

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----

**TÌM HIỂU PHƯƠNG PHÁP CỰC TIỂU NĂNG
LƯỢNG DỰA TRÊN ĐỘ ĐỒNG NHẤT VÀ ĐỘ
KHÔNG ỔN ĐỊNH CHO PHÂN ĐOẠN ẢNH**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

Sinh viên thực hiện: Lê Thị Ngọc Mai

Giáo viên hướng dẫn: PGS TS. Ngô Quốc Tạo

Mã số sinh viên: 110315

Mục lục

LỜI CẢM ƠN	4
DANH MỤC TỪ VIẾT TẮT.....	5
DANH SÁCH ẢNH.....	6
LỜI MỞ ĐẦU	7
CHƯƠNG 1: TỔNG QUAN VỀ XỬ LÝ ẢNH VÀ PHÂN ĐOẠN ẢNH	8
1.1 Tổng quan về xử lý ảnh.....	8
1.1.1 XLA là gì?	8
1.1.2 Sơ đồ tổng quát XLA	9
1.1.3 Mô tả (biểu diễn ảnh)	11
1.1.4 Các khái niệm cơ bản	13
1.2 Tổng quan về phân đoạn ảnh.....	15
CHƯƠNG 2: CÁC HƯỚNG TIẾP CẬN CHÍNH TRONG PHÂN ĐOẠN ẢNH...17	
2.1 Các phương pháp dựa trên không gian đặc trưng	17
2.2 Các phương pháp dựa trên không gian ảnh.....	17
2.3 Các phương pháp dựa trên mô hình vật lý	18
2.4 Một số thuật toán phân đoạn ảnh.....	23
2.4.1 Thuật toán Entropy cực đại	23
2.4.2 Thuật toán độ lệch nhỏ nhất	29
CHƯƠNG 3: PHƯƠNG PHÁP CỰC TIỂU NĂNG LƯỢNG DỰA TRÊN ĐỘ ĐỒNG NHẤT VÀ ĐỘ KHÔNG ỔN ĐỊNH CHO PHÂN ĐOẠN ẢNH.....36	
3.1 Giới thiệu.....	36
3.1.1 Cơ sở lý thuyết	36
3.1.2 Tối ưu và tự động ngưỡng.....	36
3.2 Lý thuyết.....	37
3.2.1 Cường độ dựa trên độ không ổn định.....	37
3.2.2 Bề mặt năng lượng và tối ưu ngưỡng.....	38
3.3 Phương pháp.....	38

3.3.1 Phân bố xác suất tiên nghiệm đối tượng $p_o(g)$ và nền $p_B(g)$	39
3.3.2 Hàm mật độ θ	40
3.3.3 Bản đồ gradient chuẩn ∇_{σ}	40
3.3.4 Tối ưu giá trị của t và σ trên bề mặt năng lượng E	40
3.4 Tiến trình giải thuật:	42
CHƯƠNG 4: CÀI ĐẶT CHƯƠNG TRÌNH VÀ ĐÁNH GIÁ	44
4.1 Cài đặt chương trình	44
4.1.1 Định dạng ảnh BMP	44
4.1.2 Cài đặt thử nghiệm	45
4.2 Một số kết quả và đánh giá	54
KẾT LUẬN	57
TÀI LIỆU THAM KHẢO	58

LỜI CẢM ƠN

Trước hết em xin gửi lời cảm ơn đến PGS TS. Ngô Quốc Tạo – Viện CNTT, Viện KH&CN Việt Nam, người thầy đã hướng dẫn em rất nhiều trong suốt quá trình tìm hiểu nghiên cứu và hoàn thành đề án tốt nghiệp từ lý thuyết đến ứng dụng. Sự hướng dẫn của thầy đã giúp em có thêm được những hiểu biết về phân đoạn ảnh, đặc biệt phương pháp cực tiểu năng lượng dựa trên độ đồng nhất và độ không ổn định cho phân đoạn ảnh.

Đồng thời em cũng xin chân thành cảm ơn các thầy cô trong bộ môn cũng như các thầy cô trong trường đã trang bị cho em những kiến thức cơ bản cần thiết để em có thể hoàn thành tốt đề án này.

Em xin gửi lời cảm ơn đến gia đình, bạn bè đã tạo mọi điều kiện thuận lợi để em có thể xây dựng thành công đề án.

Dù đã rất cố gắng để hoàn thành công việc được giao, song do trình độ còn hạn chế nên không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý và thông cảm của mọi người.

Em xin trân thành cảm ơn!

Hải Phòng, Ngày tháng 7 năm 2011

Sinh viên thực hiện

Lê Thị Ngọc Mai

DANH MỤC TỪ VIẾT TẮT

Kí hiệu viết tắt	Giải thích
XLA	Xử lý ảnh
R	Red
G	Green
B	Blue

DANH SÁCH ẢNH

Hình 1.1	Quá trình XLA	8
Hình 1.2.	Các bước cơ bản trong XLA	9
Hình 1.3.	Biểu diễn ảnh bằng mã chạy	11
Hình 1.4.	Biểu diễn ảnh bằng mã xích (8 hướng).....	12
Hình 1.5.	Biểu diễn ảnh bằng mã tứ phân.....	13
Hình 1.6	Các láng giềng của điểm ảnh (x, y)	15
Hình 2.1.	Phân đoạn theo thuật toán Entropy cực đại	28
Hình 2.2.	Phân đoạn theo thuật toán độ lệch nhỏ nhất.	34
Hình 3.2.	Ví dụ minh họa dòng năng lượng và bề mặt năng lượng.....	41
Hình 3.1.	Minh họa vực bên trong.....	42
Hình 4.1.	Ảnh lưu dưới dạng BMP đuôi .bmp	44
Hình 4.2.	Kết quả phân đoạn ảnh bông hoa và biểu đồ dòng năng lượng	54
Hình 4.3.	Kết quả phân đoạn ảnh cô gái và biểu đồ dòng năng lượng	55

LỜI MỞ ĐẦU

Trong vài thập kỷ qua, khai thác đa tầng của thông tin trong ảnh hai hay nhiều chiều vẫn là chủ đề của rất nhiều bài nghiên cứu. Đặc biệt sự thông dụng của kỹ thuật ảnh trong nhiều ngành như y học, vật lý, hóa học... đã làm đẩy mạnh quá trình xử lý ảnh bằng máy tính để khai thác dữ liệu ảnh lớn nhằm đưa ra sản phẩm mong muốn. Phân đoạn là một nhiệm vụ nổi bật nhất trong ứng dụng ảnh cụ thể như những gì liên quan tới phân loại đối tượng, hình dạng, phân tích chuyển động... Vì nhiều lý do mà xác định các đối tượng một cách chính xác và hiệu quả rất quan trọng trong xử lý ảnh trên máy tính và công việc này được gọi là phân đoạn ảnh.

Trong thời gian đầu, các phương pháp phân vùng ảnh được đưa ra chủ yếu làm việc trên các ảnh mức xám do các hạn chế về phương tiện thu thập và lưu trữ. Ngày nay, cùng với sự phát triển về các phương tiện thu nhận và biểu diễn ảnh, các ảnh màu đã hầu như thay thế hoàn toàn các ảnh mức xám trong việc biểu diễn và lưu trữ thông tin do các ưu thế vượt trội hơn hẳn so với ảnh mức xám. Do đó, các kỹ thuật, thuật giải mới thực hiện việc phân vùng ảnh trên các loại ảnh màu liên tục được phát triển để đáp ứng các nhu cầu mới. Các thuật giải, kỹ thuật này thường được phát triển dựa trên nền tảng các thuật giải phân vùng ảnh mức xám đã có sẵn.

