính xử lý đã trở nên đơn giản.

Trong hoàn cảnh đó, xử lý ảnh là một lĩnh vực đang được quan tâm và đã trở thành môn học chuyên ngành của sinh viên ngành công nghệ thông tin trong nhiều trường đại học trên cả nước. Tuy nhiên, tài liệu giáo trình còn là một điều khó khăn. Hiện tại chỉ có một số ít tài liệu bằng tiếng Anh hoặc tiếng Pháp, tài liệu bằng tiếng Việt thì rất hiếm. Với mong muốn đóng góp vào sự nghiệp đào tạo và nghiên cứu trong lĩnh vực này, chúng tôi biên soạn cuốn giáo trình ***Xử lý ảnh*** dựa trên đề cương môn học đã được duyệt. Cuốn sách tập trung vào các vấn đề cơ bản của xử lý ảnh nhằm cung cấp một nền tảng kiến thức đầy đủ và chọn lọc nhằm giúp người đọc có thể tự tìm hiểu và xây dựng các chương trình ứng dụng liên quan đến xử lý ảnh.

Giáo trình được chia làm 5 chương và phần phụ lục: Chương 1, trình bày Tổng quan về xử lý ảnh, các khái niệm cơ bản, sơ đồ tổng quát của một hệ thống xử lý ảnh và các vấn đề cơ bản trong xử lý ảnh. Chương 2, trình bày các kỹ thuật nâng cao chất lượng ảnh dựa vào các thao tác với điểm ảnh, nâng cao chất lượng ảnh thông qua việc xử lý các điểm ảnh trong lân cận điểm ảnh đang xét. Chương này cũng trình bày các kỹ thuật nâng cao chất lượng ảnh nhờ vào các phép toán hình thái. Chương 3, trình bày các kỹ thuật cơ bản trong việc phát hiện biên của các đối tượng ảnh theo cả hai khuynh hướng: Phát hiện biên trực tiếp và phát hiện biên gián tiếp. Chương 4 thể hiện cách kỹ thuật tìm xương theo khuynh hướng tính toán trực trung vị và hướng tiếp cận xấp xỉ nhờ các thuật toán làm mảnh song song và gián tiếp. Và cuối cùng là Chương 5 với các kỹ thuật hậu xử lý.

Giáo trình được biên soạn dựa trên kinh nghiệm giảng dạy của tác giả trong nhiều năm tại các khóa đại học và cao học của ĐH Công nghệ - ĐHQG Hà Nội, ĐH Khoa học tự nhiên – ĐHQG Hà Nội, Khoa Công nghệ thông tin – ĐH Thái Nguyên v.v.. Cuốn sách có thể làm tài liệu tham khảo cho sinh viên các hệ kỹ sư, cử nhân và các bạn quan tâm đến vấn đề nhận dạng và xử lý ảnh.

Các tác giả bày tỏ lòng biết ơn chân thành tới các bạn đồng nghiệp trong Phòng Nhận dạng và công nghệ tri thức, Viện Công nghệ thông tin, Bộ môn Hệ thống thông tin, Khoa Công nghệ thông tin, ĐH Thái Nguyên, Khoa Công nghệ thông tin, ĐH Công nghệ, ĐHQG Hà Nội, Khoa Toán – Cơ – Tin, ĐH Khoa học tự nhiên, ĐHQG Hà Nội đã động viên, góp ý và giúp đỡ để hoàn chỉnh nội dung cuốn sách này. Xin cảm ơn Lãnh đạo Khoa Công nghệ thông tin, ĐH Thái Nguyên, Ban Giám đốc ĐH Thái Nguyên đã hỗ trợ và tạo điều kiện để cho ra đời giáo trình này.

Mặc dù rất cố gắng nhưng tài liệu này chắc chắn không tránh khỏi những sai sót. Chúng tôi xin trân trọng tiếp thu tất cả những ý kiến đóng góp của bạn đọc cũng như các bạn đồng nghiệp để có chỉnh lý kịp thời.

Thư góp ý xin gửi về: Phạm Việt Bình,

Khoa Công nghệ thông tin – ĐH Thái nguyên.

Xã Quyết Thắng, Tp. Thái Nguyên

Điện thoại: 0280.846506

Email: pvbinh@ictu.edu.vn

Thái Nguyên, ngày 22 tháng 11 năm 2007

CÁC TÁC GIẢ

MỤC LỤC

LỜI NÓI ĐẦU	2
MỤC LỤC	4
Chương 1: TỔNG QUAN VỀ XỬ LÝ ẢNH	7
1.1. XỬ LÝ ẢNH, CÁC VẤN ĐỀ CƠ BẢN TRONG XỬ LÝ ẢNH	7
1.1.1. Xử lý ảnh là gì?	7
1.1.2. Các vấn đề cơ bản trong xử lý ảnh	7
1.1.2.1 Một số khái niệm cơ bản	7
1.1.2.2 Nắn chỉnh biến dạng	8
1.1.2.3 Khử nhiễu	9
1.1.2.4 Chỉnh mức xám:	9
1.1.2.5 Trích chọn đặc điểm	9
1.1.2.6 Nhận dạng	10
1.1.2.7 Nén ảnh	11
1.2. THU NHẬN VÀ BIỂU DIỄN ẢNH	11
1.2.1. Thu nhận, các thiết bị thu nhận ảnh	11
1.2.2. Biểu diễn ảnh	12
1.2.2.1. Mô hình Raster	12
1.2.2.2. Mô hình Vector	13
Chương 2: CÁC KỸ THUẬT NÂNG CAO CHẤT LƯỢNG ẢNH	14
2.1. CÁC KỸ THUẬT KHÔNG PHỤ THUỘC KHÔNG GIAN	14
2.1.1. Giới thiệu	14
2.1.2. Tăng giảm độ sáng	14
2.1.3. Tách ngưỡng	15
2.1.4. Bó cụm	15
2.1.5. Cân bằng histogram	16
2.1.6. Kỹ thuật tách ngưỡng tự động	17
2.1.7. Biến đổi cấp xám tổng thể	18
2.2. CÁC KỸ THUẬT PHỤ THUỘC KHÔNG GIAN	20
2.2.1. Phép cuộn và mẫu	20

2.2.2. Một số mẫu thông dụng.....	21
2.2.3. Lọc trung vị.....	22
2.2.4. Lọc trung bình.....	24
2.2.5. Lọc trung bình theo k giá trị gần nhất.....	25
2.3. CÁC PHÉP TOÁN HÌNH THÁI HỌC.....	26
2.3.1. Các phép toán hình thái cơ bản.....	26
2.3.2. Một số tính chất của phép toán hình thái.....	27
Chương 3: BIÊN VÀ CÁC PHƯƠNG PHÁP PHÁT HIỆN BIÊN.....	32
3.1. GIỚI THIỆU.....	32
3.2. CÁC PHƯƠNG PHÁP PHÁT HIỆN BIÊN TRỰC TIẾP.....	32
3.2.1. Kỹ thuật phát hiện biên Gradient.....	32
3.2.1.1. Kỹ thuật Prewitt.....	34
3.2.1.2. Kỹ thuật Sobel.....	35
3.2.1.3. Kỹ thuật la bàn.....	35
3.2.2. Kỹ thuật phát hiện biên Laplace.....	36
3.3. PHÁT HIỆN BIÊN GIÁN TIẾP.....	37
3.3.1 Một số khái niệm cơ bản.....	37
3.3.2. Chu tuyến của một đối tượng ảnh.....	38
3.3.3. Thuật toán dò biên tổng quát.....	40
Chương 4: XƯƠNG VÀ CÁC KỸ THUẬT TÌM XƯƠNG.....	44
4.1. GIỚI THIỆU.....	44
4.2. TÌM XƯƠNG DỰA TRÊN LÀM MẢNH.....	44
4.2.1. Sơ lược về thuật toán làm mảnh.....	44
4.2.2. Một số thuật toán làm mảnh.....	46
4.3. TÌM XƯƠNG KHÔNG DỰA TRÊN LÀM MẢNH.....	46
4.3.1. Khái quát về lược đồ Voronoi.....	47
4.3.2. Trục trung vị Voronoi rời rạc.....	47
4.3.3. Xương Voronoi rời rạc.....	48
4.3.4. Thuật toán tìm xương.....	49
Chương 5: CÁC KỸ THUẬT HẬU XỬ LÝ.....	52
5.1. RÚT GỌN SỐ LƯỢNG ĐIỂM BIỂU DIỄN.....	52
5.1.1. Giới thiệu.....	52

5.1.2. Thuật toán Douglas Peucker	52
5.1.2.1. Ý tưởng	52
5.1.2.2. Chương trình	53
5.1.3. Thuật toán Band width	54
5.1.3.1. Ý tưởng	54
5.1.3.2. Chương trình	56
5.1.4. Thuật toán Angles	57
5.1.4.1. Ý tưởng	57
5.1.4.2. Chương trình	57
5.2. XẤP XỈ ĐA GIÁC BỞI CÁC HÌNH CƠ SỞ	58
5.2.1 Xấp xỉ đa giác theo bất biến đồng dạng	59
5.2.2 Xấp xỉ đa giác theo bất biến afin	62
5.3. BIẾN ĐỔI HOUGH	63
5.3.1. Biến đổi Hough cho đường thẳng	63
5.3.2. Biến đổi Hough cho đường thẳng trong tọa độ cực	64
5.3.2.1. Đường thẳng Hough trong tọa độ cực	64
5.3.2.2. Áp dụng biến đổi Hough trong phát hiện góc nghiêng văn bản	65
PHỤ LỤC	68
TÀI LIỆU THAM KHẢO	76

Chương 1:

TỔNG QUAN VỀ XỬ LÝ ẢNH

1.1. XỬ LÝ ẢNH, CÁC VẤN ĐỀ CƠ BẢN TRONG XỬ LÝ ẢNH

1.1.1. 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.

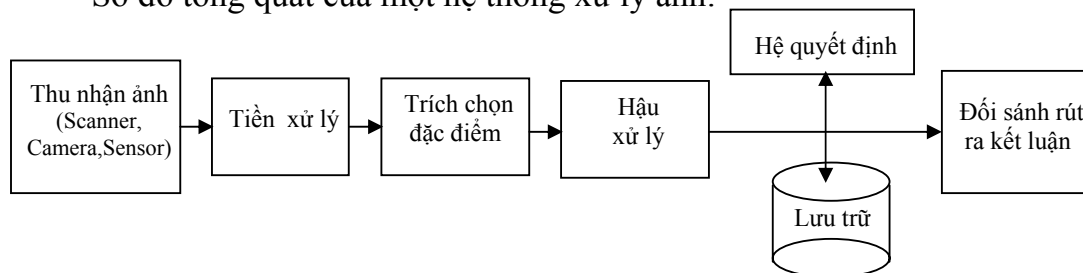
Quá trình xử lý ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.



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

1.1.2. Các vấn đề cơ bản trong xử lý ảnh

1.1.2.1 Một số khái niệm cơ bản

* *Ảnh và điểm ảnh:*

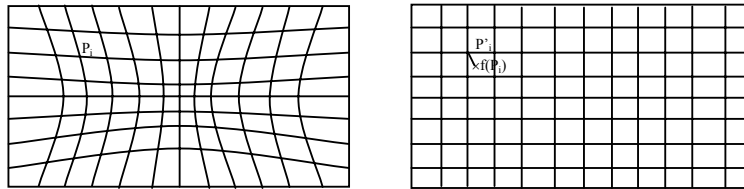
Điểm ảnh được xem như là dấu hiệu hay cường độ sáng tại 1 tọa độ trong không gian của đối tượng và ảnh được xem như là 1 tập hợp các điểm ảnh.

* *Mức xám, màu*

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

1.1.2.2 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 1.3. 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 $(P_i, P'_i) \quad i = \overline{1, n}$ có n các tập điều khiển

Tìm hàm $f: P_i \mapsto f(P_i)$ 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:

$$\phi = \sum_{i=1}^n (f(P_i) - P'_i)^2 = \sum_{i=1}^n \left[(a_1x_i + b_1y_i + c_1 - x'_i)^2 + (a_2x_i + b_2y_i + c_2 - y'_i)^2 \right]$$

Đề cho $\phi \rightarrow \min$

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

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

Tương tự tìm được a_2, b_2, c_2

\Rightarrow Xác định được hàm f

1.1.2.3 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 \rightarrow khắc phục bằng các phép lọc

1.1.2.4 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

1.1.2.5 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 Laplace, 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.

1.1.2.6 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:

- 1°. Thu nhận dữ liệu và tiền xử lý.
- 2°. Biểu diễn dữ liệu.
- 3°. Nhận dạng, ra quyết định.

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

- 1°. Đối sánh mẫu dựa trên các đặc trưng được trích chọn.
- 2°. Phân loại thống kê.
- 3°. Đối sánh cấu trúc.

4°. 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.

1.1.2.7 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 nên 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

1.2. THU NHẬN VÀ BIỂU DIỄN ẢNH

1.2.1. 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. 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

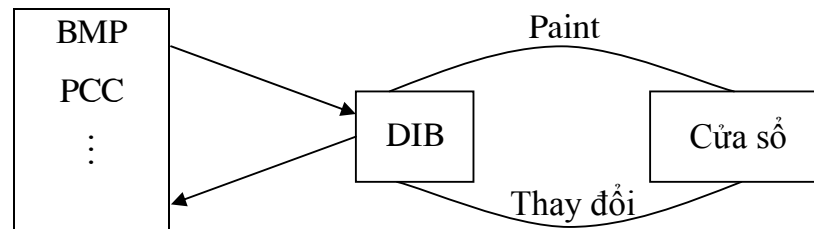
1.2.2.1. Mô hình Raster

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 hiện 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 1.4. Quá trình hiển thị và chỉnh sửa, lưu trữ ảnh thông qua DIB

1.2.2.2. 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 vector đều tập trung từ chuyển đổi từ ảnh Raster.



Hình 1.5. Sự chuyển đổi giữa các mô hình biểu diễn ảnh

Chương 2:

CÁC KỸ THUẬT NÂNG CAO CHẤT LƯỢNG ẢNH

2.1. CÁC KỸ THUẬT KHÔNG PHỤ THUỘC KHÔNG GIAN

2.1.1. Giới thiệu

Các phép toán không phụ thuộc không gian là các phép toán không phụ thuộc vị trí của điểm ảnh.

Ví dụ: Phép tăng giảm độ sáng, phép thống kê tần suất, biến đổi tần suất v.v..

Một trong những khái niệm quan trọng trong xử lý ảnh là biểu đồ tần suất (Histogram)

Biểu đồ tần suất của mức xám g của ảnh I là số điểm ảnh có giá trị g của ảnh I . Ký hiệu là $h(g)$

Ví dụ:

$$I = \begin{pmatrix} 1 & 2 & 0 & 4 \\ 1 & 0 & 0 & 7 \\ 2 & 2 & 1 & 0 \\ 4 & 1 & 2 & 1 \\ 2 & 0 & 1 & 1 \end{pmatrix}$$

g	0	1	2	4	7
$h(g)$	5	7	5	2	1

2.1.2. Tăng giảm độ sáng

Giả sử ta có I ~ kích thước $m \times n$ và số nguyên c

Khi đó, kỹ thuật tăng, giảm độ sáng được thể hiện

for ($i = 0$; $i < m$; $i++$)

for ($j = 0$; $j < n$; $j++$)

$I[i, j] = I[i, j] + c$;

- Nếu $c > 0$: ảnh sáng lên
- Nếu $c < 0$: ảnh tối đi

2.1.3. Tách ngưỡng

Giả sử ta có ảnh $I \sim$ kích thước $m \times n$, hai số Min, Max và ngưỡng θ khi đó: Kỹ thuật tách ngưỡng được thể hiện

for ($i = 0$; $i < m$; $i++$)

for ($j = 0$; $j < n$; $j++$)

$I[i, j] = I[i, j] \geq \theta ? \text{Max} : \text{Min};$

* *Ứng dụng:*

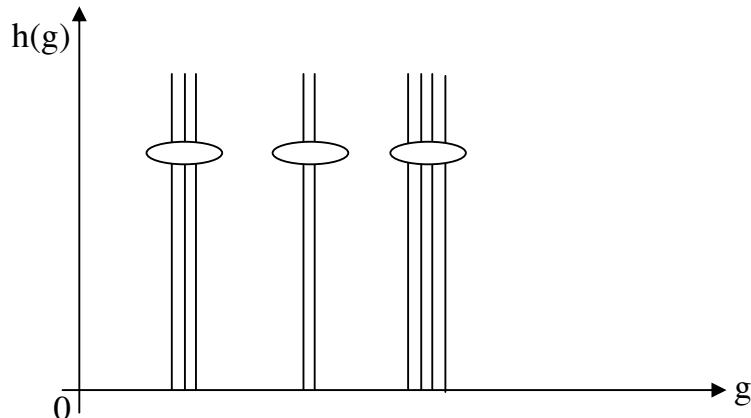
Nếu $\text{Min} = 0$, $\text{Max} = 1$ kỹ thuật chuyển ảnh thành ảnh đen trắng được ứng dụng khi quét và nhận dạng văn bản có thể xảy ra sai sót nền thành ảnh hoặc ảnh thành nền dẫn đến ảnh bị đứt nét hoặc dính.

