

MỞ ĐẦU

Xử lý ảnh (XLA) là một trong những chuyên ngành quan trọng và lâu đời của Công nghệ thông tin. XLA được áp dụng trong nhiều lĩnh vực khác nhau như y học, vật lý, hoá học, tìm kiếm tội phạm,... Mục đích chung của việc XLA thường là: (1) xử lý ảnh ban đầu để có được một bức ảnh mới theo một yêu cầu cụ thể; (2) phân tích ảnh để thu được các thông tin đặc trưng trên ảnh nhằm hỗ trợ cho việc phân loại và nhận biết ảnh; (3) phân đoạn ảnh (image segmentation) để nhận diện được các thành phần trong ảnh nhằm hiểu được kết cấu của bức ảnh ở mức độ cao hơn. Để xử lý được một bức ảnh thì phải trải qua nhiều bước, nhưng bước quan trọng và khó khăn nhất đó là phân đoạn ảnh. Nếu bước phân đoạn ảnh không tốt thì dẫn đến việc nhận diện sai lầm về các đối tượng có trong ảnh.

Trong khoảng 30 năm trở lại đây đã có rất nhiều các thuật toán được đề xuất để giải quyết bài toán phân đoạn ảnh. Các thuật toán hầu hết đều dựa vào hai thuộc tính quan trọng của mỗi điểm ảnh so với các điểm lân cận của nó, đó là: sự khác (dissimilarity) và giống nhau (similarity). Các phương pháp dựa trên sự khác nhau của các điểm ảnh được gọi là các phương pháp biên (boundary-based methods), còn các phương pháp dựa trên sự giống nhau của các điểm ảnh được gọi là phương pháp miền (region-based methods). Tuy nhiên, cho đến nay các thuật toán theo cả hai hướng này đều vẫn chưa cho kết quả phân đoạn tốt, vì cả hai loại phương pháp này đều chỉ nắm bắt được các thuộc tính cục bộ (local) của ảnh. Do đó, trong thời gian gần đây, việc tìm ra các thuật toán nắm bắt được các thuộc tính toàn cục (global) của bức ảnh đã trở thành một xu hướng.

Mục đích chính của em là tìm hiểu và hệ thống lại các phương pháp phân đoạn ảnh đã có theo các hướng: như phân đoạn theo ngưỡng, phân đoạn theo đường biên và theo miền đồng nhất. Ngoài ra, trong đề án này em cũng tìm hiểu và trình bày thêm một phương pháp được đánh giá là hiệu quả hơn các phương pháp trước đây. Phương pháp này dựa vào việc coi một bức ảnh như một đồ thị, sau đó định

nghĩa một tính chất để so sánh giữa các cặp miền của ảnh. Thuật toán này tuân theo một chiến lược tham lam, có thời gian chạy gần như tuyến tính, nhưng vẫn đảm bảo được việc phân đoạn chính xác và hiệu quả.

Ngoài phần mở đầu và kết luận, luận văn được chia làm 4 chương, cụ thể nội dung các chương như sau:

Chương 1 Trình bày sơ lược về XLA, giới thiệu các giai đoạn xử lý trong một hệ thống XLA, trong đó có bước phân đoạn ảnh. Một số khái niệm, thuật ngữ trong XLA, như điểm ảnh, mức xám, biên,... được trình bày như là các khái niệm.

Chương 2 Hệ thống lại một số thuật toán phân đoạn ảnh theo các hướng: phân đoạn theo ngưỡng, phân đoạn theo đường biên và phân đoạn theo miền đồng nhất. Trong mỗi loại phương pháp này chúng tôi trình bày ngắn gọn phương pháp và ưu nhược điểm của chúng.

Chương 3 Trình bày một thuật toán phân đoạn dựa trên đồ thị :Thuật toán coi mỗi pixel là một đỉnh của đồ thị, sự khác nhau giữa hai điểm ảnh là trọng số của cạnh nối hai đỉnh tương ứng với nhau. Thuật toán dựa theo chiến lược tham lam, nhưng có thể nắm bắt được các thuộc tính non-local của bức ảnh. Một số định lý và hệ quả liên quan đến thuật toán được trình bày và chứng minh ngắn gọn.

Chương 4 đưa ra các đoạn mã chương trình (code) bằng C++ mã hoá một số thuật toán được trình bày trong luận văn.

Khi viết báo cáo này em đã cố gắng hết sức để hoàn thành công việc được giao, song điều kiện thời gian và trình độ còn hạn chế nên không tránh khỏi thiếu sót. Em mong nhận được sự góp ý của thầy giáo hướng dẫn, thầy cô giáo và bạn bè trong khoa Công nghệ thông tin để em có được những kinh nghiệm thực tế và bổ ích để sau này có thể xây dựng được một chương trình hoàn thiện hơn.

CHƯƠNG 1 : TỔNG QUAN VỀ XỬ LÝ ẢNH VÀ PHÂN ĐOẠN ẢNH

Xử lý ảnh ngày nay đã trở thành một ngành khoa học lớn và có mặt trong nhiều lĩnh vực của cuộc sống. Điều này hoàn toàn có thể lý giải được từ một định nghĩa đơn giản về ngành khoa học này: Xử lý ảnh là ngành khoa học nghiên cứu các quá trình xử lý thông tin dạng hình ảnh^{Error! Reference source not found.}, mà hình ảnh là một trong những dạng thông tin phong phú nhất đối với chúng ta.. Trong quá trình xử lý ảnh bước quan trọng nhất và cũng là có khấn nhất là bước phân đoạn ảnh. Phân đoạn nhằm mục đích phân tách các đối tượng cấu thành nên ảnh thô để có thể sử dụng cho các ứng dụng về sau.

1.1 TỔNG QUAN VỀ XỬ LÝ ẢNH

1.1.1 Giới thiệu về Xử lý ảnh

Trong xã hội loài người, ngôn ngữ là một phương tiện trao đổi thông tin phổ biến trong quá trình giao tiếp. Bên cạnh ngôn ngữ, hình ảnh cũng là một cách trao đổi thông tin mang tính chính xác, biểu cảm khá cao và đặc biệt không bị cảm giác chủ quan của đối tượng giao tiếp chi phối. Thông tin trên hình ảnh rất phong phú, đa dạng và có thể xử lý bằng máy tính. Chính vì vậy, trong những năm gần đây sự kết hợp giữa ảnh và đồ hoạ đã trở nên rất chặt chẽ trong lĩnh vực xử lý thông tin.

Cũng như xử lý dữ liệu hình ảnh bằng đồ hoạ, việc XLA số là một lĩnh vực của tin học ứng dụng. Việc xử lý dữ liệu bằng đồ hoạ đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là những cấu trúc dữ liệu và được tạo ra bởi các chương trình. XLA số thao tác trên các ảnh tự nhiên thông qua các phương pháp và kỹ thuật mã hoá. Ảnh sau khi được thu nhận bằng các thiết bị thu nhận ảnh sẽ được biến đổi thành ảnh số theo các phương pháp số hoá được nhúng trong các thiết bị kỹ thuật khác nhau và được biểu diễn trong máy tính dưới dạng ma trận 2 chiều hoặc 3 chiều.

Mục đích của việc XLA được chia làm hai phần

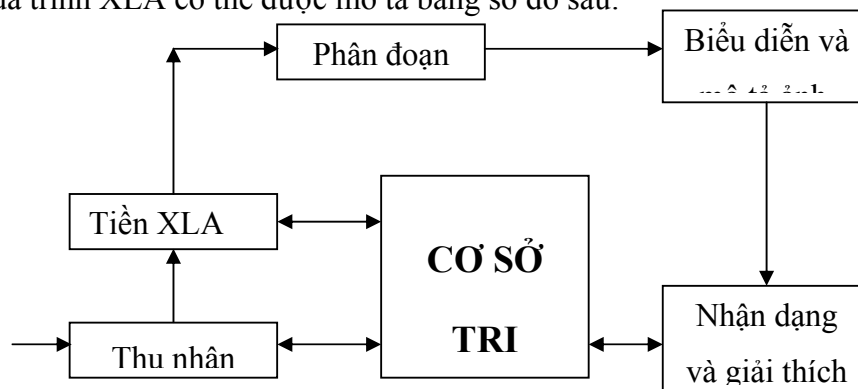
- Biến đổi ảnh làm tăng chất lượng ảnh
- Tự động nhận dạng, đoán ảnh, đánh giá nội dung của ảnh.

Phương pháp biến đổi ảnh được sử dụng trong việc xử lý các ảnh chụp từ không trung (chương trình đo đạc từ máy bay, vệ tinh và các ảnh vũ trụ) hoặc xử lý các ảnh trong y học (ảnh chụp cắt lát, ảnh siêu âm, vv...). Một ứng dụng khác của việc biến đổi ảnh là mã hoá ảnh, trong đó các ảnh được xử lý để rồi lưu trữ hoặc truyền đi.

Các phương pháp nhận dạng ảnh được sử dụng khi xử lý tế bào, nhiễm sắc thể, nhận dạng chữ vv... Thực chất của công việc nhận dạng chính là sự phân loại đối tượng thành các lớp đối tượng đã biết hoặc thành những lớp đối tượng chưa biết. Bài toán nhận dạng ảnh là một bài toán lớn, có rất nhiều ý nghĩa thực tiễn và ta cũng có thể thấy rằng để công việc nhận dạng trở nên dễ dàng thì ảnh phải được tách thành các đối tượng riêng biệt – đây là mục đích chính của bài toán phân đoạn ảnh. Nếu phân đoạn ảnh không tốt sẽ dẫn đến sai lầm trong quá trình nhận dạng ảnh, bởi vậy người ta xem công đoạn phân đoạn ảnh là vấn đề then chốt trong quá trình xử lý ảnh nói chung.

1.1.2 Quá trình XLA

Quá trình XLA có thể được mô tả bằng sơ đồ sau:



Hình 1. Quá trình xử lý ảnh

- Thu nhận ảnh: Đây là công đoạn đầu tiên mang tính quyết định đối với quá trình XLA. Ảnh đầu vào sẽ được thu nhận qua các thiết bị như camera, sensor, máy scanner, vv ... và sau đó các tín hiệu này sẽ được số hoá. Các thông số quan trọng ở bước này là độ phân giải, chất lượng màu, dung lượng bộ nhớ và tốc độ thu nhận ảnh của các thiết bị.
- Tiền xử lý: Ở bước này, ảnh sẽ được cải thiện về độ tương phản, khử nhiễu, khử bóng, khử độ lệch, v.v.. với mục đích làm cho chất lượng ảnh trở nên tốt hơn nữa và thường được thực hiện bởi các bộ lọc.
- Phân đoạn ảnh: Phân đoạn ảnh là *bước then chốt* trong XLA. Giai đoạn này nhằm phân tích ảnh thành những thành phần có cùng tính chất nào đó dựa theo biên hay các vùng liên thông. Tiêu chuẩn để xác định các vùng liên thông có thể là cùng màu, cùng mức xám hay cùng độ nhám vv ... Mục đích của phân đoạn ảnh là để có một miêu tả tổng hợp về nhiều phần tử khác nhau cấu tạo nên ảnh thô. Vì lượng thông tin chứa trong ảnh rất lớn – trong khi trong đa số các ứng dụng chúng ta chỉ cần trích chọn một vài đặc trưng nào đó, do vậy cần có một quá trình để giảm lượng thông tin khổng lồ ấy. Quá trình này bao gồm ***phân vùng ảnh*** và ***trích chọn đặc tính*** chủ yếu.
- Biểu diễn và mô tả ảnh: Kết quả của bước phân đoạn ảnh thường được cho dưới dạng dữ liệu điểm ảnh thô, trong đó hàm chứa biên của một vùng ảnh, hoặc tập hợp tất cả các điểm ảnh thuộc về chính vùng ảnh đó. Trong cả hai trường hợp, sự chuyển đổi dữ liệu thô này thành một dạng thích hợp hơn cho việc xử lý trong máy tính là rất cần thiết. Để chuyển đổi chúng, câu hỏi đầu tiên cần phải trả lời là nên biểu diễn một vùng ảnh *dưới dạng biên* hay *dưới dạng một vùng hoàn chỉnh* gồm tất cả những điểm ảnh thuộc về nó. Biểu diễn dạng biên cho một vùng phù hợp với những ứng dụng chỉ quan tâm chủ yếu đến các đặc trưng hình dạng bên ngoài của đối tượng, ví dụ như các góc cạnh và điểm uốn trên biên chẳng hạn. Biểu diễn dạng vùng lại thích hợp cho những ứng dụng khai thác các tính chất bên trong của đối tượng, ví dụ như vân ảnh hoặc cấu trúc xương của

nó. Sự chọn lựa cách biểu diễn thích hợp cho một vùng ảnh chỉ mới là một phần trong việc chuyển đổi dữ liệu ảnh thô sang một dạng thích hợp hơn cho các xử lý về sau. Chúng ta còn phải đưa ra một phương pháp mô tả dữ liệu đã được chuyển đổi đó sao cho những tính chất cần quan tâm đến sẽ được làm nổi bật lên, thuận tiện cho việc xử lý chúng.

- Nhận dạng và giải thích: Đây là bước cuối cùng trong quá trình XLA. Nhận dạng ảnh (image recognition) có thể được nhìn nhận một cách đơn giản là việc *gán nhãn* cho các đối tượng trong ảnh. Giải thích là công đoạn *gán nghĩa* cho một tập các đối tượng đã được nhận biết.

Chúng ta cũng có thể thấy rằng, không phải bất kỳ một ứng dụng XLA nào cũng bắt buộc phải tuân theo tất cả các bước xử lý đã nêu ở trên, ví dụ như các ứng dụng chỉnh sửa ảnh nghệ thuật chỉ dừng lại ở bước tiền xử lý. Một cách tổng quát thì những chức năng xử lý bao gồm nhận cả nhận dạng và giải thích thường chỉ có mặt trong hệ thống phân tích ảnh tự động hoặc bán tự động, được dùng để rút trích ra những thông tin quan trọng từ ảnh, ví dụ như các ứng dụng nhận dạng ký tự quang học, nhận dạng chữ viết tay vv...