Mục đích chính của em là tìm hiểu phương pháp cực tiểu năng lượng dựa trên độ đồng nhất và độ không ổn định cho phân đoạn ảnh. Và được trình bày trong 4 chương:

Chương 1: Trình bày tổng quan về xử lý ảnh và phân đoạn ảnh bao gồm 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, vai trò, nhiệm vụ của phân đoạn ảnh.

Chương 2: Giới thiệu các hướng tiếp cận chính trong phân đoạn ảnh, bao gồm: các phương pháp dựa trên không gian đặc trưng, các phương pháp dựa trên không gian ảnh, các phương pháp dựa trên mô hình vật lý. Trong chương này, em cũng xin trình bày hai thuật toán phân đoạn ảnh, đó là thuật toán Entropy cực đại và thuật toán độ chia nhỏ nhất.

Chương 3: Trình bày phương pháp cực tiểu năng lượng dựa trên độ đồng nhất và độ không ổn định cho phân đoạn ảnh bao gồm: giới thiệu tổng quan, cơ sở lý thuyết của phương pháp, tiến trình giải thuật.

Chương 4: Cài đặt chương trình, đưa ra một số kết quả và đánh giá.

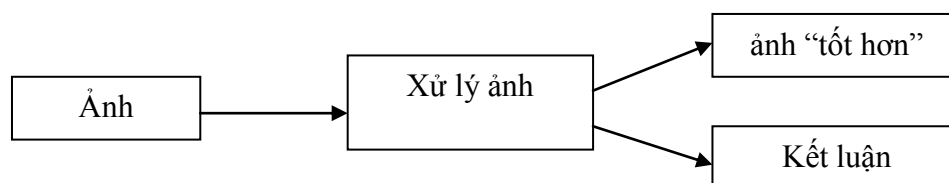
CHƯƠNG 1: TỔNG QUAN VỀ XỬ LÝ ẢNH VÀ PHÂN ĐOẠN ẢNH

Xử lý ảnh là lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ của nó phát triển rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng cho nó. 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... 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.

1.1 Tổng quan về xử lý ảnh (XLA)

1.1.1 XLA là gì?

Qua trình XLA là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra có thể là một ảnh “tốt hơn” hoặc một kết luận [1].



Hình 1.1 Quá trình XLA

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 thể xem như một hàm n bất biến $P(c_1, c_2, c_3, \dots, c_n)$. Do đó, ảnh trong XLA có thể xem như ảnh n chiều.

Mục đích của XLA là:

- ✓ *Biến đổi ảnh làm tăng chất lượng ảnh:*

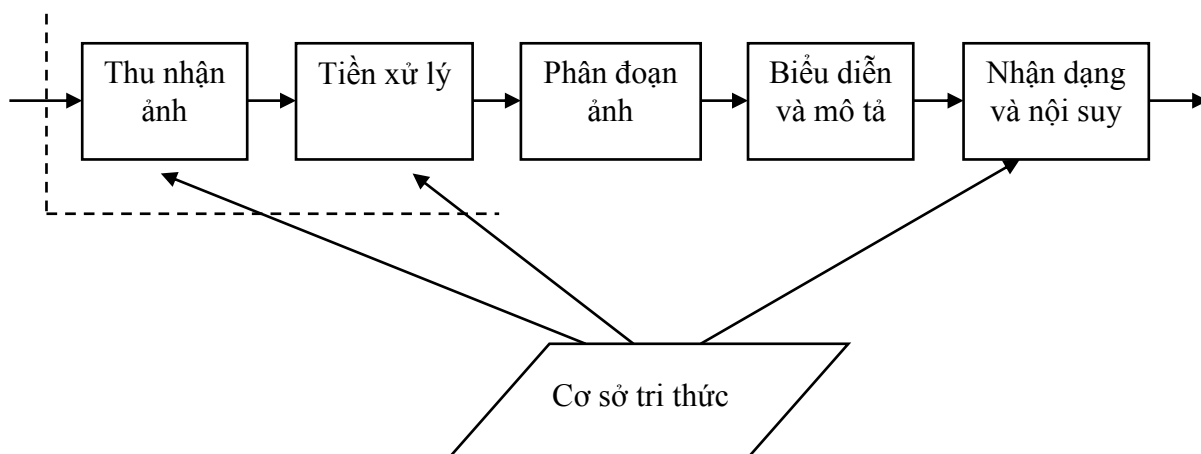
Phương pháp biến đổi ảnh được sử dụng trong việc xử lý các ảnh chụp từ không trung (chương trình đo đạc từ máy bay, vệ tinh và các ảnh vũ trụ) hoặc xử lý các ảnh trong y học (ảnh chụp cắt lát, ảnh siêu âm, vv...). Một ứng dụng khác

của việc biến đổi ảnh là mã hoá ảnh, trong đó các ảnh được xử lý để rồi lưu trữ hoặc truyền đi.

✓ *Tự động nhận dạng, đoán ảnh, đánh giá nội dung ảnh:*

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 Sơ đồ tổng quát XLA



Hình 1.2. Các bước cơ bản trong XLA

✓ *Thu nhận ảnh (Image Acquisition):*

Ảnh có thể nhận qua camera màu hoặc đen trắng, scanner hay giác quan... Thường ảnh nhận qua camera và scanner là ảnh tương tự hoặc ảnh số (với các camera đã số hóa).

Camera thường dùng là loại quét dòng; ảnh tạo ra có dạng hai chiều. Chất lượng của ảnh thu được phụ thuộc vào thiết bị thu và môi trường (ánh sáng, phong cảnh).

✓ *Tiền xử lý (Image Processing):*

Sau khi thu nhận, ảnh có thể nhiễu, độ tương phản thấp nên cần đưa vào bộ tiền xử lý để nâng cao chất lượng. Chức năng chính của bộ tiền xử lý là lọc nhiễu, nâng độ tương phản làm cho ảnh rõ hơn, nét hơn.

✓ *Phân đoạn ảnh (Image Segmentation):*

Phân đoạn ảnh là tách ảnh ban đầu thành các vùng thành phần để biểu diễn phân tích hoặc nhận dạng ảnh. Đây là phần phức tạp, khó khăn nhất trong XLA, cũng dễ gây lỗi, làm mất độ chính xác của ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này.

✓ *Biểu diễn ảnh (Image Representation):*

Ảnh đầu ra sau phân đoạn chứa các điểm ảnh của vùng ảnh (ảnh đã phân đoạn) cộng với mã liên kết với các vùng lân cận. Việc biến đổi các số liệu này thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng (*Feature Selection*) gắn với việc tách các đặc tính của ảnh dưới dạng các thông tin định lượng hoặc làm cơ sở để phân lớp đối tượng này với đối tượng khác trong phạm vi ảnh nhận được. Ví dụ: trong nhận dạng các ký tự, ta miêu tả các đặc trưng của từng ký tự, giúp phân biệt ký tự này với ký tự khác.

✓ *Nhận dạng và nội suy ảnh (Image Recognition and Interpretation):*

Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu) từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Có nhiều cách phân loại khác nhau về ảnh. Theo lý thuyết về nhận dạng, các mô hình toán học về ảnh được phân theo hai loại nhận dạng cơ bản:

- Nhận dạng theo tham số.
- Nhận dạng theo cấu trúc.

Một số đối tượng nhận dạng khá phổ biến hiện nay đang được áp dụng trong khoa học và công nghệ là nhận dạng ký tự, nhận dạng văn bản, nhận dạng vân tay, nhận dạng mã vạch, nhận dạng khuôn mặt...