2.1.4. Bó cụm

Kỹ thuật nhằm giảm bớt số mức xám của ảnh bằng cách nhóm lại số mức xám gần nhau thành 1 nhóm

Nếu chỉ có 2 nhóm thì chính là kỹ thuật tách ngưỡng. Thông thường có nhiều nhóm với kích thước khác nhau.

Để tổng quát khi biến đổi người ta sẽ lấy cùng 1 kích thước bunch_size



$$I[i, j] = I[i, j] / \text{bunch_size} * \text{bunch_size} \quad \forall (i, j)$$

Ví dụ: Bó cụm ảnh sau với $\text{bunch_size} = 3$

$$I = \begin{pmatrix} 1 & 2 & 4 & 6 & 7 \\ 2 & 1 & 3 & 4 & 5 \\ 7 & 2 & 6 & 9 & 1 \\ 4 & 1 & 2 & 1 & 2 \end{pmatrix}$$

$$I_{kq} = \begin{pmatrix} 0 & 0 & 3 & 6 & 6 \\ 0 & 0 & 3 & 3 & 3 \\ 6 & 0 & 6 & 9 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2.1.5. Cân bằng histogram

Ảnh I được gọi là cân bằng "lý tưởng" nếu với mọi mức xám g, g' ta có $h(g) = h(g')$

Giả sử, ta có ảnh $I \sim$ kích thước $m \times n$

$new_level \sim$ số mức xám của ảnh cân bằng

$TB = \frac{m \times n}{new_level} \sim$ số điểm ảnh trung bình của mỗi mức xám của ảnh cân bằng

$$t(g) = \sum_{i=0}^g h(i) \sim \text{số điểm ảnh có mức xám} \leq g$$

Xác định hàm f: $g \mapsto f(g)$

$$\text{Sao cho: } f(g) = \max \left\{ 0, \text{round} \left(\frac{t(g)}{TB} \right) - 1 \right\}$$

Ví dụ: Cân bằng ảnh sau với $new_level = 4$

$$I = \begin{pmatrix} 1 & 2 & 4 & 6 & 7 \\ 2 & 1 & 3 & 4 & 5 \\ 7 & 2 & 6 & 9 & 1 \\ 4 & 1 & 2 & 1 & 2 \end{pmatrix}$$

g	h(g)	t(g)	f(g)
1	5	5	0
2	5	10	1
3	1	11	1
4	3	14	2
5	1	15	2
6	2	17	2
7	2	19	3
9	1	20	3

$$I_{kq} = \begin{pmatrix} 0 & 1 & 2 & 2 & 3 \\ 1 & 0 & 1 & 2 & 2 \\ 3 & 1 & 2 & 3 & 0 \\ 2 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Chú ý: Ảnh sau khi thực hiện cân bằng chưa chắc đã là cân bằng "lý tưởng"

2.1.6. Kỹ thuật tách ngưỡng tự động

Ngưỡng θ trong kỹ thuật tách ngưỡng thường được cho bởi người sử dụng. Kỹ thuật tách ngưỡng tự động nhằm tìm ra ngưỡng θ một cách tự động dựa vào histogram theo nguyên lý trong vật lý là vật thể tách làm 2 phần nếu tổng độ lệch trong từng phần là tối thiểu.

Giả sử, ta có ảnh $I \sim$ kích thước $m \times n$

$G \sim$ là số mức xám của ảnh kể cả khuyết thiếu

$t(g) \sim$ số điểm ảnh có mức xám $\leq g$

$$m(g) = \frac{1}{t(g)} \sum_{i=0}^g i \cdot h(i) \sim \text{mômen quán tính TB có mức xám } \leq g$$

Hàm $f: g \mapsto f(g)$

$$f(g) = \frac{t(g)}{mxn - t(g)} [m(g) - m(G-1)]^2$$

Tìm θ sao cho:

$$f(\theta) = \max_{0 \leq g < G-1} \{f(g)\}$$

Ví dụ: Tìm ngưỡng tự động của ảnh sau

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

Lập bảng

g	$h(g)$	$t(g)$	$g \cdot h(g)$	$\sum_{i=0}^g i \cdot h(i)$	$m(g)$	$f(g)$
0	15	15	0	0	0	1.35
1	5	20	5	5	0,25	1.66

2	4	24	8	13	0,54	1.54
3	3	27	9	22	0,81	1.10
4	2	29	8	30	1,03	0.49
5	1	30	5	35	1,16	∞

Ngưỡng cần tách $\theta = 1$ ứng với $f(\theta) = 1.66$

2.1.7. Biến đổi cấp xám tổng thể

Nếu biết ảnh và hàm biến đổi thì ta có thể tính được ảnh kết quả và do đó ta sẽ có được histogram của ảnh biến đổi. Nhưng thực tế nhiều khi ta chỉ biết histogram của ảnh gốc và hàm biến đổi, câu hỏi đặt ra là liệu ta có thể có được histogram của ảnh biến đổi. Nếu có như vậy ta có thể hiệu chỉnh hàm biến đổi để thu được ảnh kết quả có phân bố histogram như mong muốn.

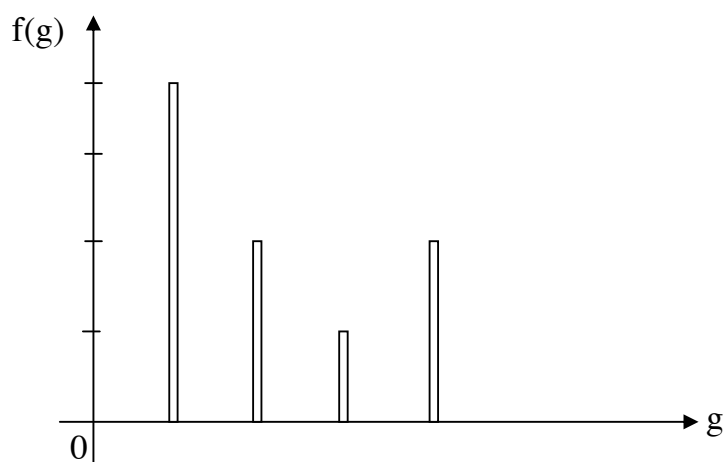
Bài toán đặt ra là biết histogram của ảnh, biết hàm biến đổi hãy vẽ histogram của ảnh mới.

Ví dụ:

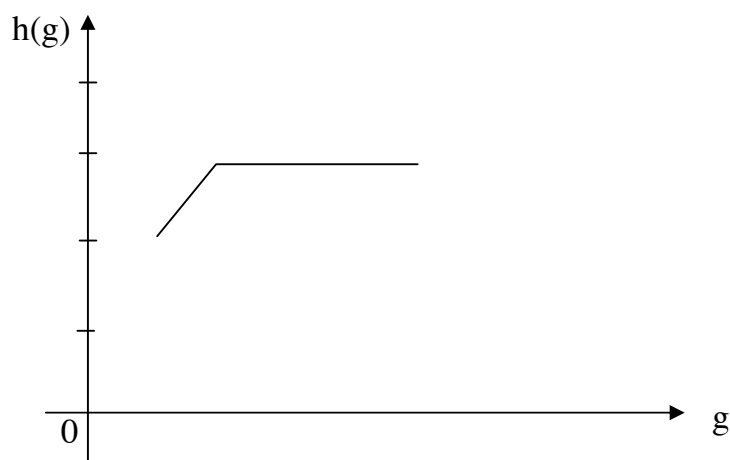
g	1	2	3	4
$h(g)$	4	2	1	2

$$f(g) = \begin{cases} g + 1 & \text{nếu } g \leq 2 \\ g & \text{nếu } g = 3 \\ g - 1 & \text{nếu } g > 3 \end{cases}$$

Bước 1: Vẽ Histogram của ảnh cũ



Bước 2: Vẽ đồ thị hàm $f(g)$



Bước 3: Vẽ Histogram của ảnh mới

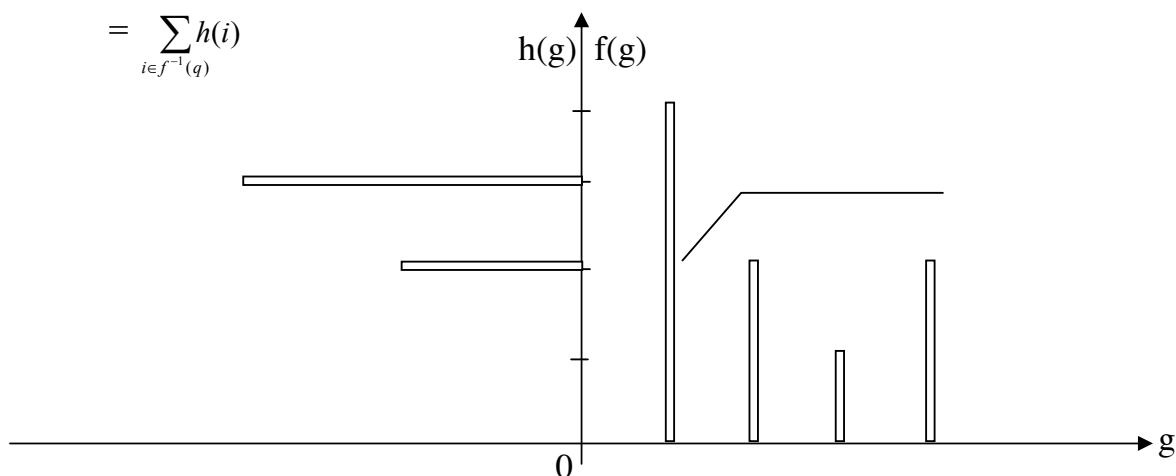
Đặt $q = f(g)$

$$h(q) = \text{card}(\{P \mid I(P) = q\})$$

$$= \text{card}(\{P \mid I(P) = f(g)\})$$

$$= \text{card}(\{P \mid g = f^{-1}(I(P))\})$$

$$= \sum_{i \in f^{-1}(q)} h(i)$$



Histogram của ảnh mới thu được bằng cách chồng hình và tính giá trị theo các $q (= f(g))$ theo công thức tính trên. Kết quả cuối thu được sau phép quay góc 90 thuận chiều kim đồng hồ.

2.2. CÁC KỸ THUẬT PHỤ THUỘC KHÔNG GIAN

2.2.1. Phép cuộn và mẫu

Giả sử ta có ảnh I kích thước $M \times N$, mẫu T có kích thước $m \times n$ khi đó, ảnh I cuộn theo mẫu T được xác định bởi công thức.

$$I \otimes T(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I(x+i, y+j) * T(i, j) \quad (2.1)$$

$$\text{Hoặc } I \otimes T(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I(x-i, y-j) * T(i, j) \quad (2.2)$$

VD:

$$I = \begin{pmatrix} 1 & 2 & 4 & 5 & 8 & 7 \\ 2 & 1 & 1 & 4 & 2 & 2 \\ 4 & 5 & 5 & 8 & 8 & 2 \\ 1 & 2 & 1 & 1 & 4 & 4 \\ 7 & 2 & 2 & 1 & 5 & 2 \end{pmatrix}$$

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$I \otimes T(x, y) = \sum_{i=0}^1 \sum_{j=0}^1 I(x+i, y+j) * T(i, j) = I(x, y) * T(0, 0) + I(x+1, y+1) * T(1, 1)$$

$$= I(x, y) + I(x+1, y+1)$$

$$I \otimes T = \begin{pmatrix} 2 & 3 & 8 & 7 & 10 & * \\ 7 & 6 & 9 & 12 & 4 & * \\ 6 & 6 & 6 & 12 & 12 & * \\ 3 & 4 & 2 & 6 & 6 & * \\ * & * & * & * & * & * \end{pmatrix} \quad \text{Tính theo (2.1)}$$

Tính theo công thức 2.2

$$I \otimes T = \begin{pmatrix} * & * & * & * & * & * \\ * & 2 & 3 & 8 & 7 & 10 \\ * & 7 & 6 & 9 & 12 & 4 \\ * & 6 & 6 & 6 & 12 & 12 \\ * & 3 & 4 & 2 & 6 & 6 \end{pmatrix}$$

*** Nhân xét:**

- Trong quá trình thực hiện phép cuộn có một số thao tác ra ngoài ảnh, ảnh không được xác định tại những vị trí đó dẫn đến ảnh thu được có kích thước nhỏ hơn.

- Ảnh thực hiện theo công thức 2.1 và 2.2 chỉ sai khác nhau 1 phép dịch chuyển để đơn giản ta sẽ hiểu phép cuộn là theo công thức 2.1

2.2.2. Một số mẫu thông dụng

- Mẫu:

$$T_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

~ Dùng để khử nhiễu \Rightarrow Các điểm có tần số cao

VD1:

$$I = \begin{pmatrix} 1 & 2 & 4 & 5 & 8 & 7 \\ 2 & 31 & 1 & 4 & 2 & 2 \\ 4 & 5 & 5 & 8 & 8 & 2 \\ 1 & 2 & 1 & 1 & 4 & 4 \\ 7 & 2 & 2 & 1 & 5 & 2 \end{pmatrix}$$
$$I \otimes T_1 = \begin{pmatrix} 55 & 65 & 45 & 46 & * & * \\ 52 & 58 & 34 & 35 & * & * \\ 29 & 27 & 35 & 35 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

Áp dụng kỹ thuật cộng hằng số với $c = -27$, ta có:

$$I_{kq} = \begin{pmatrix} 28 & 38 & 18 & 19 & * & * \\ 25 & 31 & 7 & 8 & * & * \\ 2 & 0 & 8 & 8 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

- Mẫu:

$$T_2 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

~ Dùng để phát hiện các điểm có tần số cao

VD2:

$$I \otimes T2 = -1 \begin{pmatrix} 114 & -40 & 0 & -14 & * & * \\ -22 & 5 & 14 & 16 & * & * \\ -6 & -10 & -2 & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

2.2.3. Loại trung vị

* Định nghĩa 2.1 (Trung vị)

Cho dãy $x_1; x_2...; x_n$ đơn điệu tăng (giảm). Khi đó trung vị của dãy ký hiệu là $Med(\{x_n\})$, được định nghĩa:

$$+ \text{ Nếu } n \text{ lẻ } x_{\left\lfloor \frac{n}{2} + 1 \right\rfloor}$$

$$+ \text{ Nếu } n \text{ chẵn: } x_{\left\lfloor \frac{n}{2} \right\rfloor} \text{ hoặc } x_{\left\lfloor \frac{n}{2} + 1 \right\rfloor}$$

* Mệnh đề 2.1

$$\sum_{i=1}^n |x - x_i| \rightarrow \min \text{ tại } Med(\{x_n\})$$

Chứng minh

+ Xét trường hợp n chẵn

$$\text{Đặt } M = \frac{n}{2}$$

Ta có:

$$\begin{aligned} \sum_{i=1}^n |x - x_i| &= \sum_{i=1}^M |x - x_i| + \sum_{i=1}^M |x - x_{M+i}| \\ &= \sum_{i=1}^M (|x - x_i| + |x_{M+i} - x|) \geq \sum_{i=1}^M |x_{M+i} - x_i| \\ &= \sum_{i=1}^M [(x_{M+i} - x_M) + (x_M - x_i)] \\ &= \sum_{i=1}^M |x_{M+i} - Med(\{x_i\})| + \sum_{i=1}^M |x_i - Med(\{x_i\})| \end{aligned}$$

$$= \sum_{i=1}^n |x_i - Med(\{x_i\})|$$

+ Nếu n lẻ:

Bổ sung thêm phần tử $Med(\{x_i\})$ vào dãy. Theo trường hợp n chẵn ta có:

$$\sum_{i=1}^n |x - x_i| + |Med(\{x_i\}) - Med(\{x_i\})| \rightarrow \min \text{ tại } Med(\{x_n\})$$

$$\sum_{i=1}^n |x - x_i| \rightarrow \min \text{ tại } Med(\{x_n\})$$

* Kỹ thuật lọc trung vị

Giả sử ta có ảnh I người θ của số W(P) và điểm ảnh P

Khi đó kỹ thuật lọc trung vị phụ thuộc không gian bao gồm các bước cơ bản sau:

+ **Bước 1:** Tìm trung vị

$$\{I(q) | q \in W(P)\} \rightarrow Med(P)$$

+ **Bước 2:** Gán giá trị

$$I(P) = \begin{cases} I(P) & |I(P) - Med(P)| \leq \theta \\ Med(P) & \text{Nguoc lai} \end{cases}$$

Ví dụ:

$$I = \begin{pmatrix} 1 & 2 & 3 & 2 \\ 4 & 16 & 2 & 1 \\ 4 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 \end{pmatrix}$$

$W(3 \times 3); \theta = 2$

$$I_{kq} = \begin{pmatrix} 1 & 2 & 3 & 2 \\ 4 & \textcircled{2} & 2 & 1 \\ 4 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 \end{pmatrix}$$

Giá trị 16, sau phép lọc có giá trị 2, các giá trị còn lại không thay đổi giá trị.