1.2. TỔNG QUAN VỀ PHÂN ĐOẠN ẢNH

Để phân tích các đối tượng trong ảnh, chúng ta cần phải phân biệt được các đối tượng cần quan tâm với phần còn lại của ảnh, hay còn gọi là nền ảnh. Những đối tượng này có thể tìm ra được nhờ các *kỹ thuật phân đoạn ảnh*, theo nghĩa tách phần tiền cảnh ra khỏi hậu cảnh trong ảnh. Mỗi một đối tượng trong ảnh được gọi là một vùng hay miền, đường bao quanh đối tượng ta gọi là đường biên. Mỗi một vùng ảnh phải có các đặc tính đồng nhất (ví dụ: màu sắc, kết cấu, mức xám vv...). Các đặc tính này tạo nên một véc tơ đặc trưng riêng của vùng (**feature vectors**) giúp chúng ta phân biệt được các vùng khác nhau.

Như vậy, hình dáng của một đối tượng có thể được miêu tả hoặc bởi các tham số của đường biên hoặc các tham số của vùng mà nó chiếm giữ. Sự miêu tả hình dáng dựa trên thông tin đường biên yêu cầu việc *phát hiện biên*. Sự mô tả hình dáng dựa

vào vùng đòi hỏi việc *phân đoạn ảnh* thành một số vùng đồng nhất. Có thể thấy kỹ thuật phát hiện biên và phân vùng ảnh là hai bài toán đối ngẫu của nhau. Thực vậy, dò biên để thực hiện phân lớp đối tượng và một khi đã phân lớp xong cũng có nghĩa là đã phân vùng được ảnh. Ngược lại, khi đã phân vùng, ảnh được phân lập thành các đối tượng, ta có thể phát hiện biên.

Có rất nhiều kỹ thuật phân đoạn ảnh, nhưng nhìn chung chúng ta có thể chia thành ba lớp khác nhau:

- ✓ Các kỹ thuật cục bộ (Local techniques) dựa vào các thuộc tính cục bộ của các điểm và láng giềng của nó.
- ✓ Các kỹ thuật toàn thể (global techniques) phân ảnh dựa trên thông tin chung của toàn bộ ảnh (ví dụ bằng cách sử dụng lược đồ xám của ảnh – image histogram).
- ✓ Các kỹ thuật tách (split), hợp (merge) và growing sử dụng các khái niệm đồng nhất và gần về hình học.

1.3. MỘT SỐ KHÁI NIỆM CƠ BẢN

1.3.1 Điểm ảnh - Pixel

Ảnh trong thực tế là một ảnh liên tục về không gian và về giá trị độ sáng. Để có thể XLA bằng máy tính cần phải tiến hành số hoá ảnh. Trong quá trình số hoá, người ta biến đổi tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu (rời rạc hoá về không gian) và lượng hoá thành phần giá trị mà về nguyên tắc bằng mắt thường không phân biệt được hai điểm kề nhau. Trong quá trình này người ta sử dụng khái niệm *Picture element* mà ta quen gọi là Pixel - phần tử ảnh. Như vậy, một ảnh là một tập hợp các *Pixel*

1.3.2 Mức xám – Gray level

Mức xám là kết quả *sự mã hoá* tương ứng một cường độ sáng của mỗi điểm ảnh với một giá trị số - kết quả của quá trình lượng hoá. Cách mã hoá kinh điển

thường dùng 16, 32 hay 64 mức. Phổ dụng nhất là mã hoá ở mức 256, ở mức này mỗi Pixel sẽ được mã hoá bởi 8 bit.

1.3.3 Biên

Biên là một đặc tính rất quan trọng của đối tượng trong ảnh, nhờ vào biên mà chúng ta phân biệt được đối tượng này với đối tượng kia. Một điểm ảnh có thể gọi là *điểm biên* nếu ở đó có sự thay đổi đột ngột về mức xám. Tập hợp các điểm biên tạo thành *biên* hay còn gọi là đường bao ảnh.

1.3.4 Láng giềng

Trong XLA có một khái niệm rất quan trọng, đó là khái niệm *láng giềng*. Có hai loại láng giềng: *4-láng giềng* và *8-láng giềng*

4-láng giềng của một điểm (x,y) là một tập hợp bao gồm láng giềng dọc và láng giềng ngang của nó:

$$N_4((x,y)) = \{(x+1,y), (x-1,y), (x,y+1), (x,y-1)\} \quad (1.1)$$

8-láng giềng của (x,y) là một tập cha của *4-láng giềng* và bao gồm láng giềng ngang, dọc và chéo:

$$N_8((x,y)) = N_4((x,y)) \cup \{(x+1,y+1), (x-1,y-1), (x+1,y-1), (x-1,y+1)\} \quad (1.2)$$

1.3.5 Vùng liên thông

Một vùng R được gọi là *liên thông* nếu bất kỳ hai điểm (x_A, y_A) và (x_B, y_B) thuộc vào R có thể được nối bởi một đường $(x_A, y_A) \dots (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}) \dots (x_B, y_B)$, mà các điểm (x_i, y_i) thuộc vào R và bất kỳ điểm (x_i, y_i) nào đều kề sát với điểm trước (x_{i-1}, y_{i-1}) và điểm tiếp theo (x_{i+1}, y_{i+1}) trên đường đó. Một điểm (x_k, y_k) được gọi là kề với điểm (x_l, y_l) nếu (x_l, y_l) thuộc vào láng giềng trực tiếp của (x_k, y_k) .

CHƯƠNG 2 : MỘT SỐ KỸ THUẬT PHÂN ĐOẠN ẢNH

Phân đoạn (segmentation) là một quá trình chia ảnh ra các vùng con khác nhau mà trong mỗi vùng chứa các thực thể có ý nghĩa cho việc phân lớp - mỗi thực thể được xem là một đối tượng mang những thông tin đặc trưng riêng. Có rất nhiều kỹ thuật phân đoạn ảnh, trong chương này chúng tôi giới thiệu một số kỹ thuật tiêu biểu như: Phân đoạn dựa vào ngưỡng, phân đoạn dựa vào biên, phân đoạn theo miền đồng nhất. Cũng có thể thấy rằng không có một kỹ thuật phân đoạn nào là vạn năng – theo nghĩa là có thể áp dụng cho mọi loại ảnh và cũng không có một kỹ thuật phân đoạn ảnh nào là hoàn hảo.

2.1 PHÂN ĐOẠN DỰA VÀO NGƯỠNG

2.1.1 Giới thiệu chung

Biên độ của các tính chất vật lý của ảnh (như là độ phản xạ, độ truyền sáng, màu sắc ...) là một đặc tính đơn giản và rất hữu ích. Nếu biên độ đủ lớn đặc trưng cho ảnh thì chúng ta có thể dùng ngưỡng biên độ để phân đoạn ảnh. Thí dụ, biên độ trong bộ cảm biến hồng ngoại có thể phản ánh vùng có nhiệt độ thấp hay vùng có nhiệt độ cao. Đặc biệt, kỹ thuật phân ngưỡng theo biên độ rất có ích đối với ảnh nhị phân như văn bản in, đồ họa, ảnh màu hay ảnh X-quang.

Việc chọn ngưỡng trong kỹ thuật này là một bước vô cùng quan trọng, thông thường người ta tiến hành theo các bước chung như sau:

- ✓ Xem xét lược đồ xám của ảnh để xác định và khe. Nếu ảnh có nhiều đỉnh và khe thì các khe có thể sử dụng để chọn ngưỡng.
- ✓ Chọn ngưỡng T sao cho một phần xác định trước η của toàn bộ số mẫu là thấp hơn T.
- ✓ Điều chỉnh ngưỡng dựa trên xét lược đồ xám của các điểm lân cận.

- ✓ Chọn ngưỡng bằng cách xem xét lược đồ xám của những điểm thoả tiêu chuẩn đã chọn.

Một thuật toán đơn giản trong kỹ thuật này là: giả sử rằng chúng ta đang quan tâm đến các đối tượng sáng (object) trên nền tối (background), một tham số T - gọi là ngưỡng độ sáng, sẽ được chọn cho một ảnh $f[x,y]$ theo cách:

$$\begin{array}{ll} \text{If } f[x,y] \geq T & f[x,y] = \text{object} = 1 \\ \text{Else} & f[x,y] = \text{Background} = 0. \end{array}$$

Ngược lại, đối với các đối tượng tối trên nền sáng chúng ta có thuật toán sau:

$$\begin{array}{ll} \text{If } f[x,y] < T & f[x,y] = \text{object} = 1 \\ \text{Else} & f[x,y] = \text{Background} = 0. \end{array}$$

Vấn đề chính là chúng ta nên chọn ngưỡng T như thế nào để việc phân vùng đạt được kết quả cao nhất?.

Có rất nhiều thuật toán chọn ngưỡng: ngưỡng cố định, dựa trên lược đồ, sử dụng Entropy, sử dụng tập mờ, chọn ngưỡng thông qua sự không ổn định của lớp và tính thuần nhất của vùng vv... Ở đây chúng tôi đề cập đến hai thuật toán chọn ngưỡng đó là chọn ngưỡng cố định và chọn ngưỡng dựa trên lược đồ.

2.1.2 Chọn ngưỡng cố định

Đây là phương pháp chọn ngưỡng độc lập với dữ liệu ảnh. Nếu chúng ta biết trước là chương trình ứng dụng sẽ làm việc với các ảnh có độ tương phản rất cao, trong đó các đối tượng quan tâm rất tối còn nền gần như là đồng nhất và rất sáng thì việc chọn ngưỡng $T = 128$ (xét trên thang độ sáng từ 0 đến 255) là một giá trị chọn khá chính xác. Chính xác ở đây hiểu theo nghĩa là số các điểm ảnh bị phân lớp sai là cực tiểu.

2.1.3 Chọn ngưỡng dựa trên lược đồ (Histogram)

Trong hầu hết các trường hợp, ngưỡng được chọn từ lược đồ độ sáng của vùng hay ảnh cần phân đoạn. Có rất nhiều kỹ thuật chọn ngưỡng tự động xuất phát

từ lược đồ xám $\{h[b] \mid b = 0, 1, \dots, 2^B-1\}$ đã được đưa ra. Những kỹ thuật phổ biến sẽ được trình bày dưới đây. Những kỹ thuật này có thể tận dụng những lợi thế do sự làm trơn dữ liệu lược đồ ban đầu mang lại nhằm loại bỏ những dao động nhỏ về độ sáng. Tuy nhiên các thuật toán làm trơn cần phải cẩn thận, không được làm dịch chuyển các vị trí đỉnh của lược đồ. Nhận xét này dẫn đến thuật toán làm trơn dưới đây:

$$h_{smooth}[b] = \frac{1}{W} \sum_{w=-(W-1)/2}^{(W-1)/2} h_{raw}[b-w] \quad W \text{ lẻ} \quad (2.1)$$

Trong đó, W thường được chọn là 3 hoặc 5

2.1.3.1 Thuật toán đẳng liệu

Đây là kỹ thuật chọn ngưỡng theo kiểu lặp do Ridler và Calvard đưa ra. Thuật toán được mô tả như sau:

- B1: Chọn giá trị ngưỡng khởi động $\theta_0 = 2^{B-1}$
- B2: Tính các trung bình mẫu ($m_{f,0}$) của những điểm ảnh thuộc đối tượng và ($m_{b,0}$) của những điểm ảnh nền.
- B3: Tính các ngưỡng trung gian theo công thức:

$$\theta_k = \frac{m_{f,k-1} + m_{b,k-1}}{2} \quad \text{với } k = 1, 2, \dots \quad (2.2)$$

- B4: Nếu $\theta_k = \theta_{k-1}$: Kết thúc, dừng thuật toán.

Ngược lại : Lặp lại bước 2.

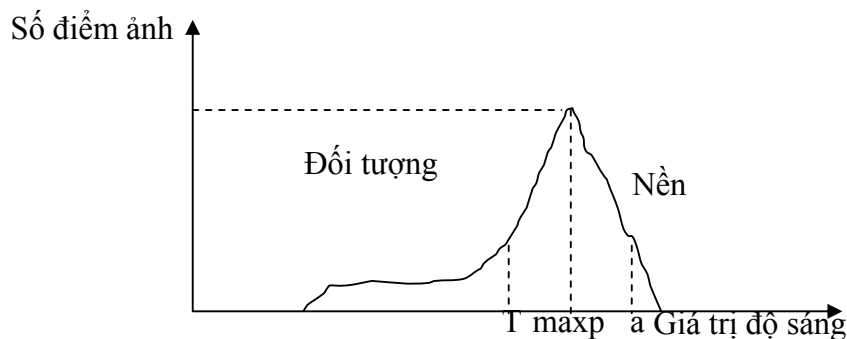
2.1.3.2 Thuật toán đối xứng nền

Kỹ thuật này dựa trên sự giả định là tồn tại hai đỉnh phân biệt trong lược đồ nằm đối xứng nhau qua đỉnh có giá trị lớn nhất trong phần lược đồ thuộc về các điểm ảnh nền. Kỹ thuật này có thể tận dụng ưu điểm của việc làm trơn được mô tả

trong phương trình (2.1). Đỉnh cực đại $\max p$ tìm được nhờ tiến hành tìm giá trị cực đại trong lược đồ. Sau đó thuật toán sẽ được áp dụng ở phía *không phải là điểm ảnh thuộc đối tượng* ứng với giá trị cực đại đó nhằm tìm ra giá trị độ sáng a ứng với giá trị phần trăm $p\%$ mà: $P(a) = p\%$, trong đó $P(a)$ là hàm phân phối xác suất về độ sáng được định nghĩa như sau:

Định nghĩa: [Hàm phân phối xác suất về độ sáng]

Hàm phân phối xác suất $P(a)$ thể hiện xác suất chọn được một giá trị độ sáng từ một vùng ảnh cho trước, sao cho giá trị này không vượt quá một giá trị độ sáng cho trước a . Khi a biến thiên từ $-\infty$ đến $+\infty$, $P(a)$ sẽ nhận các giá trị từ 0 đến 1. $P(a)$ là hàm đơn điệu không giảm theo a , do vậy $dP/da \geq 0$.