✓ *Cơ sở tri thức (Knowledge Base):*

Ảnh là một đối tượng phức tạp về đường nét, độ sáng tối, dung lượng điểm ảnh, môi trường để thu nhận ảnh phong phú, kéo theo nhiễu. Trong nhiều khâu xử lý và phân tích ảnh, ngoài việc đơn giản hóa các phương pháp toán học đảm bảo tiện lợi cho xử lý, người ta bắt buộc quy trình tiếp nhận và XLA theo cách của con người. Trong các bước xử lý đó, nhiều khâu hiện nay đã xử lý theo các phương pháp trí tuệ con người. Vì vậy, ở đây cơ sở tri thức được phát huy

1.1.3 Mô tả (biểu diễn ảnh)

Từ hình 1.1, ảnh sau khi số hóa sẽ được lưu vào bộ nhớ hoặc chuyển sang khâu tiếp theo để phân tích. Nếu lưu trữ ảnh trực tiếp từ ảnh thô, đòi hỏi dung lượng bộ nhớ phải cực lớn và không hiệu quả theo quan điểm ứng dụng và công nghệ. Thông thường, các ảnh thô đó được biểu diễn lại (hay đơn giản là mã hóa) theo các đặc điểm của ảnh gọi là đặc trưng ảnh (*Image Features*) như biên ảnh (*boundary*), vùng ảnh (*Region*). Một số phương pháp biểu diễn thường dùng:

- Biểu diễn bằng mã chạy (Run-Length Code).
- Biểu diễn bằng mã xích (Chain Code).
- Biểu diễn bằng mã tứ phân (Quad-Tree Code).
- ✓ *Biểu diễn bằng mã chạy:*

Phương pháp này thường biểu diễn cho vùng ảnh và áp dụng cho ảnh nhị phân. Một vùng ảnh R có thể mã hóa đơn giản nhờ một ma trận nhị phân:

$$U(m, n) = 1 \text{ nếu } (m, n) \in R$$

$$U(m, n) = 0 \text{ nếu } (m, n) \notin R$$

Trong đó: $U(m, n)$ là hàm mô tả mức xám ảnh tại tọa độ (m, n) . Với cách biểu diễn trên, một vùng ảnh được mô tả bằng một tập các chuỗi 0 hoặc 1. Giả sử ta mô tả ảnh nhị phân của một vùng ảnh được thể hiện theo tọa độ (x, y) theo các chiều và đặc tả chỉ đối với giá trị "1", khi đó dạng mô tả có thể là: $(x, y)r$; trong đó, (x, y) là tọa độ, r là số lượng các bit có giá trị "1" liên tục theo chiều dọc hoặc chiều ngang (xem hình 1.3).

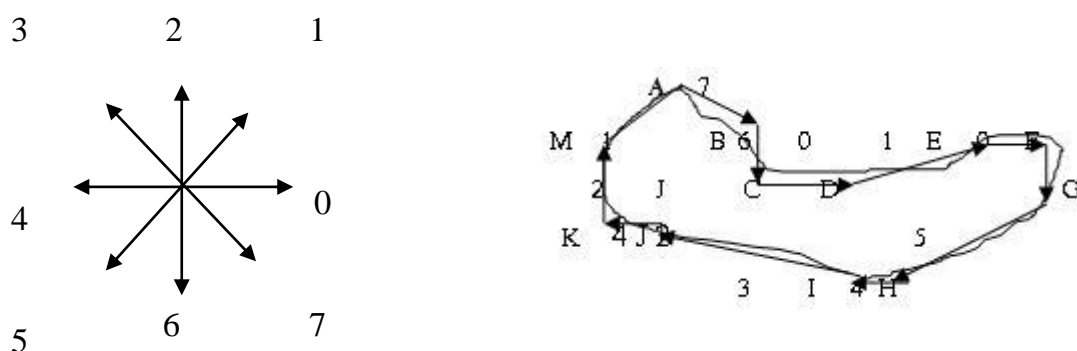
	0	1	2	3	4	5
0						
1		■	■			
2			■	■		
3				■	■	
4					■	
5						

Hình 1.3. Biểu diễn ảnh bằng mã chạy

(1, 1)2, (2, 2)2, (3, 3)2, (4, 4)1

✓ *Biểu diễn bằng mã xích:*

Phương pháp thường dùng để biểu diễn đường biên ảnh. Một đường bất kỳ được chia thành các đoạn nhỏ. Nối các điểm chia, ta có được đoạn thẳng kế tiếp được gán hướng cho đoạn thẳng đó, tạo thành một dây xích gồm các đoạn. Các hướng có thể chọn là 4, 8, 12, 24,... mỗi hướng được mã hóa theo số thập phân hoặc số nhị phân thành mã của hướng (xem hình 1.4).

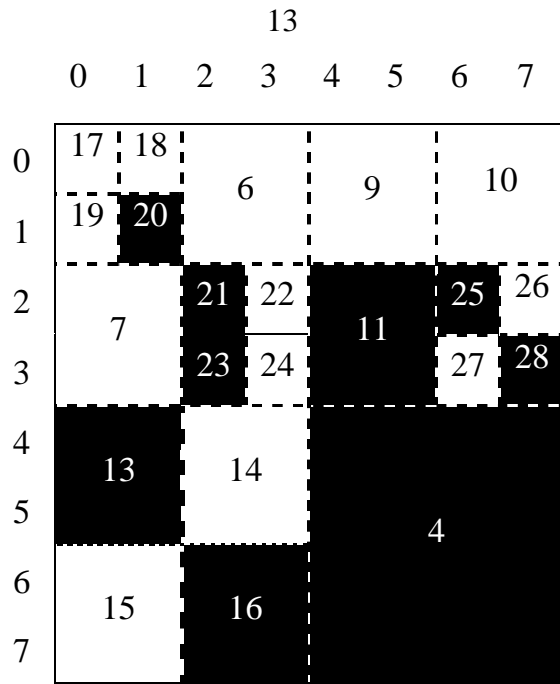


Hình 1.4. Biểu diễn ảnh bằng mã xích (8 hướng)

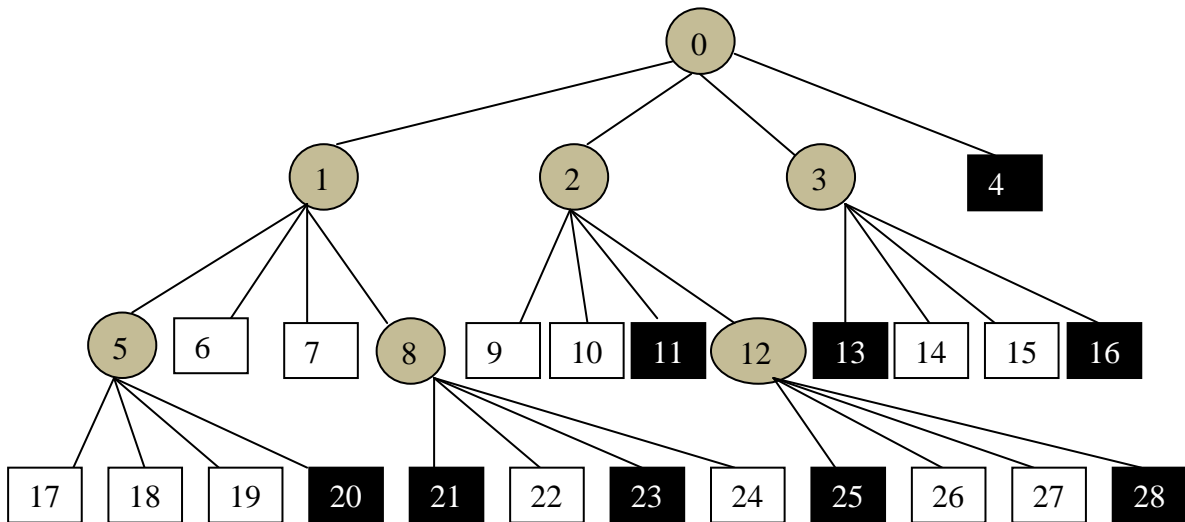
A 111 110 000 001 000 110 101 011 100 010