2.2.4. Lọc trung bình

* Định nghĩa 2.2 (Trung bình)

Cho dãy x_1, x_2, \dots, x_n khi đó trung bình của dãy ký hiệu $AV(\{x_n\})$ được định nghĩa:

$$AV(\{x_n\}) = \text{round}\left(\frac{1}{n} \sum_{i=1}^n x_i\right)$$

* Mệnh đề 2.2

$$\sum_{i=1}^n (x - x_i)^2 \rightarrow \min \text{ tại } AV(\{x_n\})$$

Chứng minh:

$$\text{Đặt: } \phi(x) = \sum_{i=1}^n (x - x_i)^2$$

Ta có:

$$\phi(x) = 2 \sum_{i=1}^n (x - x_i)$$

$$\phi'(x) = 0$$

$$\Leftrightarrow \sum_{i=1}^n (x - x_i) = 0$$

$$\Leftrightarrow x = \frac{1}{n} \sum_{i=1}^n x_i = AV(\{x_i\})$$

$$\text{Mặt khác, } \phi''(x) = 2n > 0$$

$$\Rightarrow \phi \rightarrow \min \text{ tại } x = AV(\{x_i\})$$

Kỹ thuật lọc trung bình

Giả sử ta có ảnh I, điểm ảnh P, cửa sổ W(P) và ngưỡng θ . Khi đó kỹ thuật lọc trung bình phụ thuộc không gian bao gồm các bước cơ bản sau:

+ **Bước 1:** Tìm trung bình

$$\{I(q) | q \in W(P)\} \rightarrow AV(P)$$

+ **Bước 2:** Gán giá trị

$$I(P) = \begin{cases} I(P) & |I(P) - AV(P)| \leq \theta \\ AV(P) & \text{Nguoc lai} \end{cases}$$

Ví dụ:

$$I = \begin{pmatrix} 1 & 2 & 3 & 2 \\ 4 & 16 & 2 & 1 \\ 4 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 \end{pmatrix}$$

$W(3 \times 3); \theta = 2$

$$I_{kq} = \begin{pmatrix} 1 & 2 & 3 & 2 \\ 4 & \textcircled{3} & 2 & 1 \\ 4 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 \end{pmatrix}$$

Giá trị 16 sau phép lọc trung bình có giá trị 3, các giá trị còn lại giữ nguyên sau phép lọc.

2.2.5. Lọc trung bình theo k giá trị gần nhất

Giả sử ta có ảnh I, điểm ảnh P, cửa sổ $W(P)$, ngưỡng θ và số k. Khi đó, lọc trung bình theo k giá trị gần nhất bao gồm các bước sau:

+ **Bước 1:** Tìm K giá trị gần nhất

$$\{I(q) \mid q \in W(p)\} \rightarrow \{k \sim \text{giá trị gần } I(P) \text{ nhất}\}$$

+ **Bước 2:** Tính trung bình

$$\{k \sim \text{giá trị gần } I(P) \text{ nhất}\} \rightarrow AV_k(P)$$

+ **Bước 3:** Gán giá trị

$$I(P) = \begin{cases} I(P) & |I(P) - AV_k(P)| \leq \theta \\ AV_k(P) & \text{Nguoc lai} \end{cases}$$

Ví dụ:

$$I = \begin{pmatrix} 1 & 2 & 3 & 2 \\ 4 & 16 & 2 & 1 \\ 4 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 \end{pmatrix}$$

$W(3 \times 3); \theta = 2; k = 3$

$$I_{kq} = \begin{pmatrix} 1 & 2 & 3 & 2 \\ 4 & 8 & 2 & 1 \\ 4 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 \end{pmatrix}$$

* *Nhận xét:*

- Nếu k lớn hơn kích thước của số thì kỹ thuật chính là kỹ thuật lọc trung bình

- Nếu k= 1 thì ảnh kết quả không thay đổi

⇒ Chất lượng của kỹ thuật phụ thuộc vào số phân tử lựa chọn k.

2.3. CÁC PHÉP TOÁN HÌNH THÁI HỌC

2.3.1. Các phép toán hình thái cơ bản

Hình thái là thuật ngữ chỉ sự nghiên cứu về cấu trúc hay hình học topo của đối tượng trong ảnh. Phần lớn các phép toán của "Hình thái" được định nghĩa từ hai phép toán cơ bản là phép "giãn nở" (Dilation) và phép "co" (Erosion).

Các phép toán này được định nghĩa như sau: Giả thiết ta có đối tượng X và phần tử cấu trúc (mẫu) B trong không gian Euclide hai chiều. Kí hiệu B_x là dịch chuyển của B tới vị trí x.

Định nghĩa 2.3 (DILATION)

Phép "giãn nở" của X theo mẫu B là hợp của tất cả các B_x với x thuộc X. Ta có:

$$X \oplus B = \bigcup_{x \in X} B_x$$

Định nghĩa 2.4 (EROSION)

Phép "co" của X theo B là tập hợp tất cả các điểm x sao cho B_x nằm trong X. Ta có:

$$X \ominus B = \{x : B_x \subseteq X\}$$

Ví dụ: Ta có tập X như sau:
$$X = \begin{pmatrix} 0 & x & 0 & x & x \\ x & 0 & x & x & 0 \\ 0 & x & x & 0 & 0 \\ 0 & x & 0 & x & 0 \\ 0 & x & x & x & 0 \end{pmatrix} \quad B = \begin{array}{|c|c|} \hline \otimes & x \\ \hline \end{array}$$

$$X \oplus B = \begin{pmatrix} 0 & x & x & x & x \\ x & x & x & x & x \\ 0 & x & x & x & 0 \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{pmatrix} \quad \text{và} \quad X \ominus B = \begin{pmatrix} 0 & 0 & 0 & x & 0 \\ 0 & 0 & x & 0 & 0 \\ 0 & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 \end{pmatrix}$$

Định nghĩa 2.5 (OPEN)

Phép toán mở (OPEN) của X theo cấu trúc B là tập hợp các điểm của ảnh X sau khi đã co và giãn nở liên tiếp theo B. Ta có:

$$\text{OPEN}(X, B) = (X \ominus B) \oplus B$$

Ví dụ: Với tập X và B trong ví dụ trên ta có

$$\text{OPEN}(X, B) = (X \ominus B) \oplus B = \begin{pmatrix} 0 & 0 & 0 & x & x \\ 0 & 0 & x & x & 0 \\ 0 & x & x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & x & x & x & 0 \end{pmatrix}$$

Định nghĩa 2.6 (CLOSE)

Phép toán đóng (CLOSE) của X theo cấu trúc B là tập hợp các điểm của ảnh X sau khi đã giãn nở và co liên tiếp theo B. Ta có:

$$\text{CLOSE}(X, B) = (X \oplus B) \ominus B$$

Theo ví dụ trên ta có:

$$\text{CLOSE}(X, B) = (X \oplus B) \ominus B = \begin{pmatrix} 0 & x & x & x & x \\ x & x & x & x & x \\ 0 & x & x & 0 & 0 \\ 0 & x & x & x & 0 \\ 0 & x & x & x & 0 \end{pmatrix}$$

2.3.2. Một số tính chất của phép toán hình thái

* **Mệnh đề 2.3 [Tính gia tăng]:**

$$\begin{aligned} \text{(i)} \quad X \subseteq X' &\Rightarrow \begin{cases} X \ominus B \subseteq X' \ominus B & \forall B \\ X \oplus B \subseteq X' \oplus B & \forall B \end{cases} \\ \text{(ii)} \quad B \subseteq B' &\Rightarrow \begin{cases} X \ominus B \supseteq X \ominus B' & \forall X \\ X \oplus B \subseteq X \oplus B' & \forall X \end{cases} \end{aligned}$$

Chứng minh:

$$(i) \quad X \oplus B = \bigcup_{x \in X} B_x \subseteq \bigcup_{x \in X'} B_x = X' \oplus B$$

$$X \ominus B = \{x / B_x \subseteq X\} \subseteq \{x / B_x \subseteq X'\} = X' \ominus B$$

$$(ii) \quad X \oplus B = \bigcup_{x \in X} B_x \subseteq \bigcup_{x \in X} B'_x = X \oplus B'$$

Theo định nghĩa:

$$X \ominus B' = \{x / B'_x \subseteq X\} \subseteq \{x / B_x \subseteq X\} = X \ominus B.$$

***Mệnh đề 2.4 [Tính phân phối với phép \cup]:**

$$(i) \quad X \oplus (B \cup B') = (X \oplus B) \cup (X \oplus B')$$

$$(ii) \quad X \ominus (B \cup B') = (X \ominus B) \cap (X \ominus B')$$

Chứng minh:

$$(i) \quad X \oplus (B \cup B') = (X \oplus B) \cup (X \oplus B')$$

Ta có: $B \cup B' \supseteq B$

$$X \oplus (B \cup B') \supseteq X \oplus B \quad (\text{tính gia tăng})$$

Tương tự:

$$X \oplus (B \cup B') \supseteq X \oplus B'$$

$$X \oplus (B \cup B') \supseteq (X \oplus B) \cup (X \oplus B') \quad (2.3)$$

Mặt khác,

$$\forall y \in X \oplus (B \cup B') \Rightarrow \exists x \in X \text{ sao cho } y \in (B \cup B')_x$$

$$\Rightarrow \begin{cases} y \in B_x \\ y \in B'_x \end{cases} \Rightarrow \begin{cases} y \in X \oplus B \\ y \in X \oplus B' \end{cases}$$

$$\Rightarrow y \in (X \oplus B) \cup (X \oplus B')$$

$$\Rightarrow X \oplus (B \cup B') \subseteq (X \oplus B) \cup (X \oplus B') \quad (2.4)$$

$$\text{Từ (2.3) và (2.4) ta có: } X \oplus (B \cup B') = (X \oplus B) \cup (X \oplus B')$$

$$(ii) \quad X \ominus (B \cup B') = (X \ominus B) \cap (X \ominus B')$$

Ta có: $B \cup B' \supseteq B$

$$\Rightarrow X \ominus (B \cup B') \subseteq X \ominus B \quad (\text{tính gia tăng})$$

Tương tự: $X \ominus (B \cup B') \subseteq X \ominus B'$

$$\Rightarrow X \ominus (B \cup B') \subseteq (X \ominus B) \cap (X \ominus B') \quad (2.5)$$

Mặt khác,

$$\forall x \in (X \ominus B) \cap (X \ominus B')$$

$$\text{Suy ra, } \begin{cases} x \in X \ominus B \\ x \in X \ominus B' \end{cases} \Rightarrow \begin{cases} B_x \subseteq X \\ B'_x \subseteq X \end{cases}$$

$$\Rightarrow (B \cup B')_x \subseteq X$$

$$\Rightarrow x \in X \ominus (B \cup B')$$

$$\Rightarrow X \ominus (B \cup B') \supseteq (X \ominus B) \cap (X \ominus B') \quad (2.6)$$

Từ (2.5) và (2.6) ta có: $X \ominus (B \cup B') = (X \ominus B) \cap (X \ominus B')$.

* *Ý nghĩa:*

Ta có thể phân tích các mẫu phức tạp trở thành các mẫu đơn giản thuận tiện cho việc cài đặt.

* **Mệnh đề 2.5 [Tính phân phối với phép \cap]:**

$$(X \cap Y) \ominus B = (X \ominus B) \cap (Y \ominus B)$$

Chứng minh:

$$\text{Ta có, } X \cap Y \subseteq X$$

$$\Rightarrow (X \cap Y) \ominus B \subseteq X \ominus B$$

$$\text{Tương tự: } (X \cap Y) \ominus B \subseteq Y \ominus B$$

$$\Rightarrow (X \cap Y) \ominus B \subseteq (X \ominus B) \cap (Y \ominus B) \quad (2.7)$$

Mặt khác,

$$\forall x \in (X \ominus B) \cap (Y \ominus B)$$

$$\text{Suy ra } \begin{cases} x \in X \ominus B \\ x \in Y \ominus B \end{cases} \Rightarrow \begin{cases} B_x \subseteq X \\ B_x \subseteq Y \end{cases}$$

$$\Rightarrow B_x \subseteq X \cap Y$$

$$\Rightarrow x \in (X \cap Y) \ominus B$$

$$\Rightarrow (X \cap Y) \ominus B \supseteq (X \ominus B) \cap (Y \ominus B) \quad (2.8)$$

Từ (2.7) và (2.8) ta có: $(X \cap Y) \ominus B = (X \ominus B) \cap (Y \ominus B)$.

* **Mệnh đề 2.6 [Tính kết hợp]**

$$(i) (X \oplus B) \oplus B' = X \oplus (B \oplus B')$$

$$(ii) (X \ominus B) \ominus B' = X \ominus (B \oplus B')$$

Chứng minh:

$$(i) (X \oplus B) \oplus B' = X \oplus (B' \oplus B)$$

$$\begin{aligned} \text{Ta có, } (X \oplus B) \oplus B' &= \left(\bigcup_{x \in X} B_x \right) \oplus B' \\ &= \bigcup_{x \in X} (B_x \oplus B') = \bigcup_{x \in X} (B \oplus B')_x \\ &= X \oplus (B' \oplus B) \end{aligned}$$

$$(i) (X \ominus B) \ominus B' = X \ominus (B \oplus B')$$

Trước hết ta đi chứng minh: $B'_x \subseteq X \ominus B \Leftrightarrow (B' \oplus B)_x \subseteq X$

Thật vậy, do $B'_x \subseteq X \ominus B$ nên $\forall y \in B'_x \Rightarrow y \in X \ominus B$

$$\begin{aligned} &\Rightarrow B_y \subseteq X \\ &\Rightarrow \bigcup_{y \in B'_x} B_y \subseteq X \\ &\Rightarrow (B' \oplus B)_x \subseteq X \end{aligned}$$

Mặt khác, $(B' \oplus B)_x \subseteq X \Leftrightarrow (B'_x \oplus B) \subseteq X$

$$\begin{aligned} &\Leftrightarrow \bigcup_{y \in B'_x} B_y \subseteq X \\ &\Rightarrow \forall y \in B'_x \text{ ta có } B_y \subseteq X \\ &\Rightarrow \text{hay } \forall y \in B'_x \text{ ta có } y \in X \ominus B \end{aligned}$$

Do đó, $B'_x \subseteq X \ominus B$

$$\begin{aligned} \text{Ta có, } (X \ominus B) \ominus B' &= \{x / B_x \subseteq X\} \ominus B' \\ &= \{x / B'_x \subseteq X \ominus B\} \\ &= \{x / (B' \oplus B)_x \subseteq X\} \text{ (do chứng minh ở trên)} \\ &= X \ominus (B \oplus B'). \end{aligned}$$

*** Định lý 2.1 [X bị chặn bởi các cận OPEN và CLOSE]**

Giả sử, X là một đối tượng ảnh, B là mẫu, khi đó, X sẽ bị chặn trên bởi tập CLOSE của X theo B và bị chặn dưới bởi tập OPEN của X theo B. Tức là:

$$(X \oplus B) \ominus B \supseteq X \supseteq (X \ominus B) \oplus B$$

Chứng minh:

$$\begin{aligned}
 \text{Ta có: } \forall x \in X &\Rightarrow B_x \subseteq X \oplus B \quad (\text{Vì } X \oplus B = \bigcup_{x \in X} B_x) \\
 &\Rightarrow x \in (X \oplus B) \ominus B \quad (\text{theo định nghĩa phép co}) \\
 &\Rightarrow (X \oplus B) \ominus B \supseteq X \quad (2.9)
 \end{aligned}$$

Mặt khác,

$$\begin{aligned}
 \forall y \in (X \ominus B) \oplus B, \text{ suy ra:} \\
 \exists x \in X \ominus B \text{ sao cho } y \in B_x \quad (\text{Vì } (X \ominus B) \oplus B = \bigcup_{x \in X \ominus B} B_x) \\
 \Rightarrow B_x \subseteq X \Rightarrow y \in X
 \end{aligned}$$

$$\text{Suy ra: } X \supseteq (X \ominus B) \oplus B \quad (2.10)$$

Từ (2.9) và (2.10) Ta có: $(X \oplus B) \ominus B \supseteq X \supseteq (X \ominus B) \oplus B$.