Hình 2. Minh họa thuật toán đối xứng nền

Ở đây ta đang giả thiết là ảnh có các đối tượng tối trên nền sáng. Giả sử mức là 5%, thì có nghĩa là ta phải ở bên phải đỉnh $\max p$ một giá trị a sao cho $P(a)=95\%$. Do tính đối xứng đã giả định ở trên, chúng ta sử dụng độ dịch chuyển về phía trái của điểm cực đại tìm giá trị ngưỡng T :

$$T = \max p - (a - \max p) \quad (2.3)$$

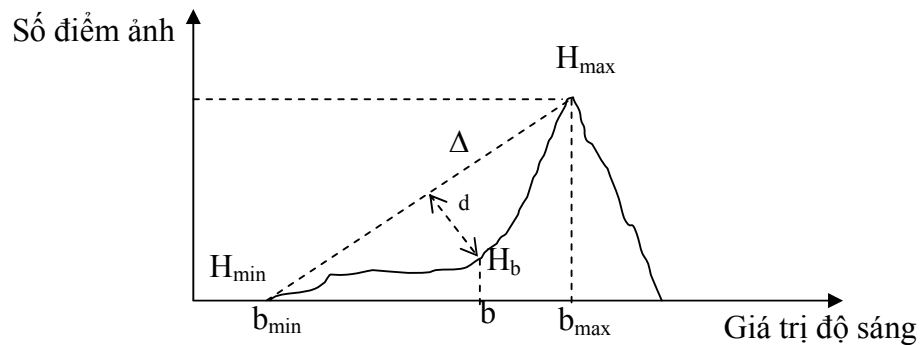
Kỹ thuật này dễ dàng điều chỉnh được cho phù hợp với tình huống ảnh có các đối tượng sáng trên một nền tối.

2.1.3.3 Thuật toán tam giác

Khi một ảnh có các điểm ảnh thuộc đối tượng tạo nên một đỉnh yếu trong lược đồ ảnh thì thuật toán tam giác hoạt động rất hiệu quả. Thuật toán này do Zack đề xuất và được mô tả như sau:

- B1: Xây dựng đường thẳng Δ là đường nối hai điểm (H_{\max}, b_{\max}) và (H_{\min}, b_{\min}) , trong đó H_{\max} là điểm có Histogram lớn nhất ứng với mức xám b_{\max} và H_{\min} là điểm có Histogram ứng với độ sáng nhỏ nhất b_{\min} .
- B2: Tính khoảng cách d từ H_b của lược đồ (ứng với điểm sáng b) đến Δ . Trong đó, $b \in [b_{\max}, b_{\min}]$.
- B3: Chọn ngưỡng $T = \text{Max}\{H_b\}$

Minh hoạ thuật toán tam giác bởi hình vẽ như sau:



Hình 3. Minh hoạ thuật toán tam giác

2.1.3.4 Chọn ngưỡng đối với Bimodal Histogram

Ngưỡng T được chọn ở tại vị trí cực tiểu địa phương của histogram nằm giữa hai đỉnh của histogram. Điểm cực đại địa phương của histogram có thể dễ dàng được phát hiện bằng cách sử dụng biến đổi chóp mũ (top hat) do Meyer đưa ra: Phụ thuộc vào tình huống chúng ta đang phải làm việc là với nhưng đối tượng sáng trên

nền tối hay đối tượng tối trên nền sáng mà phép biến đổi top hat sẽ có một trong hai dạng sau:

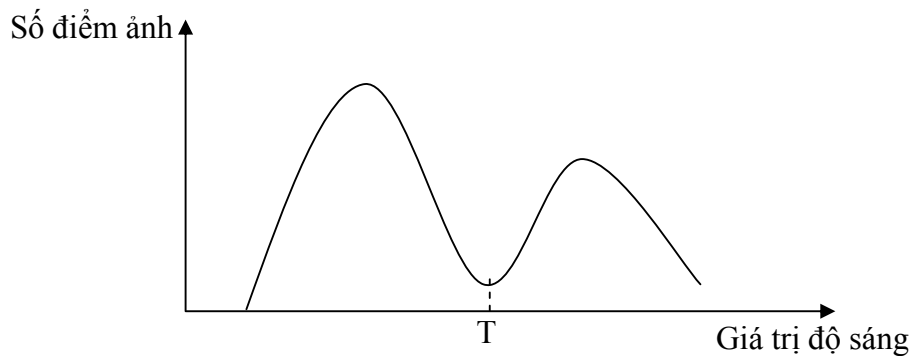
a/ Các đối tượng sáng:

$$TopHat(A, B) = A - (A \circ B) = A - \max_B(\min_B(A)) \quad (2.4)$$

b/ Các đối tượng tối:

$$TopHat(A, B) = A - (A \circ B) = A - \min_B(\max_B(A)) \quad (2.5)$$

Việc tính toán giá trị cực tiểu địa phương của histogram thì khó nếu histogram nhiều. Do đó, trong trường hợp này nên làm trơn histogram, ví dụ sử dụng thuật toán (2.1).



Hình 4. Bimodal Histogram

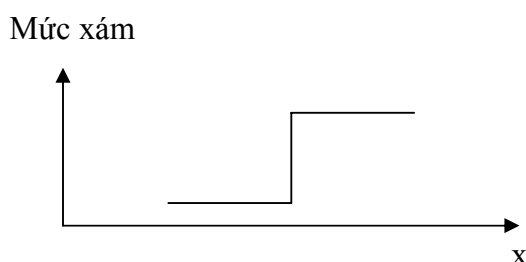
Trong một số ứng dụng nhất định, cường độ của đối tượng hay nền thay đổi khá chậm. Trong trường hợp này, histogram ảnh có thể không chứa hai thùy phân biệt rõ ràng, vì vậy có thể phải dùng ngưỡng thay đổi theo không gian. Hình ảnh được chia thành những khối hình vuông, histogram và ngưỡng được tính cho mỗi khối tương ứng. Nếu histogram cục bộ không phải là bimodal histogram thì ngưỡng được tính bằng cách nội suy ngưỡng của các khối láng giềng. Khi ngưỡng cục bộ đã có thì áp dụng thuật toán phân ngưỡng ở hình 2.1 cho khối này.

2.2 PHÂN ĐOẠN DỰA THEO ĐƯỜNG BIÊN

2.2.1 Giới thiệu chung

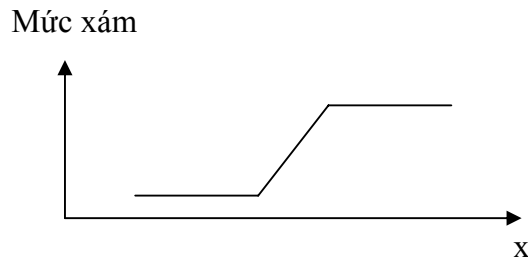
Như chúng ta đã biết, Biên là một đặc tính rất quan trọng để phân vùng các đối tượng. Có thể hình dung tầm qua trọng của biên thông qua ví dụ sau: Khi một người hoạ sĩ vẽ một cái bàn gỗ, chỉ cần phác thảo vài nét về hình dáng như cái mặt bàn, cái chân bàn mà không cần thêm các chi tiết khác, người xem đã có thể nhận ra đó là cái bàn. Vài nét phác thảo của người hoạ sĩ chính là đường biên bao quanh đối tượng. Nếu ứng dụng của ta là phân lớp nhận diện các đối tượng thì coi như nhiệm vụ đã hoàn thành. Tuy nhiên, nếu đòi hỏi thêm các chi tiết khác như vân gỗ, màu sắc, kích thước vv ... thì chừng ấy thông tin là chưa đầy đủ.

Trong toán học, người ta đưa ra khái niệm đường biên lý tưởng như sau: ***Đường biên lý tưởng là sự thay đổi giá trị cấp xám tại một vị trí xác định. Vị trí của đường biên chính là vị trí thay đổi cấp xám.*** Thể hiện của định nghĩa là hình vẽ 2



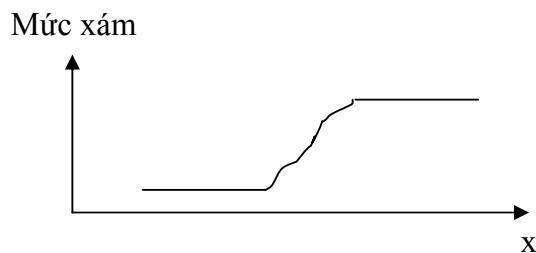
Hình 5. Đường biên lý tưởng

Một loại đường biên nữa - được gọi là đường biên bậc thang: ***Đường biên bậc thang xuất hiện khi sự thay đổi cấp xám trải rộng qua nhiều điểm ảnh. Vị trí của đường biên được xem như vị trí chính giữa của đường nối giữa cấp xám thấp và cấp xám cao.***



Hình 6. Đường biên bậc thang

Trong thực tế đường biên của chúng ta thường có dạng như sau:



Hình 7. Đường biên thực

Như đã nói ở trên, biên là một trong những đặc trưng quan trọng của ảnh, chính vì vậy mà trong nhiều ứng dụng người ta sử dụng cách phân đoạn dựa theo biên. Việc phân đoạn ảnh dựa vào biên được tiến hành qua các bước:

- Phát hiện biên và làm nổi biên
- Làm mảnh biên
- Nhị phân hoá đường biên
- Mô tả biên

2.2.2 Phát hiện biên

Phát hiện biên một cách lý tưởng là xác định được tất cả các đường bao trong các đối tượng. Có nhiều phương pháp phát hiện biên, thông thường chúng ta sử dụng phương pháp phát hiện biên trực tiếp. Phương pháp này nhằm làm nổi biên dựa vào sự biến thiên về giá trị độ sáng của điểm ảnh. Kỹ thuật chủ yếu dùng ở đây là kỹ thuật đạo hàm. Nếu lấy đạo hàm bậc nhất của ảnh ta có phương pháp

Gradient, nếu lấy đạo hàm bậc hai ta có kỹ thuật Laplace. Phương pháp này có ưu điểm là ít chịu ảnh hưởng của nhiễu, song nếu sự biến thiên của độ sáng không đột ngột thì hiệu quả đạt được là rất kém.

2.2.2.1 Kỹ thuật Gradient

Phương pháp Gradient là phương pháp dò biên cục bộ dựa vào cực đại của đạo hàm. Theo định nghĩa, Gradient là một vectơ có các thành phần biểu thị tốc độ thay đổi giá trị của điểm ảnh theo hai hướng x và y. Các thành phần của gradient được tính theo công thức:

$$\begin{aligned}\frac{\partial f(x, y)}{\partial x} &= f_x \approx \frac{f(x + dx, y) - f(x, y)}{dx} \\ \frac{\partial f(x, y)}{\partial y} &= f_y \approx \frac{f(x, y + dy) - f(x, y)}{dy}\end{aligned}\quad (2.6)$$

trong đó, dx là khoảng cách giữa các điểm theo hướng x (khoảng cách tính bằng số điểm), dy là khoảng cách giữa các điểm theo hướng y. Thực tế, người ta hay dùng với $dx = dy = 1$.

Với một ảnh liên tục $f(x, y)$, các đạo hàm riêng của nó cho phép xác định vị trí cực đại cục bộ theo hướng của biên. Thực vậy, một ảnh liên tục được biểu diễn bởi một hàm $f(x, y)$ dọc theo r với góc φ (toạ độ cực):

$$f(x, y) = f(r \cdot \cos \varphi, r \cdot \sin \varphi) \quad (2.7)$$

gradient được định nghĩa:

$$\begin{aligned}\frac{\partial f}{\partial r} &= \frac{\partial f}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial r} = f_x \cos \varphi + f_y \sin \varphi \\ \frac{\partial f}{\partial \varphi} &= \frac{\partial f}{\partial x} \frac{\partial x}{\partial \varphi} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \varphi} = -f_x r \sin \varphi + f_y r \cos \varphi\end{aligned}\quad (2.8)$$

φ là hướng của biên khi:

$$\begin{aligned}\frac{\partial f}{\partial \varphi} &= 0 \\ \Leftrightarrow -f_x r \sin \varphi + f_y r \cos \varphi &= 0 \\ \Leftrightarrow \operatorname{tg} \varphi &= \frac{f_x}{f_y} \\ \Leftrightarrow \varphi &= \operatorname{arctg} \left(\frac{f_x}{f_y} \right)\end{aligned}$$

Thực ra, đạo hàm của ảnh là không tồn tại vì $f(x,y)$ không liên tục. Ở đây, ta chỉ sử dụng mô phỏng theo ý nghĩa của đạo hàm, việc tính toán là xấp xỉ đạo hàm bằng kỹ thuật nhân chập. Trong phương pháp gradient, người ta chia nhỏ thành hai kỹ thuật (tương ứng với hai toán tử khác nhau):

- + Kỹ thuật gradient dùng toán tử gradient, lấy đạo hàm theo một hướng;
- + Kỹ thuật la bàn dùng toán tử la bàn, lấy đạo hàm theo tám hướng: Bắc, Nam, Đông, Tây, và Đông Bắc, Tây Bắc, Đông Nam, Tây Nam.

2.2.2.2 Kỹ thuật Gradient

Kỹ thuật gradient sử dụng một cặp mặt nạ H_1, H_2 trực giao (theo hai hướng vuông góc). Nếu định nghĩa g_x, g_y là gradient tương ứng theo hai hướng x, y thì biên độ của gradient tại điểm (i,j) - ký hiệu là $g(i,j)$ được tính theo công thức:

$$g(i, j) = A_0 = \sqrt{g_x^2(i, j) + g_y^2(i, j)} \quad (2.9)$$

Góc φ :

$$\varphi_r(i, j) = \arctan\left(\frac{g_x(i, j)}{g_y(i, j)}\right) \quad (2.10)$$

Có nhiều toán tử đạo hàm khác nhau đã được áp dụng. Em xin trình bày một số toán tử tiêu biểu (tương ứng là các mặt nạ khác nhau) như Toán tử Robert, toán tử Sobel, Toán tử Prewitt ...