✓ *Biểu diễn bằng mã tứ phân:*

Phương pháp mã tứ phân được dùng để mã hóa cho vùng ảnh. Vùng ảnh đầu tiên được chia làm bốn phần thường là bằng nhau bằng nhau. Nếu mỗi vùng là đồng nhất, tức là chứa toàn điểm đen (1) hay toàn điểm trắng (0) (xem hình 1.5. a)), thì gán cho vùng đó một mã và không chia tiếp. Các vùng không đồng nhất được chia tiếp làm bốn phần theo thủ tục trên cho đến khi tất cả các vùng đều đồng nhất. Các mã phân chia thành các vùng con tạo thành một cây phân chia thành các vùng đồng nhất (xem hình 1.5. b)).



a)



b)

Hình 1.5. Biểu diễn ảnh bằng mã tứ phân

a) Chia ảnh thành các vùng đồng nhất

b) Cây phân chia

1.1.4 Các khái niệm cơ bản

✓ Ảnh số

Gốc của ảnh (ảnh tự nhiên) là ảnh liên tục về không gian và độ sáng. Để xử lý bằng máy tính, ảnh cần phải được số hóa. Số hóa ảnh là sự biến đổi gần đúng một

ảnh liên tục thành một tập hợp điểm phù hợp với ảnh thật về vị trí (không gian) và độ sáng (mức xám)

✓ *Điểm ảnh (pixel)*

Điểm ảnh (pixel) là một phần tử của ảnh số tại tọa độ (x, y) trong không gian ảnh 2 chiều với độ xám hoặc màu nhất định. Kích thước và khoảng cách giữa các điểm ảnh đó được chọn sao cho mắt người cảm nhận sự liên tục về không gian và mức xám (hoặc màu) của ảnh số gần như ảnh thật. Mỗi phần tử trong ma trận gọi là một phần tử ảnh.

✓ *Mức xám*

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

- Các thang giá trị mức xám thông thường: 16, 32, 65, 128, 256 (mức 256 là phổ biến nhất, ở mức này mỗi pixel được mã hóa bởi 8bit).

- Ảnh trắng đen: là ảnh chỉ có 2 màu trắng và đen (không chứa màu khác) với mức xám ở các điểm ảnh có thể khác nhau.

- Ảnh nhị phân: là ảnh có 2 mức trắng đen phân biệt, tức là dùng 1bit mô tả 2^1 mức khác nhau. Nói cách khác: mỗi bit điểm ảnh nhị phân chỉ có thể là 0 hoặc 1.

- Ảnh màu: trong khuôn khổ lý thuyết 3 màu (Red, Green, Blue) để tạo nên thế giới màu, người ta thường dùng 3 byte để mô tả mức màu, khi đó các giá trị màu: $2^{8 \times 3} = 2^{24} \approx 16,7$ triệu màu.

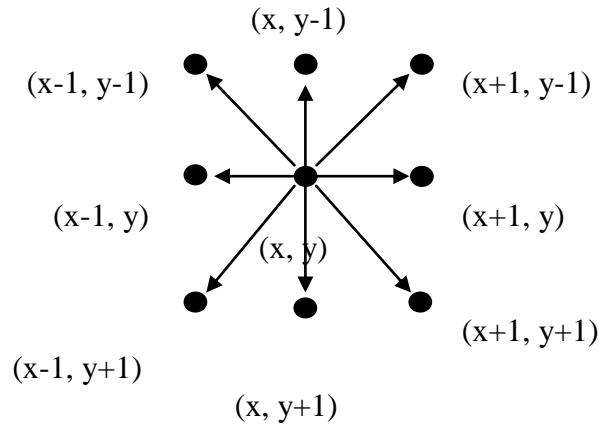
- Ảnh xám: là trường hợp đặc biệt của ảnh màu khi giá trị màu Red, Green, Blue bằng nhau.

✓ *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.

✓ *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*.



Hình 1.6 Các láng giềng của điểm ảnh (x, y) .

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:

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

✓ *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_i, y_i) nếu (x_i, y_i) thuộc vào láng giềng trực tiếp của (x_k, y_k) .

1.2 Tổng quan về phân đoạn ảnh

Phân đoạn ảnh là một thao tác ở mức thấp trong toàn bộ quá trình xử lý ảnh. Quá trình này thực hiện việc phân vùng ảnh thành các vùng rời rạc và đồng nhất với nhau hay nói cách khác là xác định các biên của các vùng ảnh đó. Các vùng ảnh đồng nhất này thông thường sẽ tương ứng với toàn bộ hay từng phần của các đối tượng thật sự bên trong ảnh. Vì thế, trong hầu hết các ứng dụng của lĩnh vực XLA, thị giác máy tính, phân đoạn ảnh luôn đóng một vai trò cơ bản và thường là bước tiền xử lý đầu tiên trong toàn bộ quá trình trước khi thực hiện các thao tác khác ở mức cao hơn như nhận dạng đối tượng, biểu diễn đối tượng, nén ảnh dựa trên đối tượng, hay truy vấn ảnh dựa vào nội dung...

Trước hết cần làm rõ khái niệm “vùng ảnh” (*Segment*) và đặc trưng vật lý của vùng. Vùng ảnh là một chi tiết, một thực thể trong toàn cảnh. Nó là tập hợp các điểm ảnh có cùng hoặc gần cùng một tính chất nào đó: mức xám, mức màu, độ nhám... Vùng ảnh là một trong hai thuộc tính của ảnh. Nói đến vùng ảnh là nói đến tính chất bề mặt. Đường bao quanh một vùng ảnh là biên ảnh (*Boundary*). Các điểm ảnh trong một vùng ảnh có độ biến thiên giá trị mức xám tương đối đồng đều hay tính kết cấu tương đồng.

Ngưỡng (*Threshold*) là một khái niệm khá quen thuộc trong xử lý ảnh cũng như rất nhiều giải thuật khác. Nó dùng để chỉ một giá trị mà người ta dựa vào để phân hoạch một tập hợp thành các miền phân biệt.

Giá trị ngưỡng thường được xác định dựa vào những điểm đặc biệt (ví dụ ở trung bình), dựa vào kinh nghiệm khảo sát. Nếu dựa vào số lượng ngưỡng áp dụng cho cùng một tập dữ liệu người ta sẽ phân ra các phương pháp ứng dụng ngưỡng đơn, ngưỡng kép, hay đa ngưỡng. Nếu dựa vào sự biến thiên của giá trị ngưỡng, trong cùng phạm vi ứng dụng người ta sẽ phân ra các phương pháp dùng ngưỡng cố định (*Constant/Fixed Threshold*) và không cố định (*Adaptive Threshold*). Ngưỡng không cố định nghĩa là giá trị của nó sẽ thay đổi tùy theo sự biến thiên của tập dữ liệu theo không gian và thời gian. Thông thường giá trị này được xác định thông qua khảo sát tập dữ liệu bằng phương pháp thống kê.