***Hệ quả 2.1 [Tính bất biến] :**

- (i) $((X \oplus B) \ominus B) \oplus B = X \oplus B$
- (ii) $((X \ominus B) \oplus B) \ominus B = X \ominus B$

Chứng minh:

$$\begin{aligned}
 \text{(i) Thật vậy, từ định lý 2.1 ta có } X &\subseteq (X \oplus B) \ominus B \\
 &\Rightarrow X \oplus B \subseteq ((X \oplus B) \ominus B) \oplus B \quad (\text{do tính chất gia tăng}) \quad (2.11)
 \end{aligned}$$

Mặt khác, cũng từ định lý 2.1 ta có $(X \ominus B) \oplus B \subseteq X \forall X$

$$\text{Do đó, thay } X \text{ bởi } X \oplus B \text{ ta có, } ((X \oplus B) \ominus B) \oplus B \subseteq X \oplus B \quad (2.12)$$

Từ (2.11) và (2.12) Ta có: $((X \oplus B) \ominus B) \oplus B = X \oplus B$

$$\begin{aligned}
 \text{(ii) Thật vậy, từ định lý 2.1 ta có } (X \ominus B) \oplus B &\subseteq X \\
 &\Rightarrow ((X \ominus B) \oplus B) \ominus B \subseteq X \ominus B \quad (\text{do tính chất gia tăng}) \quad (2.13)
 \end{aligned}$$

Mặt khác, cũng từ định lý 2.1 ta có $X \subseteq (X \oplus B) \ominus B \forall X$

$$\text{Do đó, thay } X \text{ bởi } X \ominus B \text{ ta có, } X \ominus B \subseteq ((X \ominus B) \oplus B) \ominus B \quad (2.14)$$

Từ (2.13) và (2.14) Ta có: $((X \ominus B) \oplus B) \ominus B = X \ominus B$ (đpcm).

Chương 3:

BIÊN VÀ CÁC PHƯƠNG PHÁP PHÁT HIỆN BIÊN

3.1. GIỚI THIỆU

Biên là vấn đề quan trọng trong trích chọn đặc điểm nhằm tiến tới hiểu ảnh. Cho đến nay chưa có định nghĩa chính xác về biên, trong mỗi ứng dụng người ta đưa ra các độ đo khác nhau về biên, một trong các độ đo đó là độ đo về sự thay đổi đột ngột về cấp xám. Ví dụ: Đối với ảnh đen trắng, một điểm được gọi là điểm biên nếu nó là điểm đen có ít nhất một điểm trắng bên cạnh. Tập hợp các điểm biên tạo nên biên hay đường bao của đối tượng. Xuất phát từ cơ sở này người ta thường sử dụng hai phương pháp phát hiện biên cơ bản:

Phát hiện biên trực tiếp: Phương pháp này làm nổi biên dựa vào sự biến thiên mức xám của ảnh. Kỹ thuật chủ yếu dùng để phát hiện biên ở đây là dựa vào sự biến đổi cấp xám theo hướng. Cách tiếp cận theo đạo hàm bậc nhất của ảnh dựa trên kỹ thuật Gradient, nếu lấy đạo hàm bậc hai của ảnh dựa trên biến đổi gia ta có kỹ thuật Laplace.

Phát hiện biên gián tiếp: Nếu bằng cách nào đó ta phân được ảnh thành các vùng thì ranh giới giữa các vùng đó gọi là biên. Kỹ thuật dò biên và phân vùng ảnh là hai bài toán đối ngẫu nhau vì dò biên để thực hiện phân lớp đối tượng mà khi đã phân lớp xong nghĩa là đã phân vùng được ảnh và ngược lại, khi đã phân vùng ảnh đã được phân lớp thành các đối tượng, do đó có thể phát hiện được biên.

Phương pháp phát hiện biên trực tiếp tỏ ra khá hiệu quả và ít chịu ảnh hưởng của nhiễu, song nếu sự biến thiên độ sáng không đột ngột, phương pháp tỏ ra kém hiệu quả, phương pháp phát hiện biên gián tiếp tuy khó cài đặt, song lại áp dụng khá tốt trong trường hợp này.

3.2. CÁC PHƯƠNG PHÁP PHÁT HIỆN BIÊN TRỰC TIẾP

3.2.1. Kỹ thuật phát hiện biên Gradient

Theo định nghĩa, gradient là một vectơ có các thành phần biểu thị tốc độ thay đổi giá trị của điểm ảnh, ta có:

$$\begin{cases} \frac{\partial f(x, y)}{\partial x} = f_x \approx \frac{f(x + dx, y) - f(x, y)}{dx} \\ \frac{\partial f(x, y)}{\partial y} = f_y \approx \frac{f(x, y + dy) - f(x, y)}{dy} \end{cases}$$

Trong đó, dx, dy là khoảng cách (tính bằng số điểm) theo hướng x và y.

* *Nhận xét:*

Tuy ta nói là lấy đạo hàm nhưng thực chất chỉ là mô phỏng và xấp xỉ đạo hàm bằng các kỹ thuật nhân chập (cuộn theo mẫu) vì ảnh số là tín hiệu rời rạc nên đạo hàm không tồn tại.

Ví dụ: Với dx = dy = 1, ta có:

$$\begin{cases} \frac{\partial f}{\partial x} \approx f(x + 1, y) - f(x, y) \\ \frac{\partial f}{\partial y} \approx f(x, y + 1) - f(x, y) \end{cases}$$

Do đó, mặt nạ nhân chập theo hướng x là A = $\begin{pmatrix} -1 & 1 \end{pmatrix}$

và hướng y là B = $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$

Chẳng hạn:

$$I = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 3 & 3 \\ 0 & 3 & 3 & 3 \\ 0 & 3 & 3 & 3 \end{pmatrix}$$

Ta có,

$$I \otimes A = \begin{pmatrix} 0 & 0 & 0 & * \\ 3 & 0 & 0 & * \\ 3 & 0 & 0 & * \\ * & * & * & * \end{pmatrix}; I \otimes B = \begin{pmatrix} 0 & 3 & 3 & * \\ 0 & 0 & 0 & * \\ 0 & 0 & 0 & * \\ * & * & * & * \end{pmatrix}$$

$$I \otimes A + I \otimes B = \begin{pmatrix} 0 & 0 & 0 & * \\ 3 & 0 & 0 & * \\ 3 & 0 & 0 & * \\ * & * & * & * \end{pmatrix}$$

3.2.1.1. Kỹ thuật Prewitt

Kỹ thuật sử dụng 2 mặt nạ nhập chập xấp xỉ đạo hàm theo 2 hướng x và y là:

$$H_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$
$$H_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Các bước tính toán của kỹ thuật Prewitt

+ **Bước 1**: Tính $I \otimes H_x$ và $I \otimes H_y$

+ **Bước 2**: Tính $I \otimes H_x + I \otimes H_y$

Ví dụ:

$$I = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$
$$I \otimes H_x = \begin{pmatrix} 0 & 0 & -10 & -10 & * & * \\ 0 & 0 & -15 & -15 & * & * \\ 0 & 0 & -10 & -10 & * & * \\ 0 & 0 & -5 & -5 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$
$$I \otimes H_y = \begin{pmatrix} 15 & 15 & 10 & 5 & * & * \\ 0 & 0 & 0 & 0 & * & * \\ -15 & -15 & -10 & -5 & * & * \\ -15 & -15 & -10 & -5 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

$$I \otimes H_x + I \otimes H_y = \begin{pmatrix} 15 & 15 & 0 & -5 & * & * \\ 0 & 0 & -15 & -15 & * & * \\ -15 & -15 & -20 & -15 & * & * \\ -15 & -15 & -15 & -10 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

3.2.1.2. Kỹ thuật Sobel

Tương tự như kỹ thuật Prewitt kỹ thuật Sobel sử dụng 2 mặt nạ nhân chập theo 2 hướng x, y là:

$$H_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$H_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Các bước tính toán tương tự Prewitt

+ **Bước 1**: Tính $I \otimes H_x$ và $I \otimes H_y$

+ **Bước 2**: Tính $I \otimes H_x + I \otimes H_y$

3.2.1.3. Kỹ thuật la bàn

Kỹ thuật sử dụng 8 mặt nạ nhân chập theo 8 hướng $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$

$$H_1 = \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \quad H_2 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

$$H_3 = \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix} \quad H_4 = \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix}$$

$$H_5 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix} \quad H_6 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix}$$

$$H_7 = \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix} \quad H_8 = \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix}$$

Các bước tính toán thuật toán La bàn

+ **Bước 1**: Tính $I \otimes H_i$; $i = 1, 8$

+ **Bước 2**: $\sum_{i=1}^8 I \otimes H_i$

3.2.2. Kỹ thuật phát hiện biên Laplace

Các phương pháp đánh giá gradient ở trên làm việc khá tốt khi mà độ sáng thay đổi rõ nét. Khi mức xám thay đổi chậm, miền chuyển tiếp trải rộng, phương pháp cho hiệu quả hơn đó là phương pháp sử dụng đạo hàm bậc hai Laplace.

Toán tử Laplace được định nghĩa như sau:

Ta có:
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right) \approx \frac{\partial}{\partial x} (f(x+1, y) - f(x, y))$$

$$\begin{aligned} &\approx [f(x+1, y) - f(x, y)] - [f(x, y) - f(x-1, y)] \\ &\approx f(x+1, y) - 2f(x, y) + f(x-1, y) \end{aligned}$$

Tương tự,

$$\frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial y} \right) \approx \frac{\partial}{\partial y} (f(x, y+1) - f(x, y))$$

$$\begin{aligned} &\approx [f(x, y+1) - f(x, y)] - [f(x, y) - f(x, y-1)] \\ &\approx f(x, y+1) - 2f(x, y) + f(x, y-1) \end{aligned}$$

Vậy: $\nabla^2 f = f(x+1, y) + f(x, y+1) - 4f(x, y) + f(x-1, y) + f(x, y-1)$

Dẫn tới:

$$H = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Trong thực tế, người ta thường dùng nhiều kiểu mặt nạ khác nhau để xấp xỉ rời rạc đạo hàm bậc hai Laplace. Dưới đây là ba kiểu mặt nạ thường dùng:

$$H_1 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad H_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad H_3 = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

$$\text{VD:} \quad I = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

3.3. PHÁT HIỆN BIÊN GIÁN TIẾP

3.3.1 Một số khái niệm cơ bản

*Ảnh và điểm ảnh

Ảnh số là một mảng số thực 2 chiều (I_{ij}) có kích thước ($M \times N$), trong đó mỗi phần tử I_{ij} ($i = 1, \dots, M; j = 1, \dots, N$) biểu thị mức xám của ảnh tại (i, j) tương ứng.

Ảnh được gọi là ảnh nhị phân nếu các giá trị I_{ij} chỉ nhận giá trị 0 hoặc 1.

Ở đây ta chỉ xét tới ảnh nhị phân vì ảnh bất kỳ có thể đưa về dạng nhị phân bằng kỹ thuật phân ngưỡng. Ta ký hiệu \mathfrak{S} là tập các điểm vùng (điểm đen) và $\bar{\mathfrak{S}}$ là tập các điểm nền (điểm trắng).

*Các điểm 4 và 8-láng giềng

Giả sử (i, j) là một điểm ảnh, các điểm 4-láng giềng là các điểm kề trên, dưới, trái, phải của (i, j) :

$$N_4(i, j) = \{(i', j') : |i - i'| + |j - j'| = 1\},$$

và những điểm 8-láng giềng gồm:

$$N_8(i, j) = \{(i', j') : \max(|i - i'|, |j - j'|) = 1\}.$$

Trong Hình 1.2 biểu diễn ma trận 8 láng giềng kề nhau, các điểm P_0, P_2, P_4, P_6 là các 4-láng giềng của điểm P , còn các điểm $P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7$ là các 8-láng giềng của P .

P_3	P_2	P_1
P_4	P	P_0
P_5	P_6	P_7

Hình 1.3. Ma trận 8-láng giềng kề nhau

*Đối tượng ảnh

Hai điểm $P_s, P_e \in E$, $E \subseteq \mathfrak{T}$ hoặc $\overline{\mathfrak{T}}$ được gọi là 8-liên thông (hoặc 4-liên thông) trong E nếu tồn tại tập các điểm được gọi là **đường đi** $(i_0, j_0) \dots (i_n, j_n)$ sao cho $(i_0, j_0) = P_s$, $(i_n, j_n) = P_e$, $(i_r, j_r) \in E$ và (i_r, j_r) là 8-láng giềng (hoặc 4-láng giềng tương ứng) của (i_{r-1}, j_{r-1}) với $r = 1, 2, \dots, n$

Nhận xét: Quan hệ ***k-liên thông trong E*** ($k=4,8$) là một quan hệ phản xạ, đối xứng và bắc cầu. Bởi vậy đó là một quan hệ tương đương. Mỗi lớp tương đương được gọi là một thành phần k -liên thông của ảnh. Về sau ta sẽ gọi mỗi thành phần k -liên thông của ảnh là một đối tượng ảnh.

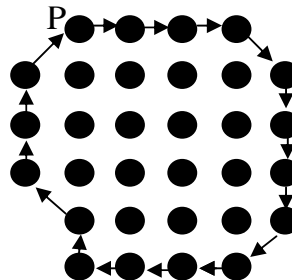
3.3.2. Chu tuyến của một đối tượng ảnh

Định nghĩa 3.1: [Chu tuyến]

Chu tuyến của một đối tượng ảnh là dãy các điểm của đối tượng ảnh P_1, \dots, P_n sao cho P_i và P_{i+1} là các 8-láng giềng của nhau ($i=1, \dots, n-1$) và P_1 là 8-láng giềng của P_n , $\forall i \exists Q$ không thuộc đối tượng ảnh và Q là 4-láng giềng của P_i (hay nói cách khác $\forall i$ thì P_i là biên 4). Kí hiệu $\langle P_1 P_2 \dots P_n \rangle$.

Tổng các khoảng cách giữa hai điểm kế tiếp của chu tuyến là độ dài của chu tuyến và kí hiệu $Len(C)$ và hướng $P_i P_{i+1}$ là hướng chắn nếu P_i và P_{i+1} là các 4 – láng giềng (trường hợp còn lại thì $P_i P_{i+1}$ là hướng lẻ).

Hình 3.1 dưới đây biểu diễn chu tuyến của ảnh, trong đó, P là điểm khởi đầu chu tuyến.



Hình 3.1. Ví dụ về chu tuyến của đối tượng ảnh

Định nghĩa 3.2 [Chu tuyến đối ngẫu]

Hai chu tuyến $C = \langle P_1 P_2 \dots P_n \rangle$ và $C^\perp = \langle Q_1 Q_2 \dots Q_m \rangle$ được gọi là đối ngẫu của nhau nếu và chỉ nếu $\forall i \exists j$ sao cho:

- (i) P_i và Q_j là 4-láng giềng của nhau.
- (ii) Các điểm P_i là vùng thì Q_j là nền và ngược lại.

Định nghĩa 3.3 [Chu tuyến ngoài]

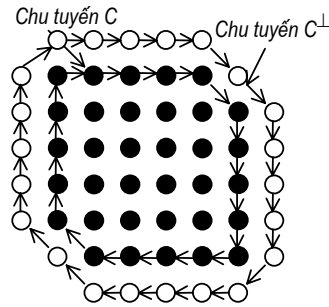
Chu tuyến C được gọi là chu tuyến ngoài (Hình 3.2a) nếu và chỉ nếu

- (i) Chu tuyến đối ngẫu C^\perp là chu tuyến của các điểm nền
- (ii) Độ dài của C nhỏ hơn độ dài C^\perp

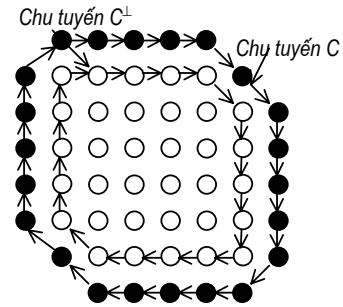
Định nghĩa 3.4 [Chu tuyến trong]

Chu tuyến C được gọi là chu tuyến trong (Hình 3.2b) nếu và chỉ nếu:

- (i) Chu tuyến đối ngẫu C^\perp là chu tuyến của các điểm nền
- (ii) Độ dài của C lớn hơn độ dài C^\perp



a) Chu tuyến ngoài



b) Chu tuyến trong

Hình 3.2. Chu tuyến trong, chu tuyến ngoài

Định nghĩa 3.5 [Điểm trong và điểm ngoài chu tuyến]

Giả sử $C = \langle P_1 P_2 \dots P_n \rangle$ là chu tuyến của một đối tượng ảnh và P là một điểm ảnh. Khi đó:

- (i) Nếu nửa đường thẳng xuất phát từ P sẽ cắt chu tuyến C tại số lẻ lần, thì P được gọi là điểm trong chu tuyến C và kí hiệu $in(P, C)$
- (ii) Nếu $P \notin C$ và P không phải là điểm trong của C , thì P được gọi là điểm ngoài chu tuyến C và kí hiệu $out(P, C)$.

Bổ đề 3.1 [Chu tuyến đối ngẫu]

Giả sử $E \subseteq \mathfrak{I}$ là một đối tượng ảnh và $C = \langle P_1 P_2 \dots P_n \rangle$ là chu tuyến của E , $C^\perp = \langle Q_1 Q_2 \dots Q_m \rangle$ là chu tuyến đối ngẫu tương ứng. Khi đó:

- (i) Nếu C là chu tuyến trong thì $in(Q_i, C) \forall i (i=1, \dots, m)$

(ii) Nếu C là chu tuyến ngoài thì $\text{in}(P_i, C^\perp) \forall i (i=1, \dots, n)$

Bổ đề 3.2 [Phần trong/ngoài của chu tuyến]

Giả sử $E \subseteq \mathfrak{S}$ là một đối tượng ảnh và C là chu tuyến của E . Khi đó:

- (i) Nếu C là chu tuyến ngoài thì $\forall x \in E$ sao cho $x \notin C$, ta có $\text{in}(x, C)$
- (ii) Nếu C là chu tuyến trong thì $\forall x \in E$ sao cho $x \notin C$, ta có $\text{out}(x, C)$

Định lý 3.1 [Tính duy nhất của chu tuyến ngoài]

Giả sử $E \subseteq \mathfrak{S}$ là một đối tượng ảnh và C_E là chu tuyến ngoài của E . Khi đó C_E là duy nhất.