+/ Toán tử Robert (Do Robert đề xuất năm 1965): Toán tử này là áp dụng trực tiếp của công thức đạo hàm tại điểm (x,y). Chọn cặp mặt nạ H_1, H_2 như sau:

$$H_1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad H_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

Với mỗi điểm ảnh $I(x,y)$ của I , gọi g_x, g_y tương ứng là các đạo hàm theo các hướng x và y , ta có:

$$\begin{cases} g_x(i, j) = I(i+1, j) - I(i, j) \\ g_y(i, j) = I(i, j+1) - I(i, j) \end{cases} \quad (2.11)$$

Điều bày tương đương với việc chập ảnh với hai mặt nạ H_1, H_2 :

$$\begin{cases} g_x(i, j) = I(i, j) \otimes H_1 \\ g_y(i, j) = I(i, j) \otimes H_2 \end{cases} \quad (2.12)$$

Người ta gọi H_1, H_2 là mặt nạ Robert.

Trong trường hợp tổng quát, giá trị gradient biên độ g và gradient hướng ϕ_r được tính bởi công thức (2.9), (2.10). Ngoài ra, để giảm thời gian tính toán ta cũng có thể dùng các chuẩn sau để tính $g(i,j)$:

$$A_1 = |g_x(i, j)| + |g_y(i, j)| \quad (2.13)$$

$$\text{Hoặc } A_2 = \text{Max}(|g_x(i, j)|, |g_y(i, j)|) \quad (2.14)$$

Một điểm nữa là: khi di chuyển mặt nạ trên ảnh, trường hợp gặp các điểm biên, thì coi các điểm ứng với mặt nạ ở bên ngoài ảnh có giá trị 0.

+/ Toán tử Solbel:

Toán tử Solbel sử dụng hai mặt nạ H_1, H_2 như sau:

$$H_1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad H_2 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.15)$$

Khi đó:

$$\begin{cases} g_x(i, j) = I(i, j) \otimes H_1 = \sum_{k=-1}^1 \sum_{t=-1}^1 (I(i+k, j+t) H_1(k+1, t+1)) \\ g_y(i, j) = I(i, j) \otimes H_2 = \sum_{k=-1}^1 \sum_{t=-1}^1 (I(i+k, j+t) H_2(k+1, t+1)) \end{cases} \quad (2.16)$$

Hình 3.2 minh hoạ việc xấp xỉ g_x, g_y trong toán tử Solbel

I(i-1,j-1)	I(i,j-1)	I(i+1,j-1)
I(i-1,j)	I(i,j)	I(i+1,j)
I(i-1,j+1)	I(i,j+1)	I(i+1,j+1)

Hình 3.2 Xấp xỉ g_x, g_y trong toán tử Solbel

+/- Toán tử Prewitt:

Sử dụng hai mặt nạ:

$$H_1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, H_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.17)$$

+/- Mặt nạ đẳng hướng (Isometric):

Sử dụng hai mặt nạ:

$$H_1 = \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}, H_2 = \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix} \quad (2.18)$$

Cần chú ý thêm là các chuẩn trong công thức (2.13), (2.14) đã tạo nên sự “vận xoắn” trong việc tính toán biên độ. Thực vậy, nếu g_x hoặc g_y bằng 0 thì $A_1 = A_2 = A_0$, nếu $g_x = g_y$ thì ta sẽ có $A_1 = g_x, A_2 = g_y, A_0 = g_x \sqrt{2}$. Sau khi thực hiện tính toán theo các công thức (2.12) và (2.16) ta thấy phương pháp Robert và Solbel dùng chuẩn A_1 .

Có thể nhận thấy rằng việc lấy đạo hàm một tín hiệu có xu hướng làm tăng nhiễu trong tín hiệu đó. Thực tế đã chứng minh các toán tử Sobel và Prewitt tốt hơn toán tử Robert vì chúng ít nhạy cảm với nhiễu hơn. Cũng với mục đích nghiên cứu các mặt nạ cho kết quả tốt hơn, người ta nghĩ đến việc xem xét các lân cận theo 8 hướng chính – đó chính là phương pháp Kirsh và gọi là toán tử Kirsh hay toán tử la bàn. Phần tiếp theo chúng tôi đề cập đến toán tử này.

2.2.2.3 Kỹ thuật la bàn

Toán tử la bàn đo gradient theo 8 hướng ngược chiều kim đồng hồ, mỗi hướng cách nhau 45^0 . Khi đó: gọi g_k là gradient la bàn theo hướng $\theta_k = \pi/2 + 2k\pi$, với $k = 0, 1, \dots, 7$.

Có nhiều toán tử la bàn khác nhau, ở đây ta chỉ trình bày một cách chi tiết toán tử Kirsh. Toán tử này sử dụng mặt nạ 3×3 , mặt nạ H_k ứng với hướng θ_k với $k = 0, 1, 2, \dots, 7$. Mặt nạ H_0 – cho hướng $\theta_0 = 0^0$ có dạng như sau:

$$H_0 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$$

Trên cơ sở mặt nạ gốc định nghĩa thêm 7 mặt nạ khác nhau từ H_1 đến H_7 cho 7 hướng còn lại: $45^0, 90^0, 135^0, 180^0, 225^0, 270^0, 315^0$.

$$H_1 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad H_2 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad H_3 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$H_4 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad H_5 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \quad H_6 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$$

$$H_7 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \tag{2.19}$$

Nếu ta kí hiệu A_i với $i = 0, 1, \dots, 7$ là gradient thu được theo 8 hướng bởi 8 mặt nạ thì biên độ gradient tại $I(i,j)$, ký hiệu là $g(i,j)$ sẽ được tính như sau:

$$g(i,j) = \text{Max}(|g_k(i,j)|, k=0..7) \quad (2.20)$$

Trong trường hợp tổng quát, giả sử có n hướng cách đều tương ứng với các mặt nạ W_i với $i=0, 1, \dots, n$ đối với ảnh I , khi đó:

$$A(x,y) = \text{Max}(|W_i^T I(x,y)|, i=0,1,\dots,n), \text{ thực chất đây chính là chuẩn } A_2.$$

2.2.2.4 Kỹ thuật 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 – ta gọi là phương pháp Laplace. Theo kỹ thuật này, vị trí biên của ảnh là chỗ trong ảnh có toán tử Laplace đổi dấu, hay nói cách khác là tại giao điểm của nó với trục hoành. Toán tử Laplace được định nghĩa như sau:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (2.21)$$

Toán tử Laplace dùng nhiều kiểu mặt nạ khác nhau để xấp xỉ rời rạc đạo hàm bậc hai. Dưới đây là ba kiểu mặt nạ hay dùng:

$$H_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad H_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad H_3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad (2.22)$$

Để thấy rõ việc xấp xỉ đạo hàm bậc hai trong không gian rời rạc bởi mặt nạ H_1 , ta xét chi tiết cách tính đạo hàm bậc hai như sau:

$$\frac{\partial^2 f}{\partial x^2} = 2f(x,y) - f(x-1,y) - f(x+1,y) \quad (2.23)$$

$$\frac{\partial^2 f}{\partial y^2} = 2f(x, y) - f(x, y-1) - f(x, y+1) \quad (2.24)$$

Lúc đó:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = -f(x-1, y) - f(x, y-1) + 4f(x, y) - f(x, y+1) - f(x+1, y) \quad (2.25)$$

Công thức (2.25) tương đương với kết quả nhân chập ảnh $f(x, y)$ với mặt nạ H_1 . Tương tự, ta cũng chứng minh được cách xấp xỉ đạo hàm bậc hai ảnh $f(x, y)$ bởi các mặt nạ H_2 và H_3 .

Trong kỹ thuật Laplace, điểm biên được xác định bởi điểm cắt điểm không. Điểm không là duy nhất cho nên kỹ thuật này thường cho đường biên mảnh - tức là đường biên có độ rộng khoảng 1 pixel. Tuy nhiên, do đạo hàm bậc hai thường không ổn định nên bản đồ biên của ảnh được xác định bởi kỹ thuật Laplace thường chứa nhiễu.

Hình ảnh tiếp theo minh họa các kỹ thuật phát hiện biên.



(a) Ảnh gốc



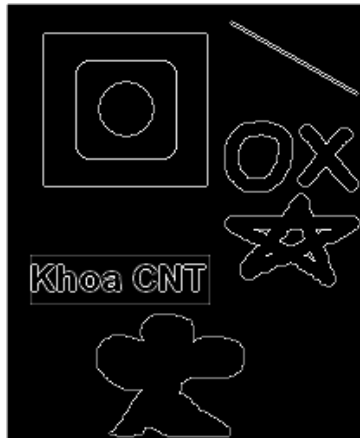
(b) Robert



(c) *Sobel*



(d) *Prewitt*



(e) *Laplace H1*



(f) *Laplace H2*

Hình 8. Minh họa một số phương pháp phát hiện biên

2.2.3 Làm mảnh biên

Làm mảnh biên thực chất là làm nổi biên với độ rộng chỉ 1 pixel. Chúng ta cũng đã biết rằng chỉ có kỹ thuật Laplace mới cho biên có độ rộng 1 pixel trong khi các kỹ thuật khác thì không hoàn toàn như thế. Vấn đề đặt ra là sau khi thu được bản đồ biên của ảnh chúng ta cần phải làm mảnh biên.

Có rất nhiều kỹ thuật làm mảnh biên đối tượng nói chung hoặc mảnh biên chữ nói riêng, ở đây chúng tôi trình bày hai thuật toán làm mảnh biên chữ, đó là: kỹ thuật “Loại bỏ các điểm không cực đại” và kỹ thuật do Sherman đề xuất.

+ Kỹ thuật loại bỏ các điểm không cực đại:

Giả sử ảnh $I(x,y)$ gồm gradient hướng và gradient biên độ (còn gọi là bản đồ hướng và bản đồ biên độ). Với mỗi điểm ảnh $I(x,y)$, ta xác định các điểm lân cận của nó theo hướng gradient, gọi các điểm đó là $I(x_1, y_1)$ và $I(x_2, y_2)$. Nếu $I(x,y)$ lớn hơn cả $I(x_1, y_1)$ và $I(x_2, y_2)$ thì giá trị của $I(x,y)$ sẽ được bảo toàn, ngược lại ta gán giá trị của nó bằng 0 và xem như bị loại bỏ khỏi biên.

+ Kỹ thuật làm mảnh biên chữ do Sherman đề xuất (về sau được Fraser cải tiến và áp dụng cho ảnh nhị phân). Kỹ thuật này được mô tả tóm tắt như sau:

Tại mỗi vị trí của số, phần tử trung tâm sẽ được xoá (đổi thành trắng) nếu nó thoả mãn một trong hai điều kiện sau:

- * Nó là điểm đen duy nhất kết nối với hai điểm đen không kề nhau.

- * Nó là điểm đen có duy nhất một lân cận cũng là điểm đen ngoại trừ không tồn tại một chuyển đổi nào tại phần tử trước nó.

2.2.4 Nhị phân hoá đường biên

Nhị phân hóa đường biên là giai đoạn then chốt trong quá trình trích chọn vì nó xác định đường bao nào thực sự cần và đường bao nào có thể loại bỏ. Nói chung, người ta thường nhị phân hóa đường biên theo cách thức làm giảm nhiễu hoặc tránh hiện tượng kéo sợi trên ảnh. Điều này cũng giải thích tại sao phân đoạn dựa theo biên có hiệu quả khi ảnh có độ tương phản tốt. Trong trường hợp ngược lại, có thể sẽ bị mất một phần đường bao hay đường bao có chân, không khép kín, v.v., do đó sẽ bất lợi cho biểu diễn sau này. Một phương pháp hay được dùng là chọn ngưỡng thích nghi. Với cách chọn này, ngưỡng sẽ phụ thuộc vào hướng của gradient nhằm làm giảm sự xoắn của biên. Đầu tiên, người ta định ra một ngưỡng nào đó và sau đó sử dụng một hệ số thích nghi thông qua lời giải toán tử đạo hàm theo hướng tìm được để tinh chỉnh.

2.2.5 Mô tả biên

Khi đã có bản đồ biên ảnh, ta cần phải biểu diễn nó dưới dạng thích hợp phục vụ cho việc phân tích và làm giảm lượng thông tin dùng để miêu tả, lưu trữ đối tượng. Người ta thường thực hiện theo nguyên tắc: tách riêng từng biên và gán cho mỗi biên một mã.

Có rất nhiều phương pháp miêu tả biên, mỗi phương pháp thích hợp với một loại ứng dụng riêng. Tuy nhiên, nhìn chung các biên sẽ được làm rõ hơn thông qua các thao tác: loại bỏ đường biên hở, khép kín đường biên, loại bỏ các chân rết bám theo đường biên vv...

Thông thường, các cấu trúc cơ sở mã hoá đường biên bao gồm 4 loại: điểm, đoạn thẳng, cung và đường cong. Tuy nhiên, nếu ta biểu diễn đường biên bởi các điểm thì rất đơn giản về mặt tính toán nhưng lại nghèo nàn về mặt cấu trúc và không cô đọng. Ngược lại, nếu biểu diễn biên bởi đường cong đa thức bậc cao thì cấu trúc dữ liệu rất cô đọng nhưng độ phức tạp tính toán lại khá lớn. Do đó, tùy từng loại ứng dụng cụ thể và từng bài toán cụ thể mà chúng ta có thể chọn cách mã hoá đường biên theo kiểu nào. Dưới đây, chúng tôi trình bày một số phương pháp mã hoá đường biên hay dùng.