Xem xét các phương pháp khác nhau cho phân đoạn ảnh mức xám với kết quả cho ra có thể chấp nhận được thì phương pháp chọn ngưỡng cố định là một trong những phương pháp phân đoạn ảnh phổ biến nhất, bởi vì nó đơn giản và tương đối bền vững. Thông thường mức xám của điểm ảnh thuộc đối tượng cơ bản khác với mức xám của nền, bởi vậy bằng cách chọn ngưỡng thích hợp ta có thể dễ dàng tách riêng đối tượng và nền. Kết quả của chọn ngưỡng là ảnh bitmap có một trạng thái cho biết các đối tượng cận cảnh, như văn bản in, một chú thích, phần lỗi của vật liệu... và trạng thái còn lại sẽ tương ứng với nền. Tùy thuộc vào ứng dụng, cận cảnh có thể được biểu diễn bởi mức xám 0, tức là màu đen đối với văn bản, còn nền được biểu diễn bởi mức xám 255 trong ảnh 8-bit. Hoặc ngược lại, cận cảnh được biểu diễn bởi màu đen, nền bằng màu trắng.

Tư tưởng chính của phân đoạn ảnh:

- Cho ngưỡng t
- Phân đoạn ảnh

$$I[x,y] = \begin{cases} 0, & I[x,y] < t \\ 1, & I[x,y] \geq t \end{cases}$$

CHƯƠNG 2: CÁC HƯỚNG TIẾP CẬN CHÍNH TRONG PHÂN ĐOẠN ẢNH

Phân đoạn ảnh là chia ảnh thành các vùng không trùng lấp. Mỗi vùng gồm một nhóm pixel liên thông và đồng nhất theo một tiêu chí nào đó. Tiêu chí này phụ thuộc vào mục tiêu của quá trình phân đoạn. Ví dụ như đồng nhất về màu sắc, mức xám, kết cấu, độ sâu của các layer... Sau khi phân đoạn mỗi pixel chỉ thuộc về một vùng duy nhất. Để đánh giá chất lượng của quá trình phân đoạn là rất khó. Vì vậy trước khi phân đoạn ảnh cần xác định rõ mục tiêu của quá trình phân đoạn là gì. Xét một cách tổng quát, ta có thể chia các hướng tiếp cận phân đoạn ảnh thành ba nhóm chính như sau:

- Các phương pháp dựa trên không gian đặc trưng
- Các phương pháp dựa trên không gian ảnh
- Các phương pháp dựa trên mô hình vật lý.

2.1 Các phương pháp dựa trên không gian đặc trưng

Nếu chúng ta giả định màu sắc bề mặt của các đối tượng trong ảnh là một thuộc tính bất biến và các màu sắc đó được ánh xạ vào một không gian màu nào đó, vậy thì chúng ta sẽ có một cái nhìn đối với mỗi đối tượng trong ảnh như là một *cum* (*cluster*) các điểm trong không gian màu đó. Mức độ phân tán của các điểm trong trong một cum được xác định chủ yếu bởi sự khác biệt về màu sắc. Một cách khác, thay vì ánh xạ các pixel trong ảnh vào một không gian màu cụ thể, ta xây dựng một *histogram* dựa trên các đặc trưng màu dạng *ad-hoc* cho ảnh đó (ví dụ như Hue), và thông thường, các đối tượng trong ảnh sẽ xuất hiện như các giá trị đỉnh trong histogram đó. Do đó, việc phân vùng các đối tượng trong ảnh tương ứng với việc xác định các cum – đối với cách biểu diễn thứ nhất – hoặc xác định các vùng cực trị của histogram – đối với cách biểu diễn thứ hai.

Các phương pháp tiếp cận này chỉ làm việc trên một không gian màu xác định. Dựa trên không gian đặc trưng, ta có các phương pháp phân đoạn: phương pháp phân nhóm đối tượng không giám sát, phương pháp phân lớp trung bình-k thích nghi, phương pháp lấy ngưỡng histogram.

2.2 Các phương pháp dựa trên không gian ảnh

Hầu hết những phương pháp được đề cập trong phần trên đều hoạt động dựa trên các không gian đặc trưng của ảnh (thông thường là màu sắc). Do đó, các vùng

ảnh kết quả là đồng nhất tương ứng với các đặc trưng đã chọn cho từng không gian. Tuy nhiên, không có gì đảm bảo rằng tất cả các vùng này thể hiện một sự cô đọng (*compactness*) về nội dung xét theo ý nghĩa không gian ảnh (ý nghĩa các vùng theo sự cảm nhận của hệ thần kinh con người). Mà đặc tính này là quan trọng thứ hai sau đặc tính về sự đồng nhất của các vùng ảnh. Do các phương pháp gom cụm cũng như xác định ngưỡng histogram đã nêu đều bỏ qua thông tin về vị trí của các pixel trong ảnh.

Trong các báo cáo khoa học về phân vùng ảnh mức xám, có khá nhiều kỹ thuật cố thực hiện việc thoả mãn cùng lúc cả hai tiêu chí về tính đồng nhất trong không gian đặc trưng của ảnh và tính cô đọng về nội dung ảnh. Tuỳ theo các kỹ thuật mà các thuật giải này áp dụng, chúng được phân thành các nhóm sau:

- Các thuật giải áp dụng kỹ thuật chia và trộn vùng.
- Các thuật giải áp dụng kỹ thuật tăng trưởng vùng.
- Các thuật giải áp dụng lý thuyết đồ thị.
- Các giải thuật áp dụng mạng neural.
- Các giải thuật dựa trên cạnh.

2.3 Các phương pháp dựa trên mô hình vật lý

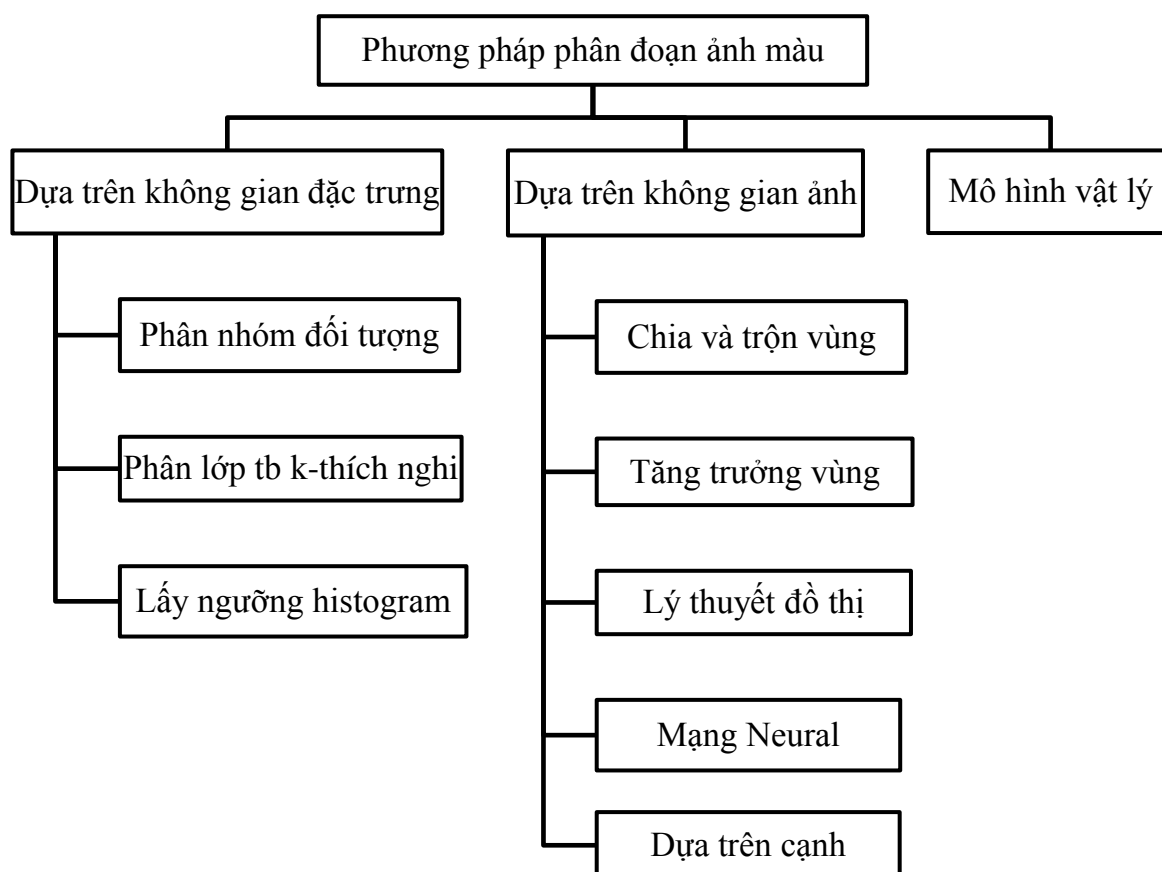
Tất cả các giải thuật được xem xét qua, không ít thì nhiều ở mặt nào đó đều có khả năng phát sinh việc phân vùng lỗi trong các trường hợp cụ thể nếu như các đối tượng trong ảnh màu bị ảnh hưởng quá nhiều bởi các vùng sáng hoặc bóng mờ, các hiện tượng này làm cho các màu đồng nhất trong ảnh thay đổi nhiều hoặc ít một cách đột ngột. Và kết quả là các thuật giải này tạo ra các kết quả phân vùng quá mức mong muốn so với sự cảm nhận các đối tượng trong ảnh bằng mắt thường. Để giải quyết vấn đề này, các giải thuật phân vùng ảnh áp dụng các mô hình tương tác vật lý giữa bề mặt các đối tượng với ánh sáng đã được đề xuất. Các công cụ toán học mà các phương pháp này sử dụng thì không khác mấy so với các phương pháp đã trình bày ở trên, điểm khác biệt chính là việc áp dụng các mô hình vật lý để minh hoạ các thuộc tính phản chiếu ánh sáng trên bề mặt màu sắc của các đối tượng.