3.3.3. Thuật toán dò biên tổng quát

Biểu diễn đối tượng ảnh theo chu tuyến thường dựa trên các kỹ thuật dò biên. Có hai kỹ thuật dò biên cơ bản. Kỹ thuật thứ nhất xét ảnh biên thu được từ ảnh vùng sau một lần duyệt như một đồ thị, sau đó áp dụng các thuật toán duyệt cạnh đồ thị. Kỹ thuật thứ hai dựa trên ảnh vùng, kết hợp đồng thời quá trình dò biên và tách biên. Ở đây ta quan tâm cách tiếp cận thứ hai.

Trước hết, giả sử ảnh được xét chỉ bao gồm một vùng ảnh 8-liên thông \mathfrak{S} , được bao bọc bởi một vành đai các điểm nền. Dễ thấy \mathfrak{S} là một vùng 4-liên thông chỉ là một trường riêng của trường hợp trên.

Về cơ bản, các thuật toán dò biên trên một vùng đều bao gồm các bước sau:

- Xác định điểm biên xuất phát
- Dự báo và xác định điểm biên tiếp theo
- Lặp bước 2 cho đến khi gặp điểm xuất phát

Do xuất phát từ những tiêu chuẩn và định nghĩa khác nhau về điểm biên, và quan hệ liên thông, các thuật toán dò biên cho ta các đường biên mang các sắc thái rất khác nhau.

Kết quả tác động của toán tử dò biên lên một điểm biên r_i là điểm biên r_{i+1} (8-láng giềng của r_i). Thông thường các toán tử này được xây dựng như một hàm đại số Boolean trên các 8-láng giềng của r_i . Mỗi cách xây dựng các toán tử đều phụ thuộc vào định nghĩa quan hệ liên thông và điểm biên. Do đó sẽ gây khó khăn cho việc khảo sát các tính chất của đường biên. Ngoài ra, vì mỗi bước dò biên đều phải kiểm tra tất cả các 8-láng giềng của mỗi điểm nên thuật toán thường kém hiệu quả. Để khắc phục các hạn chế trên, thay vì sử dụng một điểm biên ta sử dụng cặp điểm biên (một thuộc \mathfrak{S} , một thuộc $\overline{\mathfrak{S}}$), các cặp điểm này tạo nên tập nền vùng, kí hiệu là NV và phân tích toán tử dò biên thành 2 bước:

- Xác định cặp điểm nền vùng tiếp theo.
- Lựa chọn điểm biên

Trong đó bước thứ nhất thực hiện chức năng của một ánh xạ trên tập NV lên NV và bước thứ hai thực hiện chức năng chọn điểm biên.

Thuật toán dò biên tổng quát

Bước 1: Xác định cặp nền-vùng xuất phát

Bước 2: Xác định cặp nền-vùng tiếp theo

Bước 3: Lựa chọn điểm biên vùng

Bước 4: Nếu gặp lại cặp xuất phát thì dừng, nếu không quay lại bước 2.

Việc xác định cặp nền-vùng xuất phát được thực hiện bằng cách duyệt ảnh lần lượt từ trên xuống dưới và từ trái qua phải rồi kiểm tra điều kiện lựa chọn cặp nền-vùng. Do việc chọn điểm biên chỉ mang tính chất quy ước, nên ta gọi ánh xạ xác định cặp nền-vùng tiếp theo là toán tử dò biên.

Định nghĩa 3.6 [Toán tử dò biên]

Giả sử T là một ánh xạ như sau: $T: NV \rightarrow NV$

$$(b,r) \mapsto (b',r')$$

Gọi T là một toán tử dò biên cơ sở nếu nó thoả mãn điều kiện: b',r' là các 8-láng giềng của r.

Giả sử $(b,r) \in NV$; gọi $K(b,r)$ là hàm chọn điểm biên. Biên của một dạng \mathfrak{S} có thể định nghĩa theo một trong ba cách:

- Tập những điểm thuộc \mathfrak{S} có mặt trên NV, tức là $K(b,r)=r$
- Tập những điểm thuộc $\overline{\mathfrak{S}}$ có trên NV, tức là $K(b,r)=b$
- Tập những điểm ảo nằm giữa cặp nền-vùng, tức là $K(b,r)$ là những điểm nằm giữa hai điểm b và r.

Cách định nghĩa thứ ba tương ứng mỗi cặp nền-vùng với một điểm biên. Còn đối với cách định nghĩa thứ nhất và thứ hai một số cặp nền-vùng có thể có chung một điểm biên. Bởi vậy, quá trình chọn điểm biên được thực hiện như sau:

$i:= 1; (b_i,r_i):= (b_o,r_o);$

While $K(b_i,r_i) \neq K(b_n,r_n)$ **and** $i \leq 8$ **do**

Begin $(b_{i+1},r_{i+1})= T(b_i,r_i); i:= i+1;$ **End;**

Điều kiện dừng

Cặp nền-vùng thứ n trùng với cặp nền vùng xuất phát: $(b_n,r_n)= (b_o,r_o)$

* Xác định cặp nền – vùng xuất phát

Cặp nền vùng xuất phát được xác định bằng cách duyệt ảnh lần lượt từ trên xuống dưới và từ trái sang phải điểm đem đầu tiên gặp được cùng với điểm trắng trước đó (theo hướng 4) để tạo nên cặp nền vùng xuất phát.

* Xác định cặp nền vùng tiếp theo

Đầu vào: pt, dir

Ví dụ: (3, 2) 4

Point orient [] = {(1,0);(1,-1);(0,-1);(-1,-1);(-1,0);(-1,1);(0,1);(1,1)};

//Hàm tìm hướng có điểm đen gần nhất

BYTE GextNextDir(POINT pt, BYTE dir)

{

 BYTE pdir = (dir + 7) % 8;

 do {

 if (getpixel(pt.x + orient[pdir].x, pt.y + orient[pdir].y) == BLACK)

 return pdir;

 pdir = (pdir + 7) % 8;

 } while (pdir != dir);

 return ERR; //Điểm cô lập

}

//Gán giá trị cho bước tiếp theo

pdir = GetNextDir(pt, dir);

if (pdir == ERR) //Kiểm tra có là điểm cô lập không?

 return ERR; //Điểm cô lập

pt.x = pt.x + orient[pdir].x;

pt.y = pt.y + orient[pdir].y;

Để tính giá trị cho hướng tiếp theo ta lập bảng dựa trên giá trị pdir đã tính được trước đó theo các khả năng có thể xảy ra:

pdir	Điểm trắng trước đó	Trắng so với đen mới
0	1	2
1	2	4
2	3	4
3	4	6
4	5	6
5	6	0
6	7	0
7	0	2

⇒ Do đó công thức để tính hướng tiếp theo sẽ là :

$$\text{dir} = ((\text{pdir} + 3) / 2 * 2) \% 8 ;$$

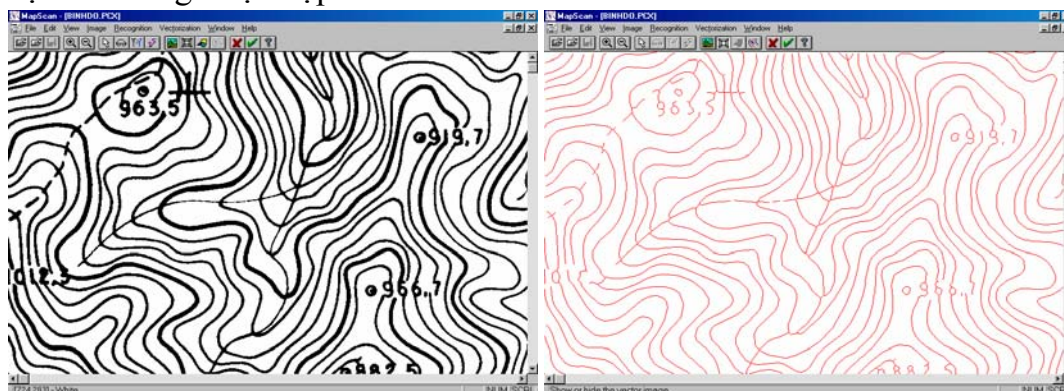
Chương 4:

XƯƠNG VÀ CÁC KỸ THUẬT TÌM XƯƠNG

4.1. 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 4.1. 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

4.2. TÌM XƯƠNG DỰA TRÊN LÀM MẢNH

4.2.1. 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 vectơ 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 tuỳ 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.

4.2.2. 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.

- 1°. 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.
- 2°. Thuật toán làm mảnh của Toumazet bảo toàn tất cả các điếm cụt 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
- 3°. 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 điếm là gây đứt nét, xương tạo ra là xương giả (có độ dày là 2 phần tử ảnh).
- 4°. Thuật toán làm mảnh của N.J.Naccache và R.Shinghal. Thuật toán có ưu điếm là nhanh, xương tạo ra có khả năng khôi phục ảnh ban đầu của đối tượng. Nhược điếm chính của thuật toán là rất nhạy với nhiễu, xương nhận được phản ánh cấu trúc của đối tượng thấp.
- 5°. 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 điếm là xương tạo ra là xương 4-liên thông và xoá mất một số cấu hình nhỏ.
- 6°. 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 điếm chính của thuật toán là tốc độ chậm.
- 7°. 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 điếm là xoá hoàn toàn một số cấu hình nhỏ, xương tạo ra là xương 4-liên thông.

4.3. TÌM XƯƠNG KHÔNG DỰA TRÊN LÀM MẢNH

Để tách được xương của đối tượng có thể sử dụng đường biên của đối tượng. Với điếm p bất kỳ trên đối tượng, ta bao nó bởi một đường biên. Nếu như có nhiều điếm biên có cùng khoảng cách ngắn nhất tới p thì p nằm trên trục trung vị. Tập tất cả các điếm 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.

4.3.1. 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 $H(P_i, P_j)$:

$$V(P_i) = \bigcap H(P_i, P_j) \quad i \neq j \quad (i = 1, \dots, n) \quad (4.1)$$

Định nghĩa 4.1 [Đa giác/Sơ đồ Voronoi]

Sơ đồ Voronoi của Ω là hợp của tất cả các $V(P_i)$

$$\text{Vor}(\Omega) = \bigcup V(P_i) \quad P_i \in \Omega \quad (\text{là một đa giác}) \quad (4.2)$$

Định nghĩa 4.2 [Đa giác Voronoi tổng quát]

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

$$\begin{aligned} V(U) &= \{P \mid \exists v \in U, \forall w \in \Omega \setminus U : d(P, v) < d(P, w)\} \\ &= \bigcup V(P_i) \quad P_i \in U \end{aligned} \quad (4.3)$$

4.3.2. Trục trung vị Voronoi rời rạc

Định nghĩa 4.3 [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 $d(.,.)$ như sau:

$$\forall (x, y) \in S: \text{map}(x, y) = \min d[(x, y), (x_i, y_i)] \quad (4.4)$$

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 là bản đồ khoảng cách của S .

Chú ý: Nếu hàm khoảng cách $d(.,.)$ là khoảng cách Euclide, thì phương trình (4.4) 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 4.4 [Tập các điểm biên sinh]

Cho $\text{map}(x, y)$ là khoảng cách ngắn nhất từ (x, y) đến biên (theo định nghĩa 4.3). 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:

$$\hat{B}(S) = \cup \text{map}^{-1}(x, y), (x, y) \in S \quad (4.5)$$

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:

$$\hat{B}(S) = \{B_1(S), \dots, B_N(S)\} \quad (4.6)$$

Định nghĩa 4.5 [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 :

$$\text{DVMA}(\hat{B}(S)) = \text{Vor}(\hat{B}(S)) \cap S \quad (4.7)$$

4.3.3. Xương Voronoi rời rạc

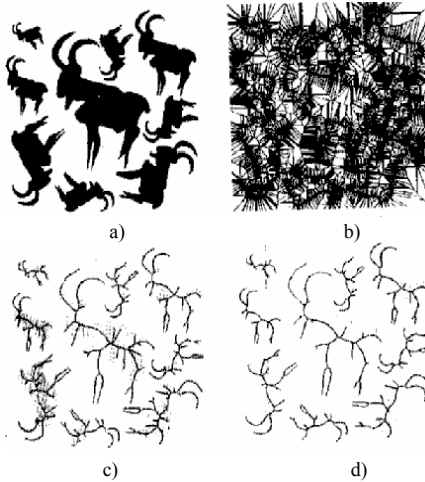
Định nghĩa 4.6 [Xương Voronoi rời rạc - DiscreteVoronoi Skeleton]

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

$$\text{Ske}^{\text{DVMA}}(\hat{B}(S), T) = \{(x, y) \mid (x, y) \in \text{DVMA}(\hat{B}(S)), \Psi(x, y) > T\} \quad (4.8)$$

Ψ : là hàm hiệu chỉnh.

Dễ thấy nếu ngưỡng T càng lớn thì càng thì số lượng điểm tham gia trong xương Voronoi càng ít (Hình 4.2).



Hình 4.2. 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, $T=18.0$

4.3.4. Thuật toán tìm xương

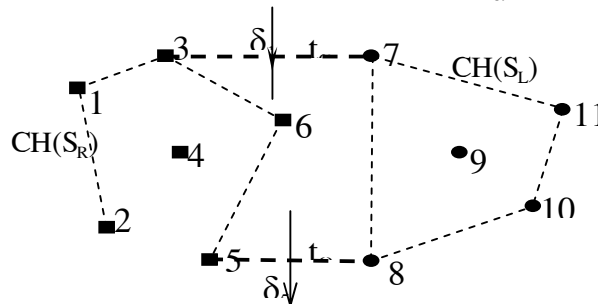
Trong mục này sẽ trình bày ý tưởng cơ bản của thuật toán tìm xương và mô tả bằng ngôn ngữ tựa Pascal.

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 dấp của thuật toán chia để trị.

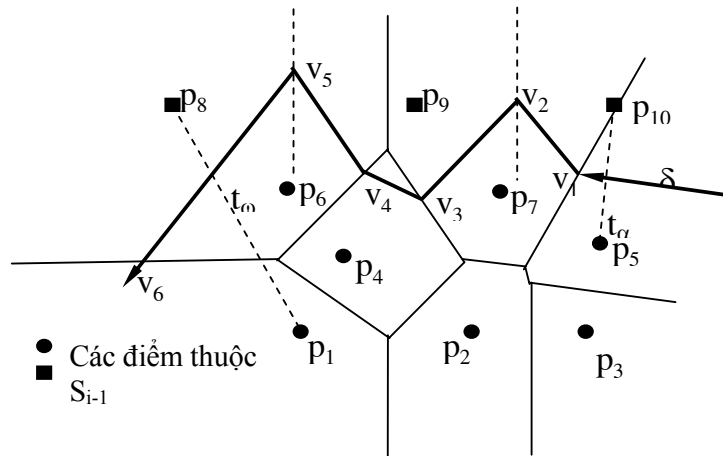
Hình 4.3 minh họa ý tưởng của thuật toán này. Mười một điểm biên được chia thành hai phần (bên trái: 1- 6, bên phải: 7-11) bởi đường gấp khúc δ , và hai sơ đồ Voronoi tương ứng $Vor(S_L)$ và $Vor(S_R)$. Để thu được sơ đồ Voronoi $Vor(S_L \cup S_R)$, ta thực hiện việc trộn hai sơ đồ trên và xác định lại 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 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 4.3. 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 và $\text{Vor}(S_L)$; 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 hoạ 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 4.4. Minh hoạ 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,

$0 \leq i \leq i_{MAX}$,

$\text{Vor}(S_i)$ 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 ;

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

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 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 5:

CÁC KỸ THUẬT HẬU XỬ LÝ

5.1. RÚT GỌN SỐ LƯỢNG ĐIỂM BIỂU DIỄN

5.1.1. Giới thiệu

Rút gọn số lượng điểm biểu diễn là kỹ thuật thuộc phần hậu xử lý. Kết quả của phần dò biên hay trích xương thu được 1 dãy các điểm liên tiếp. Vấn đề đặt ra là hiệu có thể bỏ bớt các điểm thu được để giảm thiểu không quan lưu trữ và thuận tiện cho việc đối sách hay không.