2.2.5.1 Mã hoá theo toạ độ Đêcác

Đường biên của ảnh được biểu diễn bởi một danh sách các điểm ảnh tạo nên đường bao. Gọi C là đường bao ảnh, $C(i,j)$ là các điểm thuộc C . Cách biểu diễn này rất đơn giản, việc tính toán khá nhanh nhưng có nhược điểm là không làm giảm tải được lượng thông tin. Việc mã hoá sử dụng kỹ thuật tìm kiếm thông tin theo chiều sâu trên cây. Nếu áp dụng một cách đơn thuần kỹ thuật này ta sẽ thu được một đường biên có tồn tại một số điểm xuất hiện hơn một lần. Để làm mịn biên – nghĩa là mỗi điểm trên biên chỉ xuất hiện một lần chúng ta sẽ phối hợp với việc kiểm tra 8 liên thông.

Thuật toán Contour Following được mô tả như sau:

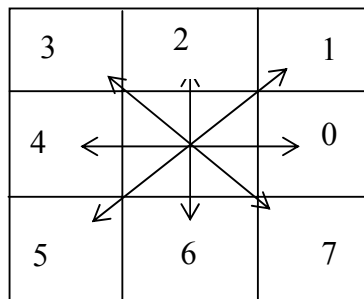
```

Void CountFoll (Pic, Depth)
{
    For each point I(x,y) do
        { If I(x,y)  $\in$  C then
            {Root  $\leftarrow$  I(x,y)
              KQ  $\leftarrow$  CountFoll (Root, 0)
              If KQ then Dem  $\leftarrow$  Dem+1.
            }
        }
}

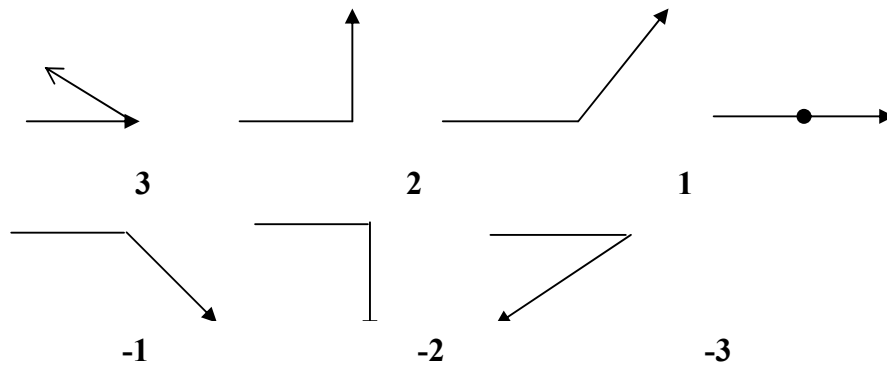
```

2.2.5.2 Mã hoá Freeman

Phương pháp này biểu diễn đường biên bằng việc sử dụng vị trí tương đối của điểm trên biên với điểm trước. Nguyên tắc mã hoá như sau: sử dụng mặt nạ ở hình < > để xác định mã của mỗi điểm trong 8 liên thông so với điểm ở tâm, sau đó từ một điểm đã cho trên biên người ta mã hoá đường biên bằng cách đi theo nó. Thông thường người ta hay mã hoá đường biên theo góc giữa các cung – xem hình < >



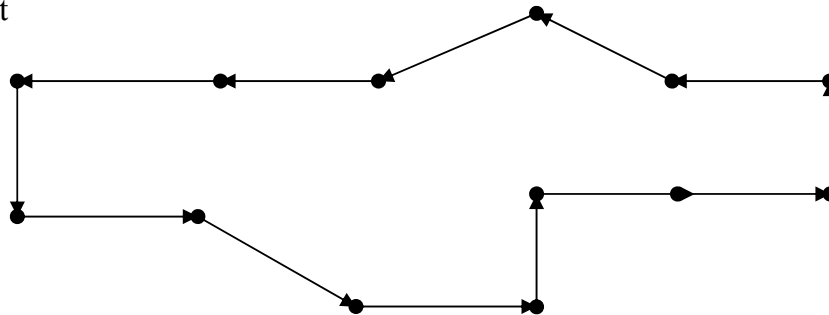
Hình 9. Liên thông và mã hướng tương ứng



Hình 10. Mã hoá theo góc

Giả sử ta có bản đồ biên như sau:

Xuất phát



Nếu mã hoá theo cung thì mã đường biên là $\{6 \ 0 \ 7 \ 0 \ 2 \ 0 \ 0 \ 2 \ 4 \ 3 \ 5 \ 4 \ 4\}$, còn nếu mã hoá theo góc thì ta có $\{2 \ 2 \ -1 \ 1 \ 2 \ -2 \ 0 \ 2 \ 2 \ -1 \ 2 \ -1 \ 0\}$

2.2.5.3 Xấp xỉ bởi đoạn thẳng

Ngược với hai cách mã hoá ở trên, kỹ thuật mã hoá bởi đoạn thẳng không cho phép khôi phục tất cả các thông tin chứa đựng trong đường biên nhưng lại có thể xấp xỉ nó bởi đoạn thẳng với độ chính xác phụ thuộc vào người dùng. Thuật toán xấp xỉ bởi đoạn thẳng được mô tả như sau:

- B1: Chọn điểm xuất phát R .
- B2: Nối R với điểm đang xét P_c – ta được đoạn thẳng RP_c .

- B2: Tính $d_j = \text{Max} \{d_i - \text{khoảng cách từ các điểm } P_i \text{ nằm giữa } R \text{ và } P_c \text{ đến đoạn thẳng } RP_c\}$
- B3: Nếu $d_j > \theta$ - ngưỡng cho trước, còn gọi là độ chính xác của xấp xỉ thì phân đoạn RP_c thành hai đoạn RP_i và P_iP_c . Sau đó, lặp lại bước 2.

Ngược lại, nếu $d_j < \theta$ - tức là đoạn thẳng đang xét “rất gần” với cung của biên thì dừng thuật toán.

Thuật toán sẽ đạt hiệu quả rất cao nếu chúng ta chọn được độ chính xác của xấp xỉ hợp lý. Độ chính xác càng thấp, thông tin mô tả càng cô đọng. Cũng trong phương pháp xấp xỉ bởi đoạn thẳng, có một cách tiếp cận khác với phương pháp trên, đó là phép biến đổi Hough [Tr 143 - 147].

2.3. PHÂN ĐOẠN THEO MIỀN ĐỒNG NHẤT

2.3.1 Giới thiệu

Giả sử rằng một miền ảnh X phải được phân thành N vùng khác nhau: R_1, \dots, R_N và nguyên tắc phân đoạn là một vị từ của công thức $P(R)$. Việc phân đoạn ảnh chia tập X thành các tập con $R_i, i = 1..N$ phải thoả mãn:

- Các vùng $R_i, i=1..N$ phải lấp kín hoàn toàn ảnh:

$$X = \bigcup_{i=1}^N R_i \quad (2.26)$$

- Hai vùng khác nhau phải là những tập hợp rời nhau:

$$R_i \cap R_j = \emptyset \text{ với } i \neq j \quad (2.27)$$

- Mỗi vùng R_i phải có tính đồng nhất:

$$P(R_i) = \text{TRUE} \text{ với } i = 1..N \quad (2.28)$$

- Nếu R_i, R_j là hai vùng rời nhau thì $(R_i \cup R_j)$ phải là một vùng ảnh không đồng nhất:

$$P(R_i \cup R_j) = \text{FALSE} \text{ với } i \neq j \quad (2.29)$$

Kết quả của việc phân vùng ảnh phụ thuộc vào dạng của vị từ P và các đặc trưng được biểu diễn bởi vector đặc trưng. Thường thì vị từ P có dạng $P(R, X, t)$, trong đó X là vector đặc trưng gắn với một điểm ảnh và t là một tập hợp các tham số (thường là các ngưỡng). Trong trường hợp đơn giản nhất, vector đặc trưng X chỉ chứa giá trị mức xám của ảnh $I(k, l)$ và vector ngưỡng chỉ gồm một ngưỡng T. Một nguyên tắc phân đoạn đơn giản có công thức:

$$P(R): f(k, l) < T \quad (2.30)$$

Trong trường hợp các ảnh màu, vector đặc trưng X có thể là ba thành phần ảnh RGB $[f_R(k, l), f_G(k, l), f_B(k, l)]^T$. Lúc đó luật phân ngưỡng có dạng:

$$P(R, X, t): ((f_R(k, l) < T_R) \& \& (f_G(k, l) < T_R) \& \& (f_B(k, l) < T_R)) \quad (2.31)$$

2.3.2 Phương pháp tách cây tứ phân

Phương pháp tách cây tứ phân dựa trên nguyên tắc kiểm tra tính hợp thức của tiêu chuẩn đồng nhất một cách tổng thể trên miền lớn. Nếu tiêu chuẩn được thỏa việc phân đoạn coi như kết thúc. Trong trường hợp ngược lại, chia miền đang xét thành 4 miền nhỏ hơn, áp dụng đệ quy bằng phương pháp trên cho mỗi miền nhỏ hơn cho đến khi tất cả các miền đều thỏa mãn tiêu chuẩn đồng nhất.

Thuật toán được mô tả như sau:

Procedure PhanDoan(Mien)

Begin

If miền đang xét không thỏa Then

Begin

Chia miền đang xét thành 4 miền: Z1, Z2, Z3, Z4

For i=1 to 4 Do PhanDoan(Zi)

End

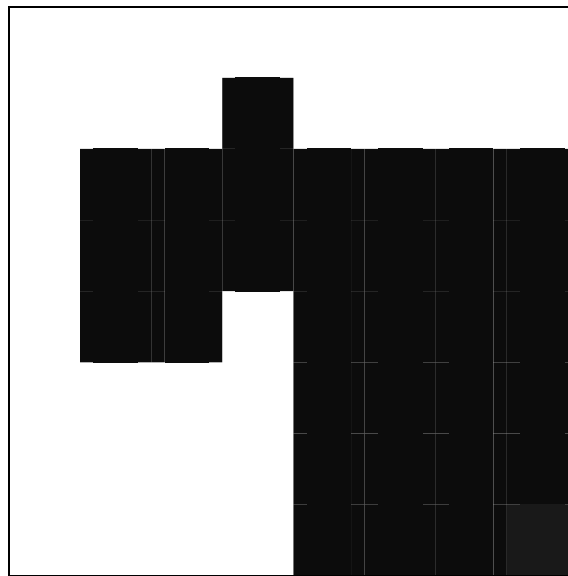
Else Exit

End;

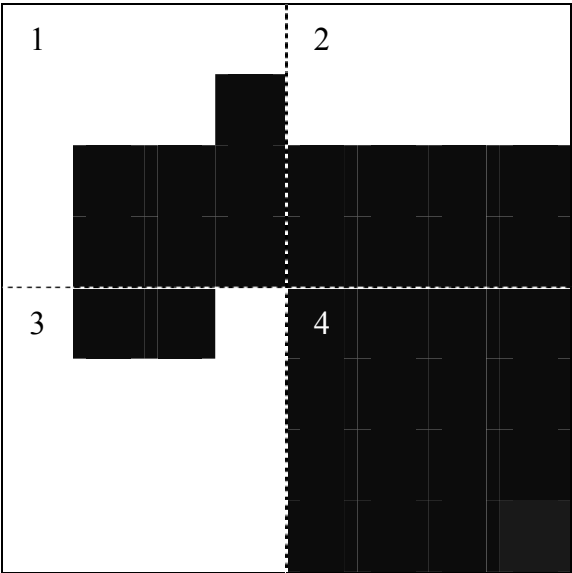
Thuật toán này tạo nên một cây mà mỗi nút cha có 4 nút con ở mọi mức, trừ mức ngoài cùng. Vì thế cây này có tên là cây tứ phân. Gốc của cây là ảnh ban đầu, một vùng thoả tiêu chuẩn tạo nên một nút lá, nếu không sẽ tạo nên một nút nhánh.

Giả sử chọn tiêu chuẩn phân vùng là màu sắc và quy ước mọi điểm của vùng là màu trắng sẽ tạo nên một nút lá trắng và tương tự như vậy với nút lá đen. Nút màu ghi có nghĩa là vùng không thuần nhất và phải tiếp tục chia.