Cột mốc quan trọng trong lĩnh vực phân vùng ảnh màu dựa trên mô hình vật lý được Shafer đặt ra. Ông giới thiệu *mô hình phản xạ lưỡng sắc* cho các vật chất điện môi không đồng nhất. Dựa trên mô hình này, Klinker đã đặt ra một giải thuật đặt ra một số giả thiết quang học liên quan đến màu sắc, bóng sáng, bóng mờ của các đối tượng và cố gắng làm phù hợp chúng với hình dạng của các cụm. Hạn chế chính của giải thuật này là nó chỉ làm việc trên các vật chất điện môi không đồng

nhất. Hai ông cùng tên Tsang đã áp dụng mô hình phản xạ lưỡng sắc trong không gian HSV để xác định các đường biên trong ảnh màu.

Healey đề xuất một *mô hình phản xạ đơn sắc* cho các vật chất kim loại. Các phương pháp đề cập trong phần này chỉ áp dụng cho hai loại vật chất là kim loại và điện môi không đồng nhất. Một thuật toán tổng quát và phức tạp hơn cũng được Maxwell và Shafer đề xuất trong.

- ✓ Tóm lại, một cái nhìn tổng quan về các phương pháp phân đoạn ảnh như sau:



- ✓ Mỗi phương pháp đều có những ưu nhược điểm nhất định:

<i>Phương pháp phân vùng</i>	<i>Ý tưởng</i>	<i>Ưu điểm</i>	<i>Khuyết điểm</i>
Dựa trên không gian đặc trưng			
Phân nhóm đối tượng	-	- Phân loại không cần giám sát.	- Không quan tâm đến các thông tin

		<ul style="list-style-type: none"> - Tồn tại các phương pháp heuristic và hữu hạn. 	<p>trong không gian ảnh.</p> <ul style="list-style-type: none"> - Có vấn đề trong việc xác định số lượng các cụm ban đầu. - Khó khăn trong việc điều chỉnh các cụm sao cho phù hợp với các vùng trong ảnh.
<p>Phân lớp trung bình k-thích nghi</p>	<ul style="list-style-type: none"> - Chọn một phần tử k, sau đó chọn ngẫu nhiên k trung tâm. Tính toán khoảng cách giữa pixel trung bình mỗi cụm. Từ khoảng cách này có thể tính toán trung bình mới của cụm và lặp lại quá trình cho đến khi mỗi pixel là một bộ phận của các cụm k. 	<ul style="list-style-type: none"> - Sở hữu tính liên tục trong không gian ảnh và tính thích nghi cục bộ đối với các vùng ảnh. - Sử dụng các ràng buộc về không gian ảnh. 	<ul style="list-style-type: none"> - Cực đại hoá một xác suất hậu nghiệm có thể bị sai do các cực trị địa phương. - Hội tụ chậm.
<p>Lấy ngưỡng histogram</p>	<ul style="list-style-type: none"> - Cho ngưỡng t - Phân đoạn ảnh $I[x,y]=\begin{cases} 0, & I[x,y] < t \\ 1, & I[x,y] \geq t \end{cases}$	<ul style="list-style-type: none"> - Không cần biết trước bất kỳ thông tin nào từ ảnh. - Các giải thuật nhanh và dễ dàng cài đặt. 	<ul style="list-style-type: none"> - Bỏ qua các thông tin về không gian ảnh. - Lấy ngưỡng trong các histogram đa chiều là một quá trình phức tạp. - Dễ ảnh hưởng bởi nhiễu xuất hiện trong ảnh.

Dựa trên không gian ảnh			
Chia và trộn vùng	<ul style="list-style-type: none"> - Dựa vào độ sáng của điểm ảnh để phân nhỏ vùng, sao cho các vùng là đồng nhất. - Trộn các vùng nhỏ theo tiêu chí nhất định. 	<ul style="list-style-type: none"> - Sử dụng các thông tin về không gian ảnh là chính. - Cho kết quả tốt với các ảnh chứa nhiều vùng màu đồng nhất. 	<ul style="list-style-type: none"> - Định nghĩa mức độ đồng nhất về màu sắc có thể phức tạp và khó khăn. - Quadtree có thể gây ra các kết quả không như mong muốn.
Tăng trưởng vùng	<ul style="list-style-type: none"> - Xem xét ảnh từ các miền nhỏ nhất rồi hợp chúng lại nếu thỏa mãn tiêu chuẩn như cùng màu, cùng mức xám... để được một miền đồng nhất lớn hơn. 	<ul style="list-style-type: none"> - Các vùng ảnh đồng nhất và liên thông. - Có một số thuật giải có tốc độ thực thi khá nhanh. 	<ul style="list-style-type: none"> - Tốn kém chi phí sử dụng bộ nhớ và tính toán. - Gặp khó khăn trong việc thu thập tập các điểm mầm và xác định các điều kiện đồng nhất đầy đủ. - Chịu ảnh hưởng bởi các đặc tính tự nhiên của kỹ thuật này.
Lý thuyết đồ thị	<ul style="list-style-type: none"> - Phát hiện biên giữa hai vùng của ảnh bằng cách so sánh sự khác nhau giữa nội vùng với sự - 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 	<ul style="list-style-type: none"> - Thể hiện tốt không gian ảnh bằng đồ thị. - Một số thuật toán có tốc độ thực hiện nhanh. 	<ul style="list-style-type: none"> - Một vài thuật giải mất khá nhiều thời gian thực hiện. - Các đặc trưng cục bộ đôi khi được sử dụng nhiều hơn các đặc trưng toàn cục.