Bài toán:

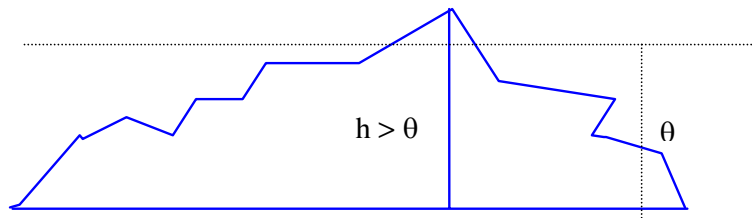
Cho đường cong gồm n điểm trong mặt phẳng $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$. Hãy bỏ bớt 1 số điểm thuộc đường cong sao cho đường cong mới nhận được là $(X_{i1}, Y_{i1}), (X_{i2}, Y_{i2}) \dots (X_{im}, Y_{im})$ “gần giống” với đường cong ban đầu.

* Một số độ đo “gần giống”

- + Chiều dài (chiều rộng) của hình chữ nhật nhá nhất chứa đường cong
- + Khoảng cách lớn nhất từ đường cong đến đoạn thẳng nối 2 đầu mút của đường cong
- + Tỷ lệ giữa chiều dài và chiều rộng của hình chữ nhật nhá nhất chứa đường con
- + Số lần đường cong cắt đoạn thẳng nối 2 đầu mút

5.1.2. Thuật toán Douglas Peucker

5.1.2.1. Ý tưởng



Hình 5.1. Đơn giản hóa đường cong theo thuật toán Douglas Peucker

Ý tưởng cơ bản của thuật toán Douglas-Peucker là xét xem khoảng cách lớn nhất từ đường cong tới đoạn thẳng nối hai đầu mút đường cong (xem Hình 5.1) có lớn hơn ngưỡng θ không. Nếu điều này đúng thì điểm xa nhất được giữ lại làm điểm chia đường cong và thuật toán được thực hiện

tương tự với hai đường cong vừa tìm được. Trong trường hợp ngược lại, kết quả của thuật toán đơn giản hoá là hai điểm đầu mút của đường cong.

Thuật toán Douglas-Peucker:

- Bước 1: Chọn ngưỡng θ .
- Bước 2: Tìm khoảng cách lớn nhất từ đường cong tới đoạn thẳng nối hai đầu đoạn đường cong h .
- Bước 3: Nếu $h \leq \theta$ thì dừng.
- Bước 4: Nếu $h > \theta$ thì giữ lại điểm đạt cực đại này và quay trở lại bước 1.

Nhận xét: Thuật toán này tỏ ra thuận lợi đối với các đường cong thu nhận được mà gốc là các đoạn thẳng, phù hợp với việc đơn giản hoá trong quá trình véc tơ các bản vẽ kỹ thuật, sơ đồ thiết kế mạch in v.v..

5.1.2.2. Chương trình

//Hàm tính đường cao từ đỉnh đến đoạn thẳng nối hai điểm dau, cuoi

float Tinhduongcao (POINT dau, POINT cuoi, POINT dinh)

```
{  
    float h;  
    || tính đường cao  
    return h ;  
}
```

//Hàm đệ quy nhằm đánh dấu loại bỏ các điểm trong đường cong

void DP Simple(POINT *pLINE,int dau,int cuoi,BOOL *chiso,float θ)

```
{  
    int i, index = dau;  
    float h, hmax = 0;  
    for(i = dau + 1; i < cuoi; i++)  
    {  
        h= Tinhduongcao(pLINE[dau], pLINE[cuoi]; pLINE[i]);  
        if(h > hmax)  
        {  
            hmax = h;  
            index = i;  
        }  
    }  
}
```

```

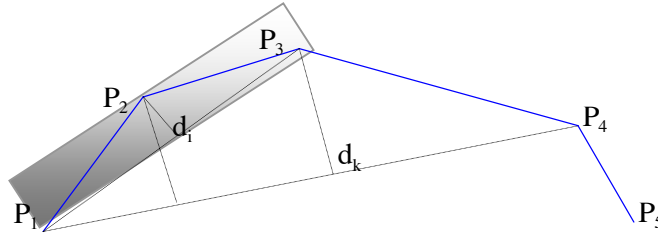
    }
}
if(hmax ≤  $\theta$ )
    for(i= dau + 1; i < cuoi, i++)
        chiso[i] = FALSE;
else
{
    DPSimple(PLINE, dau, index, chiso,  $\theta$ );
    DPSimple(PLINE, index, cuoi, chiso,  $\theta$ ) ;
}
}
//Hàm rút gọn số lượng điểm DouglasPeucker
int DouglasPeucker(POINT *pLINE, int n, float  $\theta$ )
{
    int    i, j;
    BOOL chiso [MAX_PT];
    for(i = 0; i < m; i++) //Tất cả các điểm được giữ lại
        chiso[i] = TRUE;
    DPSimple(pLINE, 0, n - 1, chiso,  $\theta$ );
    for(i = j = 0; i < n; i++)
        if (chiso [i] ==TRUE)
            pLINE[j++] = pLINE[i];
    return j;
}

```

5.1.3. Thuật toán Band width

5.1.3.1. Ý tưởng

Trong thuật toán Band Width, ta hình dung có một dải băng di chuyển từ đầu mút đường cong dọc theo đường cong sao cho đường cong nằm trong dải băng đó cho đến khi có điểm thuộc đường cong chạm vào biên của dải băng, điểm này sẽ được giữ lại. Quá trình này được thực hiện với phần còn lại của đường cong bắt đầu từ điểm vừa tìm được cho đến khi hết đường cong. Cụ thể như sau:



Hình 5.2. Đơn giản hóa đường cong với thuật toán Band Width

Bắt đầu bằng việc xác định điểm đầu tiên trên đường cong và coi đó như là một điểm chốt (P_1). Điểm thứ ba (P_3) được coi là điểm động. Điểm giữa điểm chốt và điểm động (P_2) là điểm trung gian. Ban đầu khoảng cách từ điểm trung gian đến đoạn thẳng nối điểm chốt và điểm động được tính toán và kiểm tra. Nếu khoảng cách tính được này nhỏ hơn một ngưỡng θ cho trước thì điểm trung gian có thể bỏ đi, tiến trình tiếp tục với điểm chốt là điểm chốt cũ, điểm trung gian là điểm động cũ và điểm động là điểm kế tiếp sau điểm động cũ. Trong trường hợp ngược lại, khoảng cách tính được lớn hơn ngưỡng θ cho trước thì điểm trung gian sẽ được giữ lại, tiến trình tiếp tục với điểm chốt là điểm trung gian, điểm trung gian là điểm động cũ và điểm động là điểm kế tiếp sau điểm động cũ. Tiến trình được lặp cho đến hết đường cong (Hình 5.2 minh họa thuật toán Band-Width).

Thuật toán Band-Width:

- Bước 1: Xác định điểm đầu tiên trên đường cong và coi đó như là một điểm chốt (P_1). Điểm thứ ba (P_3) được coi là điểm động. Điểm giữa điểm chốt và điểm động (P_2) là điểm trung gian.
- Bước 2: Tính khoảng cách từ điểm trung gian đến đoạn thẳng nối hai điểm chốt và điểm động.
- Bước 3: Kiểm tra khoảng cách tìm được nếu nhỏ hơn một ngưỡng θ cho trước thì điểm trung gian có thể bỏ đi. Trong trường hợp ngược lại điểm chốt chuyển đến điểm trung gian.
- Bước 4: Chu trình được lặp lại thì điểm trung gian được chuyển đến điểm động và điểm kế tiếp sau điểm động được chỉ định làm điểm động mới..

Nhận xét: Thuật toán này tăng tốc độ trong trường hợp đường ống chứa nhiều điểm, điều đó có nghĩa là độ lệch giữa các điểm trong đường thẳng là nhỏ, hay độ dày nét của đường được vectơ hoá là mảnh.

5.1.3.2. Chương trình

```
//Hàm tính đường cao từ đỉnh đến đoạn thẳng nối hai điểm dau, cuoi
float Tinhduongcao(POINT dau, POINT cuoi, POINT dinh)
{
    float h;
    || tính đường cao
    return h ;
}
//Hàm đệ quy nhằm đánh dấu loại bỏ các điểm trong đường cong
void BWSimple(POINT *pLINE, int chot, int tg, BOOL *chiso,
              float  $\theta$ , int n)
{
    if(Tinhduongcao(pLINE[chot], pLINE[tg+1], pLINE[tg])  $\leq$   $\theta$ )
        chiso[tg] = 0;
    else
        chot = tg;
    tg = tg + 1
    if(tg < n - 1)
        BWSimple (pLINE, chot, tg, chiso,  $\theta$ , n) ;
}
//Hàm rút gọn số lượng điểm BandWidth
int BandWidth(POINT *pLINE, int n, float  $\theta$ )
{
    int    i, j;
    BOOL chiso [MAX_PT];
    for (i = 0; i < n; i++)
        chiso[i]= TRUE; //Tất cả các điểm được giữ lại
    BWSimple(pLINE, 0, 1, chiso,  $\theta$ , n);
    for(i= j= 0; i < n; i++)
        if(chiso [i]== TRUE)
```



```

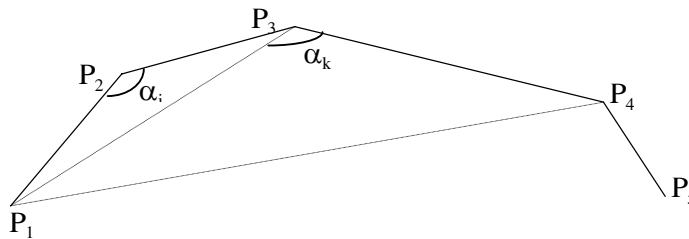
        pLINE [j ++1] = pLINE [i];
    return j;
}

```

5.1.4. Thuật toán Angles

5.1.4.1. Ý tưởng

Tương tự như thuật toán Band Width nhưng thay việc tính toán khoảng cách bởi tính góc. Cụ thể thuật toán bắt đầu với điểm đầu đường cong (P_1) là điểm chốt.



Hình 5.3. Đơn giản hóa đường cong với thuật toán Angles

Điểm thứ 3 của đường cong (P_3) là điểm động, điểm giữa điểm chốt và điểm động (P_2) là điểm trung gian

Góc tạo bởi điểm chốt, trung gian, động với điểm trung gian là đỉnh việc tính toán và kiểm tra

Nếu thì điểm trung gian có thể bỏ đi trong trường hợp ngược lại điểm chốt sẽ là điểm trung gian cũ và quá trình lặp với điểm trung gian là điểm động cũ, điểm động mới là điểm kế tiếp sau điểm động cũ. Tiến trình thực hiện cho đến hết đường cong.

5.1.4.2. Chương trình

//Hàm tính đường cao từ đỉnh đến đoạn thẳng nối hai điểm dau, cuoi

float Tinhgoc(POINT dau, POINT cuoi, POINT dinh)

```
{
```

```
    float  $\theta$ ;
```

```
    || tinhgoc (tự viết)
```

```
    return  $\theta$ ;
```

```
}
```

//Hàm đệ quy nhằm đánh dấu loại bỏ các điểm trong đường cong

void ALSimple(POINT *pLINE,int chot,int tg,BOOL *chiso,float θ ,int n)

```
{
```

```

    if(Tinhgoc(pLINE[chot], pLINE[tg], pLINE[tg+1]) >  $\theta$ )
        chiso[tg] = FALSE;
    else
        chot = tg;
    tg = tg + 1;
    if(tg < n - 1)
        ALSimple(pLINE, chot, tg, chiso,  $\theta$ , n);
}
//Hàm rút gọn số lượng điểm Angles
int Angles(POINT *pLINE, int n, float  $\theta$ )
{
    int i, j, chiso [MAX];
    for (i = 0; i < n; i++) //Tất cả các điểm được giữ lại
        chiso[i]= TRUE;
    ALSiple (PLINE, 0, 1 chiso,  $\theta$ , n) ;
    for (i = j = 0; i < n; i++)
        if (chiso ==TRUE)
            pLINE[j++]= pLINE [i];
    return j;
}

```

** Chú ý:*

Với $\theta = 0$ thuật toán DouglasPeucker và BandWidth sẽ bỏ đi các điểm giữa thẳng hàng. Thuật toán Angles phải có $\theta = 180^\circ$ để bỏ đi các điểm giữa thẳng hàng.

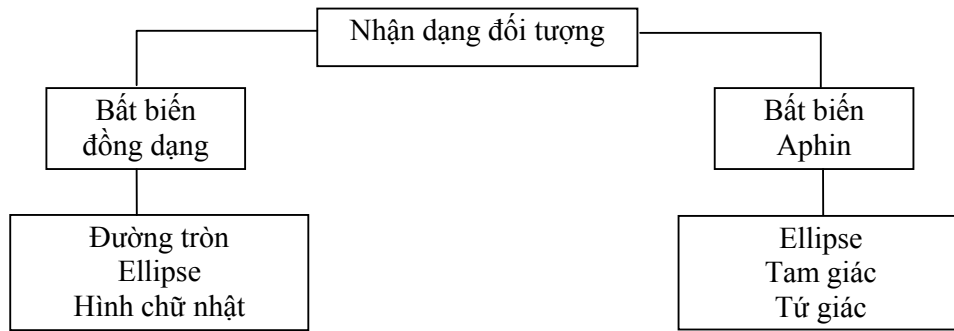
5.2. XẤP XỈ ĐA GIÁC BỞI CÁC HÌNH CƠ SỞ

Các đối tượng hình học được phát hiện thường thông qua các kỹ thuật dò biên, kết quả tìm được này là các đường biên xác định đối tượng. Đó là, một dãy các điểm liên tiếp đóng kính, sử dụng các thuật toán đơn giản hoá như Douglas Peucker, Band Width, Angle v.v.. ta sẽ thu được một polyline hay nói khác đi là thu được một đa giác xác định đối tượng dấu. Vấn đề là ta cần phải xác định xem đối tượng có phải là đối tượng cần tách hay không? Như ta đã biết một đa giác có thể có hình dạng tựa như một hình cơ

sở, có thể có nhiều cách tiếp cận xấp xỉ khác nhau. Cách xấp xỉ dựa trên các đặc trưng cơ bản sau:

Đặc trưng toàn cục: Các mô men thống kê, số đo hình học như chu vi, diện tích, tập tối ưu các hình chữ nhật phủ hay nội tiếp đa giác v.v..

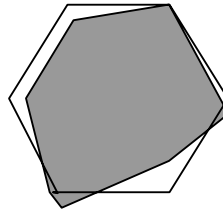
Đặc trưng địa phương: Các số đo đặc trưng của đường cong như góc, điểm lồi, lõm, uốn, cực trị v.v..



Hình 5.4. Sơ đồ phân loại các đối tượng theo bất biến

Việc xấp xỉ tỏ ra rất có hiệu quả đối với một số hình phẳng đặc biệt như tam giác, đường tròn, hình chữ nhật, hình vuông, hình ellipse, hình tròn và một đa giác mẫu.

5.2.1 Xấp xỉ đa giác theo bất biến đồng dạng



Hình 5.5. Xấp xỉ đa giác bởi một đa giác mẫu

Một đa giác với các đỉnh V_0, \dots, V_{m-1} được xấp xỉ với đa giác mẫu U_0, \dots, U_{n-1} với độ đo xấp xỉ như sau:

$$E(V, U) = \min_{0 \leq d \leq m-1} \sqrt{\frac{\Delta_d}{n}},$$

Trong đó

$$\Delta_d = \min_{0 \leq \theta \leq 2\pi, \vec{a} \in \mathbb{R}^2} \sum_{j=0}^{n-1} \left\| k R_\theta U_j + a - V_{(j+d) \bmod m} \right\|^2, \quad k = \sqrt{\frac{\text{area}(V_0 \cdots V_{m-1})}{\text{area}(U_0 \cdots U_{n-1})}}, \text{ với } R_\theta \text{ là}$$

phép quay quanh gốc tọa độ một góc θ .

Trong đó, Δ_d được tính hiệu quả bằng công thức sau:

$$\Delta_d = \sum_{j=0}^{n-1} |V_{(j+d) \bmod m}|^2 - \frac{1}{n} \left| \sum_{j=0}^{n-1} V_{(j+d) \bmod m} \right|^2 + k^2 \sum_{j=0}^{n-1} |U_j|^2 - 2k \left| \sum_{j=0}^{n-1} U_j \bar{V}_{(j+d) \bmod m} \right|$$

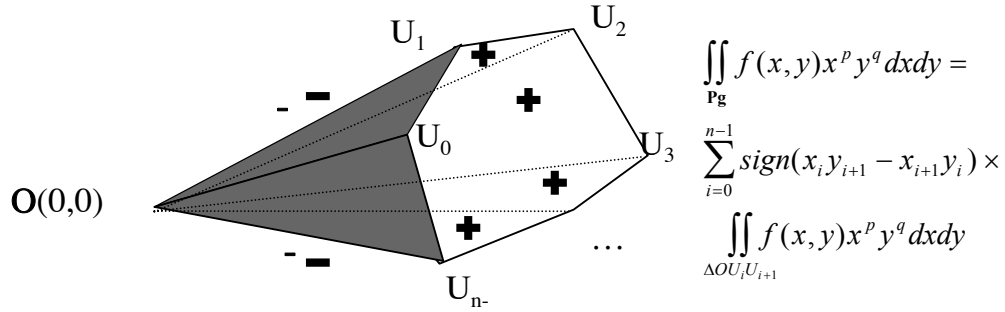
Ở đây U_j, V_j được hiểu là các số phức tại các đỉnh tương ứng. Khi $m \gg n$ thì độ phức tạp tính toán rất lớn. Với các hình đặc biệt như hình tròn, ellipse, hình chữ nhật, hình xác định duy nhất bởi tâm và một đỉnh (đa giác đều) ta có thể vận dụng các phương pháp đơn giản hơn như bình phương tối thiểu, các bất biến thống kê và hình học.