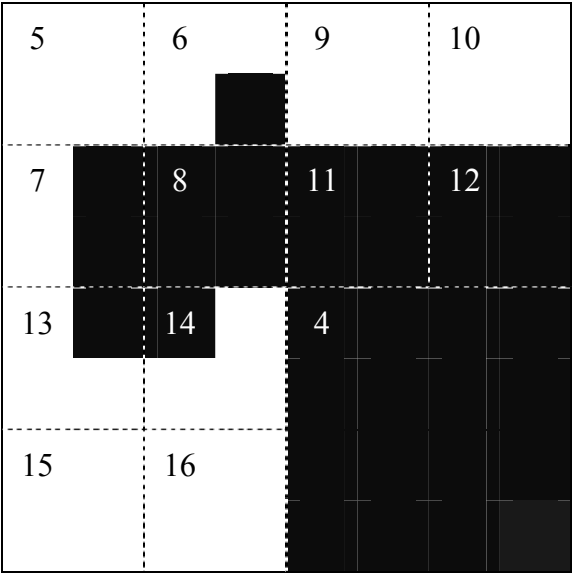
Hình 4.1 a-e minh họa thuật toán tách cây tứ phân: ảnh gốc (a) được chia thành 4 phần được kết quả phân mức 1 (b), tiếp tục thực hiện đối với các phần nhỏ, ta được phân mức 2, 3.



a) Ảnh gốc



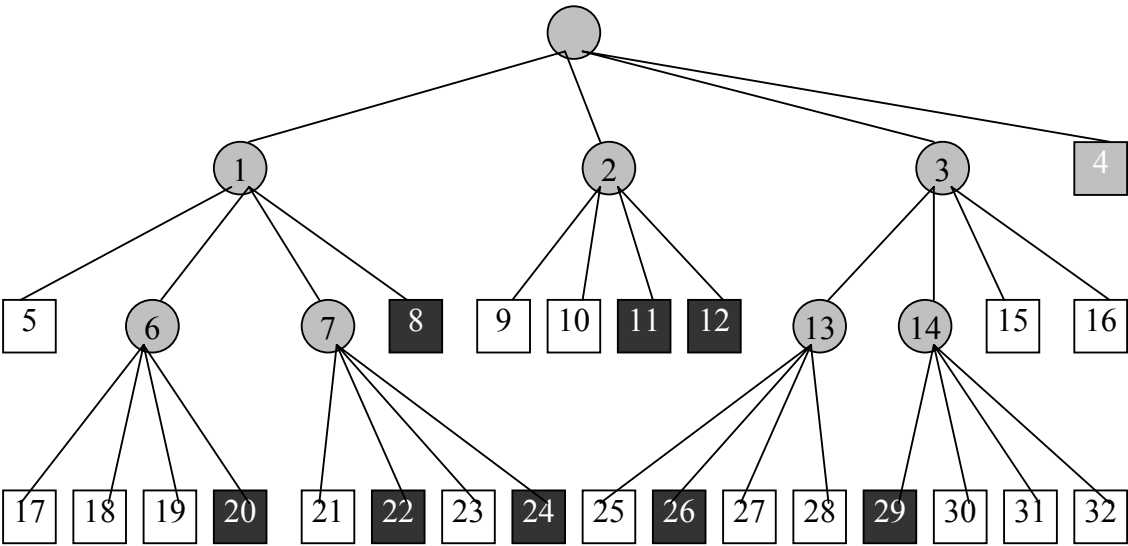
b) Phân mức 1



c) Phân mức 2

5	17	18	9	10
	19	20		
21	22	8	11	12
23	24			
25	26	29	30	4
27	28	31	32	
15	16			

d) Phân mức 3



e) Cây tương ứng

Hình 11. Phương pháp tách cây tứ phân

2.3.3 Phương pháp phân vùng bởi hợp

Phương pháp phân vùng bởi hợp thao tác ngược lại với phương pháp tách cây tứ phân, nghĩa là xuất phát từ các miền nhỏ nhất – các điểm ảnh rồi hợp chúng lại nếu thỏa mãn tiêu chuẩn đề ra để được miền đồng nhất lớn hơn. Tiếp tục với các miền thu được cho đến khi ta không thể hợp nhất chúng với nhau nữa, lúc này số miền còn lại chính là các phân vùng của ảnh. Việc hợp nhất hai miền phải thỏa mãn hai nguyên tắc sau:

- Hai vùng phải kế cận.
- Hai vùng phải đáp ứng tiêu chuẩn, như cùng màu, cùng mức xám hay cùng kết cấu vv ...

Giả sử vùng R_i có n điểm, lúc đó giá trị trung bình m_i và độ lệch tiêu chuẩn σ_i được tính theo công thức:

$$m_i = \frac{1}{n} \sum_{(k,l) \in R_i} I(k,l) \quad (2.31)$$

$$\sigma_i = \sqrt{\frac{1}{n} \sum_{(k,l) \in R_i} (I(k,l) - m_i)^2} \quad (2.32)$$

Hai vùng R_1 và R_2 có thể hợp thành một vùng nếu $|m_1 - m_2| < T$ và điểm $I(k, l)$ sẽ được hợp với vùng R_i nếu $|I(k,l) - m_i| < T$, với T là một ngưỡng.

Đầu tiên chúng ta cố gắng hợp điểm (k, l) với một trong các vùng lân cận R_i . Nếu việc hợp không thành công thì ta hợp với các vùng khác đã có. Nếu vẫn không thành công hoặc không có vùng lân cận tồn tại thì điểm này được coi là một vùng mới.

Sau khi hợp nhất (k,l) vào vùng R thì ta phải cập nhật lại giá trị trung bình và độ lệch tiêu chuẩn:

$$m_i' = \frac{1}{n+1} (I(k, l) + n * m_i) \quad (2.33)$$

$$\sigma_i^2 = \sqrt{\frac{1}{n+1} (n\sigma_i^2 + \frac{n}{n+1} [I(k, l) - m_i]^2)} \quad (2.34)$$

Nếu có nhiều hơn một vùng lân cận thỏa mãn thì hợp điểm (k, l) với vùng R_i sao cho sự khác biệt $|I(k, l) - m_i|$ nhỏ nhất.

Cũng trong phương pháp pháp phân vùng bởi hợp, có một cách tiếp cận khác với kỹ thuật trên, đó là phương pháp phân vùng dựa vào đồ thị. Phân vùng dựa trên đồ thị tìm cách hợp nhất hai miền R_i và R_j theo tính chất so sánh giữa hai cặp miền. Thuật toán này được chúng tôi trình bày chi tiết ở chương 3.

2.3.4 Phương pháp tổng hợp

Hai phương pháp vừa xét ở trên có một số nhược điểm. Phương pháp tách tạo nên một cấu trúc phân cấp và thiết lập mối quan hệ giữa các vùng. Tuy nhiên nó thực hiện việc chia quá chi tiết. Phương pháp hợp cho phép làm giảm số miền liên thông xuống tối thiểu nhưng cấu trúc hàng ngang dần trải, không cho ta thấy mối liên hệ giữa các vùng. Chính vì nhược điểm này mà ta nghĩ đến phương pháp phối hợp cả 2 phương pháp. Trước tiên dùng phương pháp tách để tạo nên cây tứ phân, phân đoạn theo hướng từ gốc đến lá. Tiếp theo tiến hành duyệt cây theo chiều ngược lại và hợp các vùng có cùng tiêu chuẩn. Với phương pháp này ta thu được miêu tả cấu trúc của ảnh với các miền liên thông có kích thước tối đa

Giải thuật trên gồm một số bước sau:

1. Kiểm tra tiêu chuẩn đồng nhất

1.1. Nếu không thỏa và số điểm trong vùng lớn hơn một điểm, tách làm 4 vùng (trên, dưới, trái, phải) bằng cách gọi đệ quy. Nếu kết quả tách xong và không tách được nữa chuyển sang bước ii.

1.2. Nếu tiêu chuẩn đồng nhất là thoả thì tiến hành hợp vùng và cập nhật giá trị trung bình cho vùng.

2. Hợp vùng: Cần kiểm tra 4 lân cận đã nêu trên. Có thể có nhiều vùng thoả mãn khi đó ta chọn vùng tối ưu rồi tiến hành hợp.

Phương pháp này thu được kết quả số vùng là nhỏ hơn phương pháp tách và ảnh được làm trơn hơn.

CHƯƠNG 3 : PHÂN ĐOẠN ẢNH DỰA VÀO ĐỒ THỊ

Phân đoạn ảnh dựa vào đồ thị là một phương pháp tiếp cận khá hiện đại dựa trên thuộc tính non-local của ảnh đầu vào. Phương pháp này phát hiện ra biên giữa hai vùng của ảnh bằng cách so sánh sự khác nhau giữa nội vùng (inter-component) với sự khác nhau với các vùng khác. Thuật toán phân đoạn dựa vào đồ thị tuân theo chiến lược tham lam, có thời gian chạy gần như tuyến tính, nhưng vẫn đảm bảo được việc phân đoạn chính xác và hiệu quả.

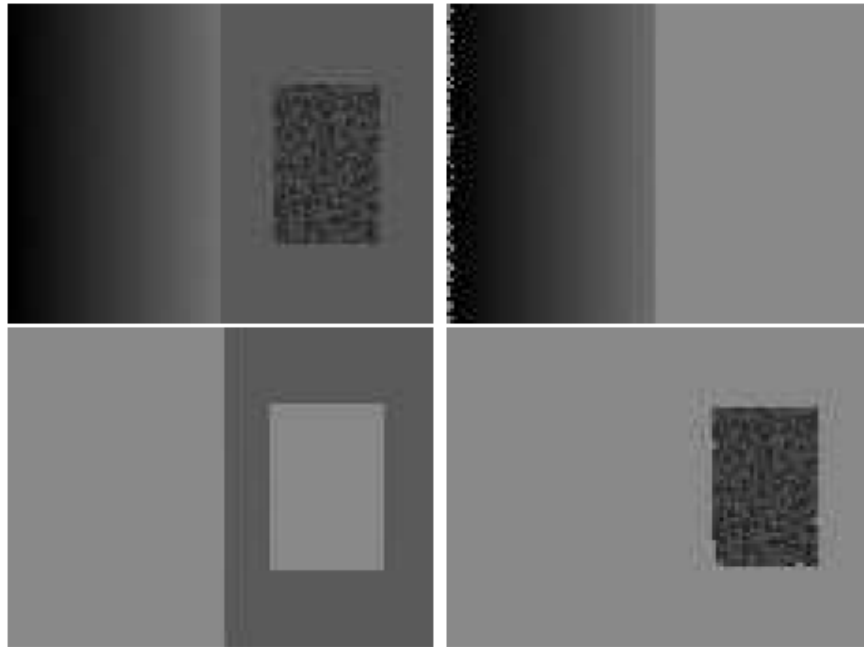
3.1 Giới thiệu

Các phương pháp phân đoạn ảnh cổ điển đều có chung một nhược điểm là chạy rất chậm trong các ứng dụng XLA và hầu như không nắm bắt được các thuộc tính non-local quan trọng của ảnh. Vì vậy, hầu hết các nghiên cứu của những năm gần đây đều có xu hướng tìm kiếm một kỹ thuật phân đoạn có khả năng xử lý trong cơ sở dữ liệu ảnh lớn một cách nhanh chóng, chính xác và hiệu quả. Kỹ thuật phân đoạn dựa vào đồ thị được mô tả ở đây không những vừa nắm bắt được các đặc tính non-local mà độ phức tạp tính toán chỉ là $O(n \log n)$ đối với bức ảnh có n điểm ảnh (pixel).

Giống như các phương pháp phân cụm cổ điển, phương pháp này cũng dựa trên việc chọn các cạnh từ một đồ thị. Đồ thị này được xây dựng bằng cách coi mỗi điểm ảnh là một đỉnh, hai điểm ảnh kề nhau thì được nối bởi một cạnh vô hướng, trọng số trên một cạnh thể hiện sự khác nhau giữa hai điểm ảnh. Tuy nhiên, phương pháp này thực hiện việc điều chỉnh sự phân đoạn dựa vào mức độ thay đổi giữa các miền lân cận của ảnh.

Lấy một ví dụ đơn giản thể hiện việc nắm bắt được các đặc tính non-local của phương pháp này. Hãy để ý vào ảnh phía trên bên trái của hình []; hầu hết ta đều nói rằng bức ảnh này có ba miền phân biệt: một hình chữ nhật ở nửa bên trái, một hình

chữ nhật đặc ở giữa nửa bên phải và phần bao quanh hình chữ nhật đặc này. Vì sao ta khẳng định được như thế? Chắc chắn đó là thuộc tính quan trọng của sự tri giác (perceptually) và chúng tôi tin rằng các đặc trưng này cũng sẽ được nắm bắt bởi thuật toán phân đoạn.



Hình 12. Ví dụ về nhận dạng các vùng ảnh

Phương pháp phân đoạn dựa vào đồ thị sẽ tìm dấu hiệu đường biên giữa hai vùng bằng cách so sánh hai đại lượng: một là dựa vào cường độ khác nhau dọc theo đường biên và hai là dựa vào cường độ khác nhau giữa các điểm ảnh với mỗi vùng.

3.2 Phân đoạn dựa vào đồ thị

Cho $G = (V, E)$ là một đồ thị vô hướng với các đỉnh $v_i \in V$, là tập hợp các phần tử cần được phân đoạn và các cạnh $(v_i, v_j) \in E$, tương ứng với các cặp đỉnh lân cận nhau. Mỗi cạnh $(v_i, v_j) \in E$ có một trọng số tương ứng, trọng số là một số không âm đo sự khác nhau giữa hai phần tử lân cận v_i và v_j , ký hiệu $w(v_i, v_j)$. Ở đây trọng số của cách cạnh đo sự khác nhau giữa hai điểm nối bởi cạnh đó (có nhiều mức độ khác nhau: màu sắc, vị trí, sự vận động hoặc các thuộc tính khác).

Như vậy phân đoạn một bức ảnh là việc phân chia V thành các thành phần, mà mỗi thành phần (hoặc miền) $C \in V$ tương đương với một thành phần liên thông trong đồ thị $G' = \langle V, E' \rangle$, $E' \subseteq E$.

3.3 Tính chất của so sánh cặp miền

Để có thể dễ dàng định lượng dấu hiệu của một đường biên giữa hai vùng trong ảnh, chúng ta định nghĩa một tính chất D . Tính chất này dựa vào độ đo sự khác nhau giữa các phần tử dọc theo một đường biên của hai thành phần liên quan nhằm đo sự khác nhau giữa các phần tử lân cận trong mỗi thành phần. Kết quả là so sánh sự khác nhau giữa nội vùng (inter-component) với sự khác nhau với các vùng khác.

Trước hết, ta định nghĩa độ-khác-nội vùng (internal difference) và độ-khác-giữa-hai-vùng (difference between two components).

Độ-khác-nội-vùng (internal difference) của một thành phần $C \subseteq V$ là trọng số lớn nhất trong cây tọa nhánh tối thiểu của thành phần đó, kí hiệu $Int(C)$. Khi đó:

$$Int(C) = \max_{e \in MST(C, E)} w(e) \quad (3.1)$$

Độ-khác-giữa-hai-vùng (difference between two components) $C_1, C_2 \subseteq V$, là trọng số nhỏ nhất của các cạnh nối giữa hai vùng, kí hiệu là $Dif(C_1, C_2)$. Khi đó:

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w((v_i, v_j)) \quad (3.2)$$

Nếu không có cạnh nối nào giữa C_1 và C_2 thì đặt $Dif(C_1, C_2) = \infty$. Độ đo sự khác nhau này về nguyên lý thì vẫn có vẻ mơ hồ, vì nó chỉ phản ánh được cạnh có trọng số nhỏ nhất nối giữa hai thành phần. Điều này cũng phản ánh rất rõ trong quá trình chạy thử nghiệm.