	việc phân đoạn chính xác và hiệu quả.		
Mạng neural	-	<ul style="list-style-type: none"> - Mức độ song song hoá cao và có tốc độ thực thi nhanh. - Khả năng chống chịu tốt trước các thay đổi xấu. - Một công cụ hữu hiệu cho các ứng dụng nhận dạng và xử lý ảnh y khoa. 	<ul style="list-style-type: none"> - Màu sắc có thể làm tăng độ phức tạp của mạng. - Quá trình học cần phải biết trước số lượng các phân lớp/cụm.
Dựa trên cạnh	-	<ul style="list-style-type: none"> - Là phương pháp được hỗ trợ mạnh bởi các toán tử dò biên. - Có hiệu năng tốt với các ứng dụng dò biên đối tượng theo đường cong. 	<ul style="list-style-type: none"> - Khó khăn trong việc định nghĩa một hàm gradient cho các ảnh màu. - Nhiều hoặc các ảnh có độ tương phản kém ảnh hưởng xấu đến kết quả phân vùng.
Mô hình vật lý			
		<ul style="list-style-type: none"> - Khẳng định tính chắc chắn đối với các vùng bóng sáng/tối, và vùng bóng chuyển tiếp (diffuse hoặc shade) - Phân vùng các đối tượng dựa vào thành phần vật liệu cấu tạo 	<ul style="list-style-type: none"> - Bị giới hạn vào một số lượng nhất định các loại vật chất hình thành nên đối tượng. - Khó khăn trong việc xác định vùng bóng sáng và bóng chuyển tiếp trong các ảnh thực.

			<ul style="list-style-type: none"> - Một vài giải thuật đòi hỏi các thông tin về hình dạng đối tượng (không luôn luôn đáp ứng được). - Chi phí tính toán khá cao.
--	--	--	---

2.4 Một số thuật toán phân đoạn ảnh

2.4.1 Thuật toán Entropy cực đại

✓ *Tiến trình giải thuật*

- Chia ảnh làm 2 vùng w_0, w_1

Trong đó: $w_0 = \{g \mid g \leq \theta\}$

$$w_1 = \{g \mid g > \theta\}$$

- Tìm Entropy của 2 vùng

$$\text{Entropy}(w_0) = - \sum p_{0i} \log_2 p_{0i}$$

$$\text{Entropy}(w_1) = - \sum p_{1i} \log_2 p_{1i}$$

Trong đó: p_{0i} là xác suất điểm ảnh rơi vào vùng w_0

p_{1i} là xác suất điểm ảnh rơi vào vùng w_1

$$p_{0i} = \frac{h(i)}{T(w_0)} \quad i \leq \theta$$

$$p_{1i} = \frac{h(i)}{T(w_1)} \quad i > \theta$$

Với: $h(i)$ là giá trị histogram của mức xám i

$T(w_i)$ là tổng số điểm ảnh trong vùng w_i .

- Tìm θ sao cho :

Entropy(w_0) + Entropy(w_1) cực đại $\Rightarrow \theta$ là ngưỡng cần tìm

- ✓ *Cài đặt chương trình*
 - Thực hiện phân ngưỡng theo thuật toán Entropy cực đại trên từng màu R, G, B. Sau khi phân đoạn đưa ra biểu đồ histogram của từng màu và ngưỡng của nó.
 - Input: Ảnh cần phân đoạn.
Output: Ảnh đã được phân đoạn, biểu đồ histogram và ngưỡng.

```

void CImageSegmentationDoc::OnRunEntropy()
{
    CImageSegmentationDoc *pDoc = pImageSegmentationDoc;//
    CDC dcMem;
    BITMAP bm;
    pDoc->m_bmBitmap.GetBitmap(&bm);
    RGBQUAD *pRGB = new RGBQUAD[256];
    dcMem.CreateCompatibleDC(NULL);
    dcMem.SelectObject(pDoc->m_bmBitmap);
    ::GetDIBColorTable(dcMem,0,256,pRGB );
    BYTE* p = new BYTE[bm.bmWidth*bm.bmHeight];
    pDoc->m_bmBitmap.GetBitmapBits(bm.bmWidth*bm.bmHeight,p);

    int nguong=min+1;
    int maxnguong=0;
    double maxe=0;
    double tw0=0,tw1=0;
    double entropyw0=0,entropyw1=0,entropy=0;
    double p1[256];
    int h,w;
    int i,j;

```



```

h=bm.bmHeight;
w=bm.bmWidth;

while(nguong<max)
{
    entropyw0=entropyw1=entropy=0;

    for(i=min;i<nguong;i++) tw0+=Histo[i];
    for(j=nguong;j<=max;j++) tw1+=Histo[j];

    for(i=min;i<=max;i++)
    {
        if(i<nguong) p1[i]=(double)(Histo[i]/tw0);
        else p1[i]=(double)(Histo[i]/tw1);
    }
    for(i=min;i<=max;i++)
    {
        if(i<nguong)
        {
            entropyw0+=p1[i]*log2(p1[i]);
        }
        else
        {
            entropyw1+=p1[i]*log2(p1[i]);
        }
    }
    entropy=-(entropyw0)-(entropyw1);
}

```

```

//----- tim entropy max
if(nguong==min+1)
{
    maxe=entropy;
    maxnguong=nguong;
}

if(entropy>maxe)
{
    maxe=entropy;
    maxnguong=nguong;
}
nguong++;
}

nguong=maxnguong;
//----- hieu chinh anh
for(i=0;i<h;i++)
for(int j=0;j<w;j++)
{
    BYTE vtri = p[i*w+j];
    if(vtri>nguong) vtri=255;
    else vtri=0;
    p[i*w+j]=(BYTE)vtri;
}

::SetDIBColorTable(dcMem,0,255,pRGB);
pDoc->m_bmBitmap.SetBitmapBits(bm.bmWidth*bm.bmHeight,p);

```

```
pDoc->UpdateAllViews(NULL);
```

}

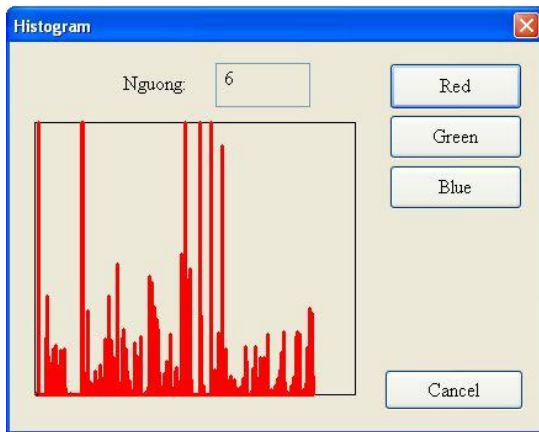
✓ Một số kết quả



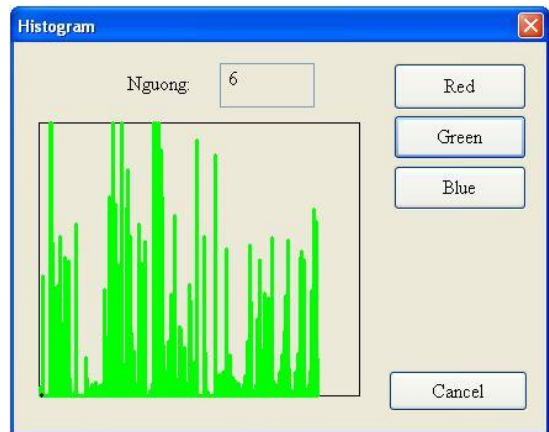
a)