Định nghĩa 5.1

Cho đa giác P_g có các đỉnh $U_0, U_1, \dots, U_n (U_0 \equiv U_n)$ Khi đó mô men bậc $p+q$ được xác định như sau:

$$M_{pq} = \iint_{P_g} x^p y^q dx dy.$$

Trong thực hành để tính tích phân trên người ta thường sử dụng công thức Green hoặc có thể phân tích phần bên trong đa giác thành tổng đại số của các tam giác có hướng $\Delta O U_i U_{i+1}$.

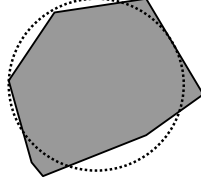


Hình 5.6. Phân tích miền đa giác thành tổng đại số các miền tam giác

a. Xấp xỉ đa giác bằng đường tròn

Dùng phương pháp bình phương tối thiểu, ta có độ đo xấp xỉ:

$$E(Pg, Cr) = \min_{a,b,c \in \mathbb{R}} \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i^2 + y_i^2 + ax_i + by_i + c)^2}$$



Hình 5.7. Xấp xỉ đa giác bằng đường tròn

b. Xấp xỉ đa giác bằng ellipse

Cũng như đối với đường tròn phương trình xấp xỉ đối với ellipse được cho bởi công thức:

$$E(Pg, El) = \min_{a,b,c,d,e \in \mathbb{R}} \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i^2 + ay_i^2 + bx_i y_i + cx_i + dy_i + e)^2}$$

Một biến thể khác của phương pháp bình phương tối thiểu khi xấp xỉ các đường cong bậc hai được đưa ra trong [7].

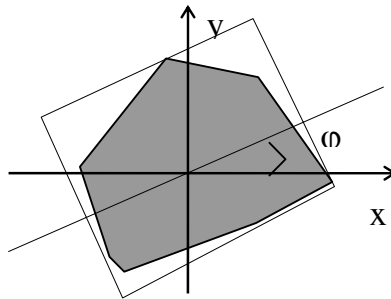
c. Xấp xỉ đa giác bởi hình chữ nhật

Sử dụng tính chất diện tích bất biến qua phép quay, xấp xỉ theo diện tích như sau: Gọi $\mu_{11}, \mu_{20}, \mu_{02}$ là các mô men bậc hai của đa giác (tính theo diện tích). Khi đó góc quay được tính bởi công thức sau:

$$\tan 2\varphi = \frac{2\mu_{11}}{\mu_{20} - \mu_{02}}$$

Gọi diện tích của hình chữ nhật nhỏ nhất có các cạnh song song với các trục quán tính và bao quanh đa giác Pg là S.

$$\text{Kí hiệu } E(Pg, \text{Rect}) = \sqrt{S - \text{area}(Pg)}$$



Hình 5.8. Xấp xỉ đa giác bằng hình chữ nhật

d. Xấp xỉ đa giác bởi đa giác đều n cạnh

Gọi $M(x_0, y_0)$ là trọng tâm của đa giác, lấy một đỉnh Q tùy ý của đa giác, xét đa giác đều n cạnh Pg' tạo bởi đỉnh Q với tâm là M .

$$\text{Kí hiệu } E(Pg, Pg') = \sqrt{|area(Pg) - area(Pg')|}$$

$E(Pg, E_n) = \min E(Pg, Pg')$ khi Q chạy khắp các đỉnh của đa giác.

5.2.2 Xấp xỉ đa giác theo bất biến afin

Trong [7] đưa ra mô hình chuẩn tắc về bất biến afin, cho phép chúng ta có thể chuyển bài toán xấp xỉ đối tượng bởi bất biến afin về bài toán xấp xỉ mẫu trên các dạng chuẩn tắc. Như vậy có thể đưa việc đối sánh các đối tượng với mẫu bởi các bất biến đồng dạng, chẳng hạn việc xấp xỉ bởi tam giác, hình bình hành, ellipse tương đương với xấp xỉ tam giác đều, hình vuông, hình tròn v.v... Thủ tục xấp xỉ theo bất biến afin một đa giác với hình cơ sở được thực hiện tuần tự như sau:

+ Bước 0:

Phân loại bất biến afin các dạng hình cơ sở

Dạng hình cơ sở	Dạng chuẩn tắc
Tam giác	Tam giác đều
Hình bình hành	Hình vuông
Ellipse	Đường tròn
...	...

+ Bước 1:

Tìm dạng chuẩn tắc cơ sở Pg' thỏa mãn điều kiện:

$$\begin{cases} m_{01} = m_{10} = 0 & (\text{phép tịnh tiến}) \\ m_{02} = m_{20} = 1 & (\text{phép co giãn theo hai trục } x, y) \\ m_{13} = m_{31} = 0 & \end{cases} \quad (**)$$

+ Bước 2:

Xác định biến đổi afin T chuyển đa giác thành đa giác Pg ở dạng chuẩn tắc (thỏa mãn tính chất (**)).

Xấp xỉ đa giác Pg với dạng chuẩn tắc cơ sở Pg' tìm được ở bước 1 với độ đo xấp xỉ $E(Pg, Pg')$.

+ Bước 3:

Kết luận, đa giác ban đầu xấp xỉ $T^{-1}(Pg')$ với độ đo xấp xỉ $E(Pg, Pg')$.

Đối với bước 1 trong [7] đã đưa ra hai ví dụ sau:

Ví dụ 1:

Tồn tại duy nhất tam giác đều $\Delta P_1P_2P_3$ thỏa mãn tính chất (**) là

$$P_1=(0,-2\alpha), P_2=(\sqrt{3}\alpha,\alpha), P_3=(-\sqrt{3}\alpha,\alpha), \alpha=\frac{\sqrt[4]{2}\sqrt{3}}{\sqrt{3}}.$$

Ví dụ 2:

Tồn tại hai hình vuông $\square P_1P_2P_3P_4$ thỏa mãn tính chất (**) là

Hình vuông thứ nhất có 4 đỉnh tương ứng là $(-p,-p), (-p,p), (p,-p), (p,p)$, với $p=\sqrt[4]{\frac{3}{4}}$

Hình vuông thứ hai có 4 đỉnh tương ứng là $(-p,0), (p,0), (0,-p), (0,p)$, với $p=\sqrt[4]{3}$.

5.3. BIẾN ĐỔI HOUGH

5.3.1. Biến đổi Hough cho đường thẳng

Bằng cách nào đó ta thu được một số điểm vấn đề đặt ra là cần phải kiểm tra xem các điểm có là đường thẳng hay không

Bài toán:

Cho n điểm $(x_i; y_i)$ $i = 1, n$ và ngưỡng θ hãy kiểm tra n điểm có tạo thành đường thẳng hay không?

** Ý tưởng*

Giả sử n điểm nằm trên cùng một đường thẳng và đường thẳng có phương trình

$$y = ax + b$$

Vì (x_i, y_i) $i = 1, n$ thuộc đường thẳng nên $y_i = ax_i + b, \forall i = 1, n$

$$\Leftrightarrow b = -x_i a + y_i; \forall i = 1, n$$

Như vậy, mỗi điểm $(x_i; y_i)$ trong mặt phẳng sẽ tương ứng với một số đường thẳng $b = -x_i a + y_i$ trong mặt phẳng tham số a, b . n điểm $(x_i; y_i)$ $i = 1, n$ thuộc đường thẳng trong mặt phẳng tương ứng với n đường thẳng trong mặt phẳng tham số a, b giao nhau tại 1 điểm và điểm giao chính là a, b . Chính là hệ số xác định phương trình của đường thẳng mà các điểm nằm vào.

** Phương pháp:*

- Xây dựng mảng chỉ số $[a, b]$ và gán giá trị 0 ban đầu cho tất cả các phần tử của mảng

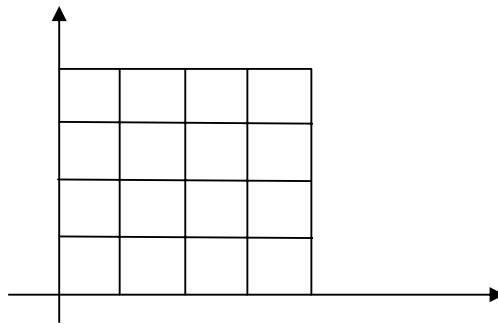
- Với mỗi (x_i, y_i) và $\forall a, b$ là chỉ số của phần tử mảng thỏa mãn $b = -x_i a + y_i$ tăng giá trị của phần tử mảng tương ứng lên 1

- Tìm phần tử mảng có giá trị lớn nhất nếu giá trị lớn nhất tìm được so với số phần tử lớn hơn hoặc bằng ngưỡng θ cho trước thì ta có thể kết luận các điểm nằm trên cùng 1 đường thẳng và đường thẳng có phương trình $y = ax + b$ trong đó a, b tương ứng là chỉ số của phần tử mảng có giá trị lớn nhất tìm được:

Ví dụ:

Cho 5 điểm $(0, 1); (1, 3); (2, 5); (3, 5); (4, 9)$ và $\theta = 80\%$. Hãy kiểm tra xem 5 điểm đã cho có nằm trên cùng một đường thẳng hay không? Hãy cho biết phương trình đường thẳng nếu có?

- Lập bảng chỉ số $[a, b]$ và gán giá trị 0



$$+ (0, 1): b = 1$$

$$+ (1, 3): b = -a + 3$$

$$+ (2, 5): b = -2a + 5$$

$$+ (3, 5): b = -3a + 5$$

$$+ (4, 9): b = -4a + 9$$

- Tìm phần tử lớn nhất có giá trị 4

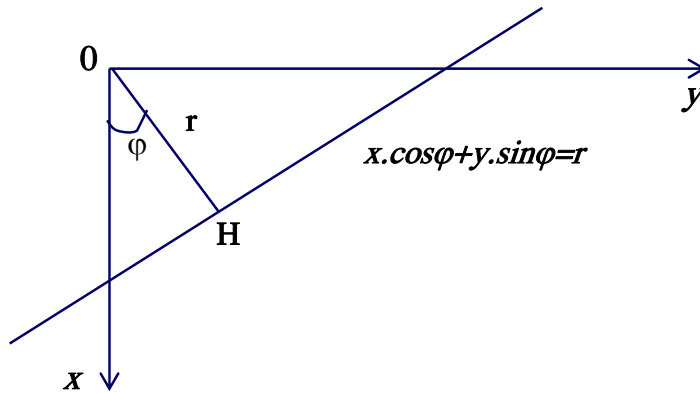
$$4/5 = 80\%$$

- Kết luận: 5 điểm này nằm trên cùng 1 đường thẳng

$$\text{Phương trình: } y = 2x + 1$$

5.3.2. Biến đổi Hough cho đường thẳng trong tọa độ cực

5.3.2.1. Đường thẳng Hough trong tọa độ cực



Hình 5.9. Đường thẳng Hough trong tọa độ cực

Mỗi điểm (x,y) trong mặt phẳng được biểu diễn bởi cặp (r,φ) trong tọa độ cực.

Tương tự mỗi đường thẳng trong mặt phẳng cũng có thể biểu diễn bởi một cặp (r,φ) trong tọa độ cực với r là khoảng cách từ gốc tọa độ tới đường thẳng đó và φ là góc tạo bởi trục OX với đường thẳng vuông góc với nó, hình 5.9 biểu diễn đường thẳng hough trong tọa độ Decard.

Ngược lại, mỗi một cặp (r,φ) trong tọa độ cực cũng tương ứng biểu diễn một đường thẳng trong mặt phẳng.

Giả sử $M(x,y)$ là một điểm thuộc đường thẳng được biểu diễn bởi (r,φ) , gọi $H(X,Y)$ là hình chiếu của gốc tọa độ O trên đường thẳng ta có:

$$X = r \cdot \cos\varphi \text{ và } Y = r \cdot \sin\varphi$$

Mặt khác, ta có: $\vec{OH} \cdot \vec{HA} = 0$

Từ đó ta có mối liên hệ giữa (x,y) và (r,φ) như sau: $x \cdot \cos\varphi + y \cdot \sin\varphi = r$.

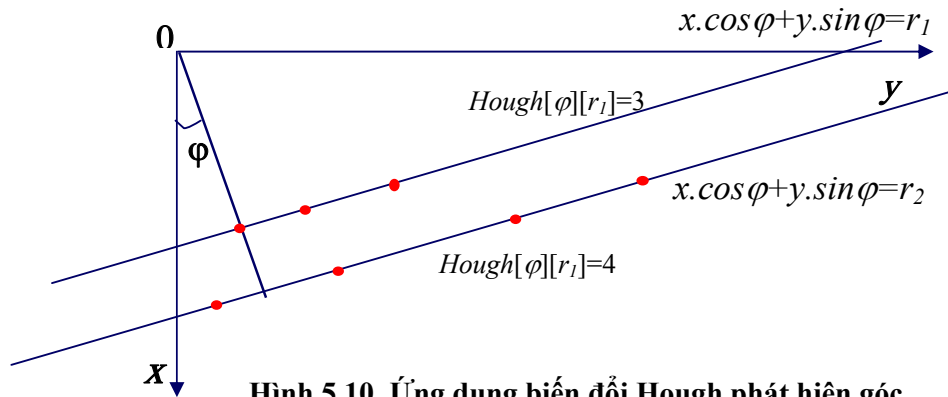
Xét n điểm thẳng hàng trong tọa độ Đề các có phương trình $x \cdot \cos\varphi_0 + y \cdot \sin\varphi_0 = r_0$. Biến đổi Hough ánh xạ n điểm này thành n đường sin trong tọa độ cực mà các đường này đều đi qua (r_0, φ_0) . Giao điểm (r_0, φ_0) của n đường sin sẽ xác định một đường thẳng trong hệ tọa độ đề các. Như vậy, những đường thẳng đi qua điểm (x,y) sẽ cho duy nhất một cặp (r,φ) và có bao nhiêu đường qua (x,y) sẽ có bấy nhiêu cặp giá trị (r,φ) .

5.3.2.2. Áp dụng biến đổi Hough trong phát hiện góc nghiêng văn bản

Ý tưởng của việc áp dụng biến đổi Hough trong phát hiện góc nghiêng văn bản là dùng một mảng tích lũy để đếm số điểm ảnh nằm trên một đường thẳng trong không gian ảnh. Mảng tích lũy là một mảng hai chiều với chỉ số hàng của mảng cho biết góc lệch φ của một đường thẳng và chỉ số cột chính là giá trị r khoảng cách từ gốc tọa độ tới đường thẳng đó. Sau đó tính tổng số điểm ảnh nằm trên những đường thẳng song song nhau theo

các góc lệch thay đổi. Góc nghiêng văn bản tương ứng với góc có tổng giá trị mảng tích lũy cực đại.

Theo biến đổi Hough, mỗi một đường thẳng trong mặt phẳng tương ứng được biểu diễn bởi một cặp (r, φ) . Giả sử ta có một điểm ảnh (x, y) trong mặt phẳng, vì qua điểm ảnh này có vô số đường thẳng, mỗi đường thẳng lại cho một cặp (r, φ) nên với mỗi điểm ảnh ta sẽ xác định được một số cặp (r, φ) thoả mãn phương trình Hough.



Hình 5.10. Ứng dụng biến đổi Hough phát hiện góc

Hình vẽ trên minh họa cách dùng biến đổi Hough để phát hiện góc nghiêng văn bản. Giả sử ta có một số điểm ảnh, đây là những điểm giữa đáy các hình chữ nhật ngoại tiếp các đối tượng đã được lựa chọn từ các bước trước. Ở đây, ta thấy trên mặt phẳng có hai đường thẳng song song nhau. Đường thẳng thứ nhất có ba điểm ảnh nên giá trị mảng tích lũy bằng 3, đường thẳng thứ hai có giá trị mảng tích lũy bằng 4. Do đó, tổng giá trị mảng tích lũy cho cùng góc φ trường hợp này bằng 7.