Một khái niệm có liên quan trong định nghĩa về tính chất D là giá trị khác-nội-vùng nhỏ nhất, kí hiệu $MInt$. Giá trị $MInt$ được định nghĩa như sau:

$$MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1), Int(C_1) + \tau(C_2)) \quad (3.3)$$

Hàm ngưỡng τ điều khiển mức độ khác nhau giữa hai thành phần, sao cho giá trị này phải lớn hơn các giá trị khác-nội-vùng của các thành phần để nhằm mục đích nhận ra đường biên giữa chúng. Đối với các thành phần nhỏ, $Int(C)$ là ko đủ tốt để ước lượng các đặc tính của dữ liệu. Trong một số trường hợp khi $|C| = 1$, thì $Int(C) = 0$ với $|C|$ là kích thước của thành phần C . Khi đó chúng ta sử dụng một hàm ngưỡng dựa trên kích thước của thành phần:

$$\tau(C) = k / |C| \quad (3.4)$$

với k là một tham số hằng. Trong thực tế thì k được chọn không nhỏ hơn kích thước của thành phần nhỏ nhất.

Lúc này tính chất so sánh giữa hai cặp miền C_1 và C_2 , kí hiệu $D(C_1, C_2)$ được định nghĩa như sau:

$$D(C_1, C_2) = \begin{cases} \text{true if } D(C_1, C_2) > MInt(C_1, C_2) \\ \text{false otherwise} \end{cases} \quad (3.5)$$

3.4 Thuật toán và các tính chất

Trong mục này chúng tôi đưa ra một thuật toán phân đoạn sử dụng tiêu chuẩn quyết định D đã mô tả ở trên. Ta sẽ chỉ ra rằng phân đoạn bằng thuật toán này sẽ tuân theo các thuộc tính không quá thô (too coarse) và cũng không quá mịn (too fine), theo các định nghĩa sau đây.

3.4.1 Định nghĩa 1

Một phân đoạn được xem là *quá mịn* nếu tồn tại một số cặp miền $C_1, C_2 \in S$ mà giữa hai miền này ko có dấu hiệu của đường biên

Để định nghĩa được những khái niệm bổ sung cho phân đoạn quá thô, chúng tôi đưa ra khái niệm tinh chỉnh (refinement) của một phân đoạn.

Cho hai phân đoạn S và T của cùng một tập cơ sở, ta nói rằng T là một tinh chỉnh (refinement) của S khi mỗi thành phần của T được chứa trong (hoặc bằng) một số thành phần của S . Và ta cũng nói rằng T là một tinh chỉnh đúng (proper refinement) của S khi $T \neq S$.

Chú ý rằng nếu T là tinh chỉnh đúng của S thì T có thể được chứa bởi một hoặc một số các miền trong S và S được gọi là thô hơn T .

3.4.2 Định nghĩa 2

Một phân đoạn được xem là *quá thô* khi tồn tại một tinh chỉnh đúng của S mà phân đoạn đó vẫn chưa là quá mịn.

Vấn đề đặt ra là liệu có phải luôn luôn tồn tại phân đoạn không quá thô cũng không quá mịn hay không? Và nếu tồn tại thì phân đoạn đó có là duy nhất không?.

Thực tế cho thấy là nói chung luôn có thể có nhiều hơn một phân đoạn không quá thô cũng không quá mịn, do đó phân đoạn này là không duy nhất. Đây là một tính chất đặc biệt của phân đoạn ảnh dựa trên đồ thị và được chứng minh chi tiết ở mục 3.4.3

3.4.3 Tính chất 1

Với một đồ thị hữu hạn $G = (V, E)$ bất kỳ luôn tồn tại một số phân đoạn S không quá thô mà cũng không quá mịn.

Chứng minh: Chúng ta dễ dàng nhận thấy là tính chất này đúng. Thật vậy, nếu phân đoạn mà tất cả các phần tử đều nằm trong một thành phần, thì phân đoạn này là không quá mịn, vì nó chỉ có đúng một thành phần (định nghĩa 1). Nếu mà phân đoạn này cũng không quá thô thì coi như xong. Ngược lại, theo định nghĩa 2, thì sẽ có một tinh chỉnh đúng mà ko quá mịn. Lấy một trong số các tinh chỉnh đó và lặp lại thủ tục này cho đến khi chúng ta sẽ thu được một phân đoạn không quá thô.

Trở lại với thuật toán phân đoạn dựa trên đồ thị, thuật toán này gần với thuật toán Kruskal xây dựng cây tủa nhánh tối thiểu của một đồ thị. Độ phức tạp của thuật toán là $O(m \log m)$, trong đó m là số cạnh của đồ thị.

3.4.4 Thuật toán 1

Thuật toán phân đoạn

Input: Đồ thị $G = (V, E)$, gồm n đỉnh và m cạnh.

Output: Một phân đoạn của V thành các thành phần $S = (C_1, C_2, \dots)$.

Thuật toán:

- Bước 0: Sắp xếp các cạnh của G theo thứ tự không giảm của trọng số.

$$\pi = (o_1, o_2, \dots, o_m)$$

- Bước 1: Bắt đầu với phân đoạn S^0 , lúc này mỗi đỉnh nằm trong một thành phần.

- Bước 2: Lặp lại bước 3 với $q = 1, \dots, m$

- Bước 3: Xây dựng S^q từ S^{q-1} như sau: Cho v_i và v_j là hai đỉnh nối với nhau bởi cạnh thứ q , tức là $o_q = (v_i, v_j)$. Nếu v_i và v_j nằm trong hai thành phần tách rời nhau của S^{q-1} và $w(o_q)$ nhỏ hơn sự khác-nhau-nội-vùng của cả hai thành phần thì trộn hai thành phần này với nhau, ngược lại không làm gì cả. Cụ thể hơn, gọi C_i^{q-1} là thành phần của S^{q-1} chứa v_i và C_j^{q-1} là thành phần của S^{q-1} chứa v_j . Nếu $C_i^{q-1} \neq C_j^{q-1}$ và $w(o_q) \leq MInt(C_i^{q-1}, C_j^{q-1})$ thì S^q thu được từ S^{q-1} bằng cách trộn C_i^{q-1} với C_j^{q-1} . Ngược lại $S^q = S^{q-1}$.

- Bước 4: Trả về kết quả $S = S^m$.

Chúng ta sẽ chứng minh rằng phân đoạn S được xây dựng trong thuật toán trên là tuân theo các thuộc tính toàn cục khi sử dụng tính chất so sánh cặp miền đã định nghĩa trong phần trước. Nghĩa là mặc dù thuật toán chỉ dựa vào các quyết định

tham lam nhưng phân đoạn được xây dựng vẫn thỏa mãn các thuộc tính toàn cục. Để chứng minh điều này chúng ta xem xét các bổ đề và các định lý sau đây:

3.4.5 Bổ đề 1

Giả sử C_i^{q-1} và C_j^{q-1} biểu diễn hai thành phần được nối với nhau bằng cạnh $o_q = (v_i, v_j)$ thì $C_i = C_i^{q-1}$ hoặc $C_j = C_j^{q-1}$ mà C_i là thành phần chứa v_i và C_j là thành phần chứa v_j trong phân đoạn S cuối cùng.

Chứng minh: Khi hai thành phần không được trộn với nhau thì có hai trường hợp có thể xảy ra.

Hoặc là $\omega(o_q) > \text{Int}(C_i^{q-1}) + \tau(C_i^{q-1})$, hoặc là $\omega(o_q) > \text{Int}(C_j^{q-1}) + \tau(C_j^{q-1})$. Vì các cạnh được sắp xếp theo chiều không giảm của trọng số, nên $\omega(o_k) \geq \omega(o_q)$ với mọi $k \geq q+1$. Do đó không có thêm phép trộn nào xảy ra nữa.

3.4.6 Định lý 1

Phân đoạn S sử dụng tính chất so sánh miền D và thuật toán 1 là không quá mịn theo định nghĩa 1.

Chứng minh: Theo định nghĩa, để S là quá quá mịn thì phải có một số cặp thành phần nào đó mà D không nắm bắt được. Thế thì phải tồn tại ít nhất một cạnh giữa hai thành phần cùng cặp vì theo bước 3 của thuật toán thì chúng không được trộn thành 1 miền. Chẳng hạn cho $o_q = (v_i, v_j)$ là cạnh có thứ tự đầu tiên. Trong trường hợp này thì theo thuật toán trên C_i^{q-1} và C_j^{q-1} không được trộn vào nhau, nghĩa là $w(o_q) > \text{MInt}(C_i^{q-1}, C_j^{q-1})$. Theo bổ đề 1 chúng ta biết rằng $C_i = C_i^{q-1}$ hoặc $C_j = C_j^{q-1}$. Một trong hai điều này xảy ra nghĩa là $w(o_q) > \text{MInt}(C_i, C_j)$, điều này chứng tỏ D nắm bắt được cả C_i và C_j . Đây là điều mâu thuẫn, vậy định lý được chứng minh.

3.4.7 Định lý 2

Phân đoạn S sử dụng tính chất so sánh miền D và thuật toán 1 là không quá thô theo định nghĩa 2.

Chứng minh: Để S là không quá thô thì phải có một số phép tinh chỉnh hợp lý T , sao cho nó vẫn chưa là quá mịn. Xem xét cạnh có trọng số nhỏ nhất e nằm trong thành phần $C \in S$ nhưng khác với $A, B \in T$. Theo định nghĩa về phép tinh chỉnh thì $A \subset C$ và $B \subset C$. Vì T là không quá mịn, nên hoặc là $w(e) > \text{Int}(A) + \tau(A)$ hoặc $w(e) > \text{Int}(B) + \tau(B)$. Không mất tính tổng quát, giả sử biểu thức đầu đúng, khi đó bằng cách xây dựng một kết nối từ A tới một thành phần con của C . Trọng số của cạnh này cũng lắm là bằng $w(e)$ vì $w(e) > \text{Int}(A)$. Mà theo thuật toán thì khi xem xét đến các cạnh trong tập đã sắp xếp theo chiều không giảm của trọng số, phải xem xét tất cả các cạnh trong cây khung $\text{MST}(A, E)$ trước khi xem xét các cạnh từ A đến một thành phần khác của C . Do đó thuật toán phải xây dựng A trước C , và trong bước xây dựng C thì phải trộn A với một thành phần con của C . Trọng số của cạnh nối giữa A và thành phần này lớn nhất là $w(e)$. Tuy nhiên thuật toán đã không trộn A vì $w(e) > \text{Int}(A) + \tau(A)$, đây là một mâu thuẫn. Vậy định lý đã được chứng minh.

3.4.8 Định lý 3

Phân đoạn theo thuật toán 1 không phụ thuộc vào việc sắp xếp các cạnh theo thứ tự không giảm của trọng số.

Chứng minh: Bất kì một thứ tự sắp xếp nào cũng được thay đổi chỉ bằng cách đảo vị trí của các phần tử liên kề. Điều này chỉ ra rằng bằng cách đổi chỗ thứ tự của hai cạnh liên kề cùng trọng số thì sự sắp xếp không giảm của các cạnh vẫn không thay đổi, và kết quả phân đoạn được sinh ra theo thuật toán 1 cũng không thay đổi.

Cho e_1 và e_2 là hai cạnh có cùng trọng số liên kề nhau trong dãy cạnh sau khi sắp xếp. Rõ ràng là khi thuật toán xem xét cạnh đầu tiên trong hai cạnh thì chúng kết nối giữa hai thành phần tách rời hay nói chính xác hơn là hai cặp của các thành phần, nên thứ tự của hai cạnh này là không thành vấn đề. Chỉ còn trường hợp chúng ta cần thiết kiểm tra đó là kiểm tra là khi e_1 kết nối giữa thành phần A và thành phần B và e_2 là kết nối giữa một trong hai thành phần A hoặc B với một thành phần C .

Bây giờ chúng ta chỉ ra rằng e_1 là căn nguyên của phép trộn khi xem xét sau e_2 . Điều này ngụ ý rằng $w(e_1) \leq MInt(A, B)$. Nếu e_2 thay vì được xem xét trước e_1 , thì cả e_2 và e_1 vẫn là căn nguyên của phép trộn, hoặc là e_2 cũng là căn nguyên của phép trộn trong trường hợp thành phần mới của $B \cup C$ có $Int(B \cup C) = w(e_2) = w(e_1)$. Do đó chúng ta biết rằng $w(e_1) \leq MInt(A, B \cup C)$ vẫn ngụ ý rằng e_1 vẫn là căn nguyên của phép trộn. Mặt khác, giả sử rằng e_1 không là căn nguyên của phép trộn nếu nó được xem xét sau e_2 . Tức là $w(e_1) > MInt(A, B)$. Do đó hoặc là (1) $w(e_1) > Int(A) + \tau(A)$ hoặc là (2) $w(e_2) > Int(B) + \tau(B)$. Trong trường hợp (1) vẫn đúng nếu e_2 được xem xét trước. Trong trường hợp (2) nếu e_2 được xem xét trước thì nó có thể không phải là căn nguyên của một phép trộn vì $w(e_1) = w(e_2)$ và do đó $w(e_2) > MInt(B, C)$. Vậy khi xem xét e_1 sau e_2 chúng ta vẫn có $w(e_1) > MInt(A, B)$ và e_1 không là căn nguyên của phép trộn.

3.4.9 Độ phức tạp tính toán

Thời gian thực hiện của thuật toán này được chia làm hai phần:

Một là thời gian cần thiết để sắp xếp dãy trọng số theo chiều không giảm (bước 0). Đối với dãy số nguyên thì điều này có thể thực hiện trong thời gian tuyến tính. Có rất nhiều phương pháp sắp xếp có thể thực hiện trong thời gian $O(m \log m)$ với m là số lượng cạnh.