Gọi $Hough[360][Max]$ là mảng tích lũy, giả sử M và N tương ứng là chiều rộng và chiều cao của ảnh, ta có các bước chính trong quá trình áp dụng biến đổi Hough phát hiện góc nghiêng văn bản như sau:

- + **Bước 1:** Khai báo mảng chỉ số $Hough[\varphi][r]$ với $0 \leq \varphi \leq 3600$ và $0 \leq r \leq \sqrt{M * M + N * N}$.
- + **Bước 2:** Gán giá trị khởi tạo bằng 0 cho các phần tử của mảng.
- + **Bước 3:** Với mỗi cặp (x, y) là điểm giữa đáy của hình chữ nhật ngoại tiếp một đối tượng.
 - Với mỗi φ_i từ 0 đến 360 tính giá trị r_i theo công thức $r_i = x \cdot \cos \varphi_i + y \cdot \sin \varphi_i$
 - Làm tròn giá trị r_i thành số nguyên gần nhất là r_0
 - Tăng giá trị của phần tử mảng $Hough[\varphi_i][r_0]$ lên một đơn vị.

+ **Bước 4**: Trong mảng Hough[φ][r] tính tổng giá trị các phần tử theo từng dòng và xác định dòng có tổng giá trị lớn nhất.

Do số phần tử của một phần tử mảng Hough[φ_0][r_0] chính là số điểm ảnh thuộc đường thẳng $x.\cos\varphi_0 + y.\sin\varphi_0 = r_0$ vì vậy tổng số phần tử của một hàng chính là tổng số điểm ảnh thuộc các đường thẳng tương ứng được biểu diễn bởi góc φ của hàng đó. Do đó, góc nghiêng của toán văn bản chính là hàng có tổng giá trị các phần tử mảng lớn nhất.

Phụ lục 1:

MỘT SỐ ĐỊNH DẠNG TRONG XỬ LÝ ẢNH

Hiện nay trên thế giới có trên 50 khuôn dạng ảnh thông dụng. Sau đây là một số định dạng ảnh hay dùng trong quá trình xử lý ảnh hiện nay.

1. Định dạng ảnh IMG

Ảnh IMG là ảnh đen trắng, phần đầu của ảnh IMG có 16 byte chứa các thông tin:

- 6 byte đầu: dùng để đánh dấu định dạng ảnh. Giá trị của 6 byte này viết dưới dạng Hexa: 0x0001 0x0008 0x0001
- 2 byte tiếp theo: chứa độ dài mẫu tin. Đó là độ dài của dãy các byte kề liền nhau mà dãy này sẽ được lặp lại một số lần nào đó. Số lần lặp này sẽ được lưu trong byte đếm. Nhiều dãy giống nhau được lưu trong một byte.
- 4 byte tiếp: mô tả kích cỡ pixel.
- 2 byte tiếp: số pixel trên một dòng ảnh.
- 2 byte cuối: số dòng ảnh trong ảnh.

Ảnh IMG được nén theo từng dòng, mỗi dòng bao gồm các gói (pack). Các dòng giống nhau cũng được nén thành một gói. Có 4 loại gói sau:

- Loại 1: Gói các dòng giống nhau.

Quy cách gói tin này như sau: 0x00 0x00 0xFF Count. Ba byte đầu tiên cho biết số các dãy giống nhau, byte cuối cho biết số các dòng giống nhau.

- Loại 2: Gói các dãy giống nhau.

Quy cách gói tin này như sau: 0x00 Count. Byte thứ hai cho biết số các dãy giống nhau được nén trong gói. Độ dài của dãy ghi ở đầu tệp.

- Loại 3: Dãy các Pixel không giống nhau, không lặp lại và không nén được.

Quy cách gói tin này như sau: 0x80 Count. Byte thứ hai cho biết độ dài dãy các pixel không giống nhau không nén được.

- Loại 4: Dãy các Pixel giống nhau.

Tuỳ theo các bit cao của byte đầu tiên được bật hay tắt. Nếu bit cao được bật (giá trị 1) thì đây là gói nén các byte chỉ gồm bit 0, số các byte được nén được tính bởi 7 bit thấp còn lại. Nếu bit cao tắt (giá trị 0) thì đây là gói nén các byte gồm toàn bit 1. Số các byte được nén được tính bởi 7 bit còn lại.

Các gói tin của file IMG rất đa dạng do ảnh IMG là ảnh đen trắng, do vậy chỉ cần 1 bit cho 1 pixel thay vì 4 hoặc 8 như đã nói ở trên. Toàn bộ ảnh chỉ có những điểm sáng và tối tương ứng với giá trị 1 hoặc 0. Tỷ lệ nén của kiểu định dạng này là khá cao.

2. Định dạng ảnh PCX

Định dạng ảnh PCX là một trong những định dạng ảnh cổ điển. Nó sử dụng phương pháp mã hoá loạt dài RLE (Run – Length – Encoded) để nén dữ liệu ảnh. Quá trình nén và giải nén được thực hiện trên từng dòng ảnh. Thực tế, phương pháp giải nén PCX kém hiệu quả hơn so với kiểu IMG. Tập PCX gồm 3 phần: đầu tệp (header), dữ liệu ảnh (Image data) và bảng màu mở rộng.

Header của tệp PCX có kích thước cố định gồm 128 byte và được phân bố như sau:

- 1 byte: chỉ ra kiểu định dạng. Nếu là PCX/PCC thì nó luôn có giá trị là 0Ah.
- 1 byte: chỉ ra version sử dụng để nén ảnh, có thể có các giá trị sau:
 - + 0: version 2.5.
 - + 2: version 2.8 với bảng màu.
 - + 3: version 2.8 hay 3.0 không có bảng màu.
 - + 5: version 3.0 có bảng màu.
- 1 byte: chỉ ra phương pháp mã hoá. Nếu là 0 thì mã hoá theo phương pháp BYTE PACKED, ngược lại là phương pháp RLE.
- 1 byte: Số bit cho một điểm ảnh plane.
- 1 word: toạ độ góc trái của ảnh. Với kiểu PCX nó có giá trị là (0,0), cũn PCC thì khác (0,0).
- 1 word: toạ độ góc phải dưới.
- 1 word: kích thước bề rộng và bề cao của ảnh.

- 1 word: số điểm ảnh.
- 1 word: độ phân giải màn hình.
- 1 word.
- 48 byte: chia nó thành 16 nhóm, mỗi nhóm 3 byte. Mỗi nhóm này chứa thông tin về một thanh ghi màu. Như vậy ta có 16 thanh ghi màu.
- 1 byte: không dùng đến và luôn đặt là 0.
- 1 byte: số bít plane mà ảnh sử dụng. Với ảnh 16 màu, giá trị này là 4, với ảnh 256 màu (1pixel/8bits) thì số bít plane lại là 1.
- 1 byte: số bytes cho một dòng quét ảnh.
- 1 word: kiểu bảng màu.
- 58 byte: không dùng.

Định dạng ảnh PCX thường được dùng để lưu trữ ảnh và thao tác đơn giản, cho phép nén và giải nén nhanh. Tuy nhiên, vì cấu trúc của nó cố định, nên trong một số trường hợp làm tăng kích thước lưu trữ. Cũng vì nhược điểm này mà một số ứng dụng sử dụng một kiểu định dạng khác mềm dẻo hơn: định dạng TIFF (Targed Image File Format) sẽ mô tả dưới đây.

3. Định dạng ảnh TIFF

Kiểu định dạng TIFF được thiết kế để làm nhẹ bớt các vấn đề liên quan đến việc mở rộng tệp ảnh cố định. Về cấu trúc, nó cũng gồm 3 phần chính:

- Phần Header(IFH): cú trong tất cả các tệp TIFF và gồm 8 byte:
 - + 1 word: chỉ ra kiểu tạo tệp trên máy tính PC hay máy Macintosh. Hai loại này khác nhau rất lớn ở thứ tự các byte lưu trữ trong các số dài 2 hay 4 byte. Nếu trường này có giá trị là 4D4Dh thì đó là ảnh cho máy Macintosh, nếu là 4949h là của máy PC.
 - + 1 word: version. từ này luôn có giá trị là 42. đây là đặc trưng của file TIFF và không thay đổi.
 - + 2 word: giá trị Offset theo byte tính từ đầu tới cấu trúc IFD là cấu trúc thứ hai của file. Thứ tự các byte này phụ thuộc vào dấu hiệu trường đầu tiên.

- Phần thứ 2(IFD): Không ở ngay sau cấu trúc IFH mà vị trí được xác định bởi trường Offset trong đầu tệp. Có thể có một hay nhiều IFD cùng tồn tại trong một file.

Một IFD bao gồm:

- + 2 byte: chứa các DE (Directory Entry).
- + 12 byte là các DE xếp liên tiếp, mỗi DE chiếm 12 byte.
- + 4 byte: chứa Offset trở tới IFD tiếp theo. Nếu đây là IFD cuối cùng thì trường này có giá trị 0.
- Phần thứ 3: các DE: các DE có độ dài cố định gồm 12 byte và chia làm 4 phần:
 - + 2 byte: chỉ ra dấu hiệu mà tệp ảnh đó được xây dựng.
 - + 2 byte: kiểu dữ liệu của tham số ảnh. Có 5 kiểu tham số cơ bản:
 - 1: BYTE (1 byte)
 - 2: ASCII (1 byte)
 - 3: SHORT (2 byte).
 - 4: LONG (4 byte)
 - 5: RATIONAL (8 byte)
 - + 4 byte: trường độ dài chứa số lượng chỉ mục của kiểu dữ liệu đó chỉ ra. Nó không phải là tổng số byte cần thiết để lưu trữ. Để có số liệu này ta cần nhân số chỉ mục với kiểu dữ liệu đã dùng.
 - + 4 byte: đó là Offset tới điểm bắt đầu dữ liệu liên quan tới dấu hiệu, tức là liên quan với DE không phải lưu trữ vật lý cùng với nó nằm ở một vị trí nào đó trong file.

Dữ liệu chứa trong tệp thường được tổ chức thành các nhóm dòng (cột) quét của dữ liệu ảnh. Cách tổ chức này làm giảm bộ nhớ cần thiết cho việc đọc tệp. Việc giải nén được thực hiện theo 4 kiểu khác nhau được lưu trữ trong byte dấu hiệu nén.

4. Định dạng file ảnh BITMAP

Mỗi file BITMAP gồm đầu file chứa các thông tin chung về file, đầu thông tin chứa các thông tin về ảnh, một bảng màu và một mảng dữ liệu ảnh. Khuôn dạng được cho như sau:

```
BITMAPFILEHEADER bmfh;  
BITMAPINFOHEADERbmih;  
RGBQUAD          aColors[];  
BYTE              aBitmapBits[];
```

Trong đó, các cấu trúc được định nghĩa như sau:

```
typedef struct tagBITMAPFILEHEADER { /* bmfh */  
    UINT    bfType;  
    DWORD   bfSize;  
    UINT    bfReserved1;  
    UINT    bfReserved2;  
    DWORD   bfOffBits;  
} BITMAPFILEHEADER;  
  
typedef struct tagBITMAPINFOHEADER { /* bmih */  
    DWORD   biSize;  
    LONG    biWidth;  
    LONG    biHeight;  
    WORD    biPlanes;  
    WORD    biBitCount;  
    DWORD   biCompression;  
    DWORD   biSizeImage;  
    LONG    biXPelsPerMeter;  
    LONG    biYPelsPerMeter;  
    DWORD   biClrUsed;  
    DWORD   biClrImportant;  
} BITMAPINFOHEADER, *LPBITMAPINFOHEADER;
```

với

biSize	kích thước của BITMAPINFOHEADER
biWidth	Chiều rộng của ảnh, tính bằng số điểm ảnh
biHeight	Chiều cao của ảnh, tính bằng số điểm ảnh

biPlanes	Số plane của thiết bị, phải bằng 1
biBitCount	Số bit cho một điểm ảnh
biCompression	Kiểu nén
biSizeImage	Kích thước của ảnh tính bằng byte
biXPelsPerMeter	độ phân giải ngang của thiết bị, tính bằng điểm ảnh trên met
biYPelsPerMeter	độ phân giải dọc của thiết bị, tính bằng điểm ảnh trên met
biClrUsed	Số lượng các màu thực sự được sử dụng
biClrImportant	Số lượng các màu cần thiết cho việc hiển thị, bằng 0 nếu tất cả các màu đều cần để hiển thị

Nếu `bmi.biBitCount > 8` thì mảng màu `rgbq[]` trống, ngược lại thì mảng màu có $2^{bmi.biBitCount}$ phần tử.

```
typedef struct tagRGBQUAD {    /* rgbq */
    BYTE    rgbBlue;
    BYTE    rgbGreen;
    BYTE    rgbRed;
    BYTE    rgbReserved;
} RGBQUAD;
```

Ta cũng có:

```
typedef struct tagBITMAPINFO {
    BITMAPINFOHEADER    bmiHeader;
    RGBQUAD              bmiColors[1];
} BITMAPINFO, *PBITMAPINFO;
```

Phụ lục 2:

CÁC BƯỚC THAO TÁC VỚI FILE AVI

AVI là chuẩn video thường được tích hợp trong các thư viện của các môi trường lập trình. Để xử lý video, cần có các thao tác cơ bản để chuyển về xử lý ảnh các khung hình (các frames).

1. Bước 1: Mở và đóng thư viện

Trước mọi thao tác với file AVI, chúng ta phải mở thư viện:

AVIFileInit()

Hàm này không cần tham số, có nhiệm vụ khởi động thư viện cung cấp các hàm thao tác với file AVI. (Đó là thư viện vfw32.lib, được khai báo trong file vfw.h).

Sau tất cả các thao tác bạn phải nhớ đóng thư viện đã mở lúc đầu, chỉ bằng lệnh:

AVIFileExit()

Nếu thiếu bất cứ hàm nào, dù là mở hay đóng thư viện thì trình biên dịch đều sẽ thông báo lỗi.

2. Bước 2: Mở và đóng file AVI để thao tác:

Sau khi mở thư viện, bạn phải mở file AVI bạn định thao tác:

AVIFileOpen(PAVIFILE ppfile, LPCSTR fname, UINT mode, CLSID pclsidHandler)*

Thực chất, hàm này tạo ra một vùng đệm chứa con trỏ trỏ đến file có tên là fname cần mở. Và ppfile là con trỏ trỏ đến vùng bộ đệm đó. Tham số mode quy định kiểu mở file; chẳng hạn OF_CREATE để tạo mới, OF_READ để đọc, OF_WRITE để ghi Tham số cuối dùng là NULL.

Trước khi đóng thư viện, bạn phải đóng file AVI đã mở, bằng cách dùng hàm:

AVIFileRelease(PAVIFILE pfile)

Trong đó, pfile là con trỏ trỏ đến file cần đóng.

3. Bước 3:

Mở dòng dữ liệu hình ảnh hay âm thanh trong file AVI đã mở ra để thao tác:

*AVIFileGetStream(PAVIFILE pfile, PAVISTREAM * ppavi, DWORD fccType, LONG lParam)*

Trong đó, pfile là con trỏ đến file đã mở; ppavi trỏ đến dòng dữ liệu kết quả; fccType là loại dòng dữ liệu chọn để mở, là streamtypeAUDIO nếu là tiếng và streamtypeVIDEO nếu là hình,... lParam đếm số loại dòng được mở, là 0 nếu chỉ thao tác với một loại dòng dữ liệu.

Sau các thao tác với dòng dữ liệu này, bạn nhớ phải đóng nó lại:

AVIStreamRelease(PAVITREAM pavi).

4. Bước 4: Trường hợp thao tác với dữ liệu hình của phim

Chuẩn bị cho thao tác với khung hình (frames):

AVIStreamGetFrameOpen(PAVISTREAM pavi, LPBITMAPINFOHEADER lpbiWanted)

Trong đó pavi trỏ đến dòng dữ liệu đã mở, lpbiWanted là con trỏ trỏ đến cấu trúc mong muốn của hình ảnh, ta dùng NULL để sử dụng cấu trúc mặc định.

Hàm này trả về đối tượng có kiểu PGETFRAME để dùng cho bước 5.

Sau khi thao tác với các frame rồi, phải gọi hàm :

AVIStreamGetFrameClose(PGETFRAME pget)

5. Bước 5: Thao tác với frame

Dùng hàm

AVIStreamGetFrame(PGETFRAME pget, LONG lpos)

Hàm này trả về con trỏ trỏ đến dữ liệu của frame thứ lpos. Dữ liệu đó có kiểu là DIB đã định khối.

Thực hiện các thao tác mong muốn.

TÀI LIỆU THAM KHẢO

- [1]. Lương Mạnh Bá, Nguyễn Thanh Thủy (2002), *Nhập Môn Xử lý ảnh số*, Nxb Khoa học và Kỹ thuật, 2002.
- [2]. Anil K.Jain (1989), *Fundamental of Digital Image Processing*. Prentice Hall, Engwood cliffs.
- [3]. J.R.Paker (1997), *Algorithms for Image processing and Computer Vision*. John Wiley & Sons, Inc.
- [4]. Randy Crane (1997), *A simplified approach to image processing*, Prentice-Hall, Inc.
- [5]. John C.Russ (1995), *The Image Procesing Handbook*. CRC Press, Inc.
- [6]. Adrian Low (1991), *Introductory Computer Vision and Image Processing*, Copyright (c) 1991 by McGrow Hill Book Company (UK) Limited.
- [7]. T. Pavlidis (1982), *Algorithms for Graphics and Image Processing*, Computer Science Press.