Hai là thời gian thực hiện bước 1-3. Để kiểm tra được hai đỉnh có cùng chung trong một thành phần hay không chúng tôi sử dụng biến set-find trên mỗi đỉnh nhằm lưu lại số hiệu thành phần mà đỉnh đó đang phụ thuộc vào. Để trộn hai thành phần lại với nhau chúng tôi chỉ việc hiệu chỉnh lại các biến set-find của một trong hai tập đỉnh. $MInt$ được tính trong một hằng thời gian nếu biết được Int và kích thước của mỗi thành phần. Int cũng được tính trong một hằng thời gian cho mỗi phép trộn, vì cạnh có trọng số lớn nhất trong cây khung nhỏ nhất của một thành phần là căn nguyên của phép trộn. Có được điều này vì bổ đề 1 nói rằng căn nguyên của phép trộn chính là cạnh có trọng số nhỏ nhất giữa hai thành phần được trộn.

Kích thước của thành phần sau khi trộn bằng tổng kích thước của hai thành phần trước khi trộn. Vậy độ phức tạp tính toán từ bước 1 đến bước 3 của thuật toán là $O(m\alpha(m))$ trong đó α là hàm Ackerman nghịch đảo, m là số cạnh của đồ thị.

CHƯƠNG 4: CÀI ĐẶT THỬ NGHIỆM

4.1 Định dạng ảnh PPM (PPM format)

PPM là một định dạng ảnh khá cũ, nó dùng để miêu tả một vài ảnh màu đơn giản. PPM file giống như một text file với việc không nén bất kỳ một dữ liệu ảnh nào. Điều đó tạo điều kiện thuận lợi cho việc đọc và xử lý file.

Với một PPM file ảnh trong thực tế



Ta có thể miêu tả bức ảnh đó như sau:

```
P3
4 3
255
0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 255 255 255 255 0
0 150 150 255 0 0 0 255 0 0 150 150 255 150 150 255
```

Ở đây P3 là một PPM file , dòng tiếp theo 4 3 là cho chúng ta biết về chiều rộng và chiều cao của ảnh. Tiếp theo là dùng để chỉ ra là bức ảnh 256 màu (0....255) .Tất cả các số ở dòng tiếp theo là dữ liệu về điểm ảnh, bắt đầu từ giá trị đầu tiên ở bên trái dọc theo từng hàng. Ví dụ ,0 0 0 mô tả điểm ảnh đầu tiên về bên trái có giá trị RGB là(0,0,0) thì đó là màu đen(black), với giá trị màu cuối cùng 150 150 255 dùng để mô tả điểm ảnh cuối cùng ở góc phải tương ứng với giá trị màu RGB(150,150,255) đó là màu xanh da trời(blue).Chúng ta có thể thấy rằng dữ liệu ảnh trong PPM file là không được nén , bởi vậy nó chiếm dung lượng khá lớn. Ví dụ với một bức ảnh pháo hoa theo định dạng JPG (một định dạng ảnh khác) nó chỉ chiếm 8K ,nhưng biểu

diễn với định dạng ảnh PPM thì nó phải chiếm 176K. Thuận lợi của định dạng ảnh PPM là khá dễ dàng trong khi làm việc với các điểm ảnh (có thể đọc và dễ dàng rút trích từ PPM file) nhưng nó lại chiếm một khoảng không gian lưu trữ khá lớn. Do vậy khi viết chương trình cần sử dụng PPM file ta nên dùng ảnh nhỏ để thuận lợi cho việc test chương trình.

4.1 Cài đặt thử nghiệm

Các thuật toán quen thuộc như sử dụng ngưỡng cố định, phát hiện biên, thuật toán đẳng liêu,... đã được cài đặt khá nhiều, do vậy trong khuôn khổ đề án em tiến hành cài đặt theo đúng thuật toán phân đoạn ảnh bằng đồ thị bằng C++ trong môi trường Microsoft. Thuật toán chạy nhanh và phân đoạn của bức ảnh tương đối chính xác.

Dưới đây là một số đoạn mã chính của chương trình:

Thủ tục phân đoạn một bức ảnh theo thuật toán 1:

Input:

- *im: Con trỏ trỏ đến bức ảnh màu đầu vào.
- sigma: tham số để làm trơn điểm ảnh.
- c:
- min_size:

Output:

- Bức ảnh sau khi phân đoạn. Mỗi vùng được gán một màu ngẫu nhiên.

Nội dung:

```
image<rgb> *segment_image(image<rgb> *im, float sigma, float c, int min_size,  
                           int *num_ccs) {  
    int width = im->width();
```



```
int height = im->height();

image<float> *r = new image<float>(width, height);
image<float> *g = new image<float>(width, height);
image<float> *b = new image<float>(width, height);

// lam tron theo tung thanh phan mau (red, green, blue)

int y, x;
for (y = 0; y < height; y++)
{
    for (x = 0; x < width; x++)
    {
        imRef(r, x, y) = imRef(im, x, y).r;
        imRef(g, x, y) = imRef(im, x, y).g;
        imRef(b, x, y) = imRef(im, x, y).b;
    }
}

image<float> *smooth_r = smooth(r, sigma);
image<float> *smooth_g = smooth(g, sigma);
image<float> *smooth_b = smooth(b, sigma);

// xay dung do thi tu buc anh input sau khi da lam tron

edge *edges = new edge[width*height*4];

int num = 0;

for (y = 0; y < height; y++)
{
```

```
        for (x = 0; x < width; x++)  
        {  
            if (x < width-1)  
            {  
                edges[num].a = y * width + x;  
                edges[num].b = y * width + (x+1);  
                edges[num].w = diff(smooth_r, smooth_g, smooth_b, x, y, x+1, y);  
                num++;  
            }  
            if (y < height-1)  
            {  
                edges[num].a = y * width + x;  
                edges[num].b = (y+1) * width + x;  
                edges[num].w = diff(smooth_r, smooth_g, smooth_b, x, y, x, y+1);  
                num++;  
            }  
            if ((x < width-1) && (y < height-1))  
            {  
                edges[num].a = y * width + x;  
                edges[num].b = (y+1) * width + (x+1);  
                edges[num].w = diff(smooth_r, smooth_g, smooth_b, x, y, x+1, y+1);  
                num++;  
            }  
        }
```

```
    if ((x < width-1) && (y > 0)) {  
        edges[num].a = y * width + x;  
        edges[num].b = (y-1) * width + (x+1);  
        edges[num].w = diff(smooth_r, smooth_g, smooth_b, x, y, x+1, y-1);  
        num++;  
    }  
}  
}  
}  
  
delete smooth_r;  
delete smooth_g;  
delete smooth_b;  
  
//goi thu tuc de phan doan anh, thuc chat la phan doan graph.  
universe *u = segment_graph(width*height, num, edges, c);  
for (int i = 0; i < num; i++) {  
    int a = u->find(edges[i].a);  
    int b = u->find(edges[i].b);  
    if ((a != b) && ((u->size(a) < min_size) || (u->size(b) < min_size)))  
        u->join(a, b);  
}  
  
delete [] edges;  
  
*num_ccs = u->num_sets();
```

```
image<rgb> *output = new image<rgb>(width, height);

// lay ngau nhien cac mau cho moi thanh phan sau phan doan
rgb *colors = new rgb[width*height];
for (i = 0; i < width*height; i++)
    colors[i] = random_rgb();

for (y = 0; y < height-1; y++) {
    for (x = 0; x < width-1; x++) {
        int comp = u->find(y * width + x);
        imRef(output, x, y) = colors[comp];
    }
}

return output; // tra ve buc anh sau khi phan doan
}
```

Thủ tục phân đoạn một đồ thị segment_graph

Input:

- num_vertices : số lượng đỉnh.
- numb_edges: số lượng cạnh
- *edges: danh sách cạnh. mỗi cạnh gồm 3 tham số. a - đỉnh đầu, b-đỉnh cuối, w-trọng số của cạnh.
- c : số lượng thành phần.

```
universe *segment_graph(int num_vertices, int num_edges, edge *edges,
```

```
float c) {  
  
    // sap xep cac canh theo chieu khong giam  
    std::sort(edges, edges + num_edges);  
  
    // tao cac thanh phan lien thong, moi thanh phan chi co 1 dinh  
    universe *u = new universe(num_vertices);  
  
    // khoi tao nguoi  
    float *threshold = new float[num_vertices];  
    int i = 0;  
    for (i = 0; i < num_vertices; i++)  
        threshold[i] = THRESHOLD(1,c);  
  
    // duyet tuan tu cac canh trong danh sach canh da sap xep  
    for (i = 0; i < num_edges; i++)  
    {  
        edge *pedge = &edges[i];  
  
        //  
  
        int a = u->find(pedge->a);  
        int b = u->find(pedge->b);  
  
        if (a != b) {  
            if ((pedge->w <= threshold[a]) &&  
                (pedge->w <= threshold[b])) {
```

```
    u->join(a, b);

    a = u->find(a);

    threshold[a] = pedge->w + THRESHOLD(u->size(a), c);

    }

}

}

delete threshold;

return u;

}

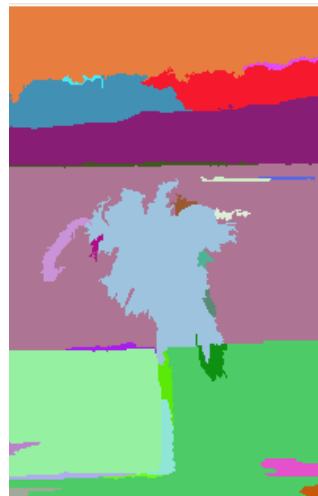
//ham find(int x) de tim mot phan tu trong tap hop
int universe::find(int x) {
    int y = x;
    while (y != elts[y].p)
        y = elts[y].p;
    elts[x].p = y;
    return y;
}

// ham joint de ket hop 2 mien lien thong thanh 1
void universe::join(int x, int y) {
    if (elts[x].rank > elts[y].rank) {
```

```
    elts[y].p = x;
    elts[x].size += elts[y].size;
} else {
    elts[x].p = y;
    elts[y].size += elts[x].size;
    if (elts[x].rank == elts[y].rank)
        elts[y].rank++;
}
num--;
}
```

4.3 Một số kết quả minh họa

Phân đoạn ảnh cây dừa với các tham số $K=0.5$, $\text{sing}= 500$, $\text{min_size}=50$ thì kết quả ảnh được chia thành 29 phần nhỏ và đổ màu một cách ngẫu nhiên.



Với ảnh cô gái ta phân đoạn với tham số như sau $K=0.5$, $\text{sing} = 1000$, $\text{min_size}=10$ thì chia bức ảnh thành 206 thành phần khác nhau .



KẾT LUẬN

5.1 Nội dung của đồ án

5.1.1 Các kết quả đạt được

Trong quá trình nghiên cứu tài liệu và thực hiện đồ án dưới sự định hướng của thầy hướng dẫn em thấy bản thân đã đạt được một số kết quả như sau:

- Tìm hiểu được một cách tổng quan các vấn đề về XLA và phân đoạn ảnh. Em đã có một cách nhìn có hệ thống về các phương pháp phân đoạn ảnh và các thuật toán trong mỗi phương pháp. Đồng thời biết được điểm mạnh/yếu của từng phương pháp và có thể đưa ra cách lựa chọn phương pháp phù hợp với từng loại ảnh. .
- Trong chương 3 em đã tìm hiểu và cài đặt được một phương pháp cải tiến phương pháp phân đoạn dựa vào đồ thị. Phương pháp này phân đoạn nhanh và hiệu quả. Nó sử dụng được các thuộc tính local và non-local của bức ảnh để tăng cường khả năng phân đoạn chính xác.
- Ngoài ra, trong quá trình nghiên cứu em cũng tự tích lũy thêm cho mình các kiến thức về toán học, về kỹ thuật lập trình,... Và quan trọng là rèn luyện kỹ năng để thực hiện một nghiên cứu khoa học.

5.1.2 Một số hạn chế cần khắc phục

Bên cạnh những kết quả đạt được em tự thấy bản luận văn vẫn còn một số hạn chế.

- Chưa đưa ra được một phương pháp phân đoạn mới hoàn toàn. Trong khuôn khổ một đồ án tốt nghiệp ,em mới chỉ trình bày lại các kiến thức tìm hiểu được chứ chưa đề xuất được một phương pháp hoàn toàn mới.

- Do thời gian có hạn, nên việc trình bày các thuật toán phân đoạn cũng chưa được hệ thống và khoa học. Có nhiều thuật toán được trình bày sơ lược.
- Đồ án cũng chưa chỉ ra được các ứng dụng thực tế của các thuật toán phân đoạn.

5.2 Công việc tiếp theo

Dựa trên những kết quả bước đầu đã đạt được trong đồ án, em có đề xuất một số cải tiến thuật toán phân đoạn để phân đoạn hiệu quả hơn trong tương lai.

- Xây dựng một ứng dụng xử lý ảnh hoàn chỉnh dựa theo các thuật toán đã trình bày trong luận văn. Ứng dụng này nhằm phân đoạn ảnh để nhận diện được các thành phần có trong ảnh. Trích rút ra các đối tượng có trong ảnh và đặt tên cho chúng.
- Các thuật toán phân đoạn trình bày trong luận văn áp dụng đối với ảnh tĩnh, trong thời gian tới, em hy vọng có thể tìm hiểu và phát triển các thuật toán phân đoạn đối với ảnh động hoặc các đoạn video ngắn.

TÀI LIỆU THAM KHẢO

Tiếng việt

1. .Lương Mạnh Bá,Nguyễn Thanh Thủy ,*Nhập môn xử lý ảnh số*,NXB Khoa học và kỹ thuật, Hà Nội.
2. Võ Đức Khánh(2003), *Giáo trình xử lý ảnh* ,NXB Thống kê,Hà Nội
3. NguyễnKim Sách(1997), *Xử lý ảnh và video số*, NXB Khoa học kỹ thuật Hà Nội

Tiếng Anh

4. Baris S., Manjunath B. S., and Charles Ke. (2001), “Image Segmentation using Curve Evolution”, *Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp 1141-1145.
5. Cooper M. C. (1998), “The tractability of segmentation and scene analysis”, *International journal of Computer Vision*”, vol 30, pp 27-42.
6. Felzenszwalb P. Huttenlocker D. (2004) “Efficient Graph-Based Image Segmentation”, *International Journal of Computer Vision*, Volume 59, Number 2.