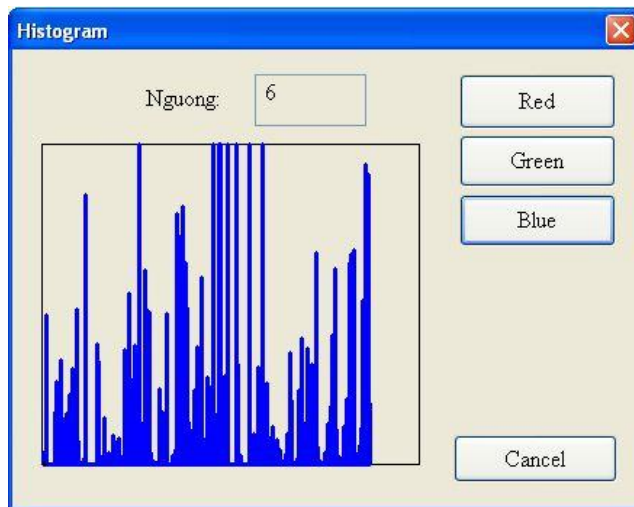
b)



c)



d)



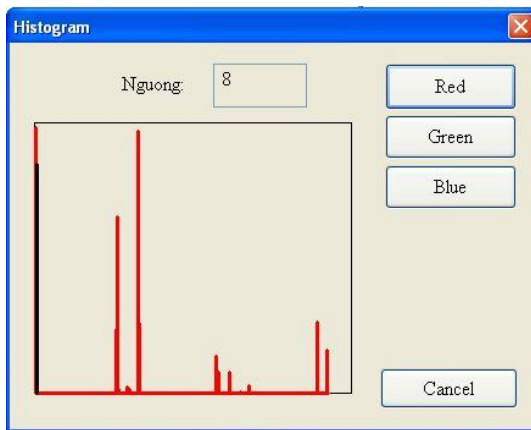
e)



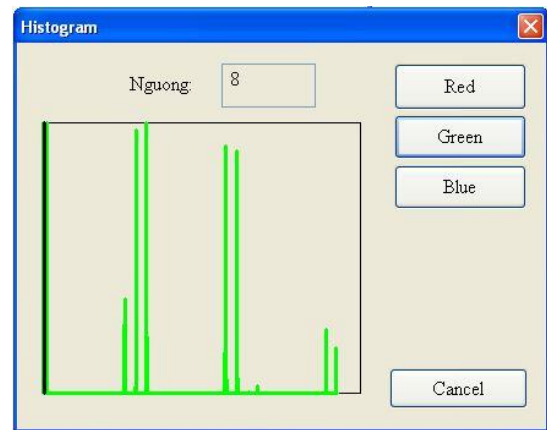
f)



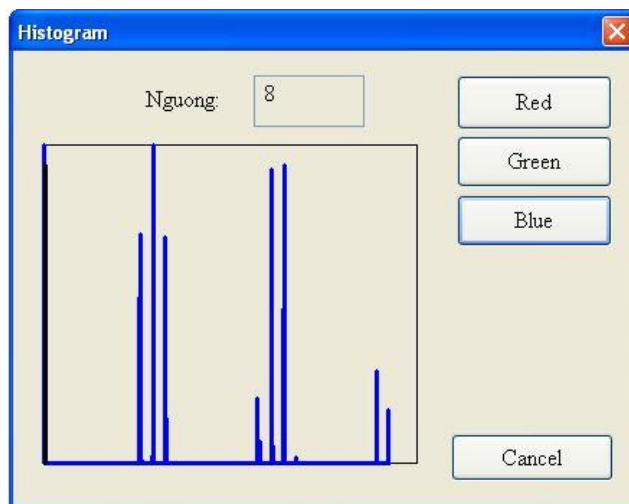
g)



h)



i)



k)

Hình 2.1. Phân đoạn theo thuật toán Entropy cực đại
 a) và f). Ảnh gốc. b) và g). Ảnh sau khi phân đoạn.
 c), d), e), h), i), k) là biểu đồ histogram và ngưỡng của các ảnh được phân đoạn (vạch đen là ngưỡng).

2.4.2 Thuật toán độ lệch nhỏ nhất

✓ *Tiến trình giải thuật*

- Chia ảnh thành 2 vùng w_0, w_1 như thuật toán Entropy cực đại
- Tính độ lệch chuẩn

$$\sigma_0^2 = \frac{\sum (h(i) - \bar{h}(w_0))^2}{|w_0|}$$

$$\sigma_1^2 = \frac{\sum (h(i) - \bar{h}(w_1))^2}{|w_1|}$$

Trong đó: $h(i)$ là giá trị histogram của mức xám i .

$|w_0|$ là lực lượng vùng w_0 .

$|w_1|$ là lực lượng vùng w_1 .

$$\bar{h}(w_0) = \frac{\sum_{i < \theta} h(i)}{|w_0|} \qquad \bar{h}(w_1) = \frac{\sum_{i \geq \theta} h(i)}{|w_1|}$$

- Tìm θ sao cho:

$\sigma_0^2 + \sigma_1^2$ đạt giá trị cực tiểu, khi đó θ là ngưỡng cần tìm

✓ *Cài đặt chương trình*

- Thực hiện phân ngưỡng theo thuật toán độ lệch nhỏ nhất trên từng màu R, G, B. Sau khi phân đoạn đưa ra biểu đồ histogram của từng màu và ngưỡng của nó.
- Input: Ảnh cần phân đoạn.
Output: Ảnh đã phân đoạn, biểu đồ histogram và ngưỡng.

```
void CImageSegmentationDoc::OnRunDolech()
```

```
{
```

```
    CImageSegmentationDoc *pDoc = pImageSegmentationDoc;//
```

```
    CDC dcMem;
```

```
    BITMAP bm;
```

```
    pDoc->m_bmBitmap.GetBitmap( &bm);
```

```
    RGBQUAD *pRGB = new RGBQUAD[256];
```

```

dcMem.CreateCompatibleDC(NULL);
dcMem.SelectObject(pDoc->m_bmBitmap);
::GetDIBColorTable(dcMem,0,256,pRGB );
BYTE* p = new BYTE[bm.bmWidth*bm.bmHeight];
pDoc->m_bmBitmap.GetBitmapBits(bm.bmWidth*bm.bmHeight,p);

```

```

int nguong=min+1;
int minl=0;
int minnguong=0;
double hw0=0,hw1=0;
double dolechw0=0,dolechw1=0,dolech=0;
int w1=0,w0=0;
double hw[256];
int h,w;
int i,j;

```

```

h=bm.bmHeight;
w=bm.bmWidth;

```

```

while(nguong<max)
{
    dolechw0=dolechw1=dolech=w1=w0=0;
    for(i=min;i<nguong;i++)
    {
        hw0+=Histo[i];
        w0++;
    }
    for(j=nguong;j<=max;j++)

```

```

{
    hw1+=Histo[i];
    w1++;
}
hw0=hw0/w0;
hw1=hw1/w1;
for(i=min;i<=max;i++)
{
    if(i<nguong)hw[i]=(double)((Histo[i]-hw0)*(Histo[i]-hw0));
    else hw[i]=(double)((Histo[i]-hw1)*(Histo[i]-hw1));
}
for(i=min;i<=max;i++)
{
    if(i<nguong) dolechw0+=hw[i];
    else dolechw1+=hw[i];
}
dolechw0=dolechw0/w0;
dolechw1=dolechw1/w1;
dolech=dolechw0+dolechw1;
//----- tim do lech min
if(dolech<minl)
{
    minnguong=nguong;
    minl=(int)dolech;
}
nguong++;
}
nguong=minnguong;

```

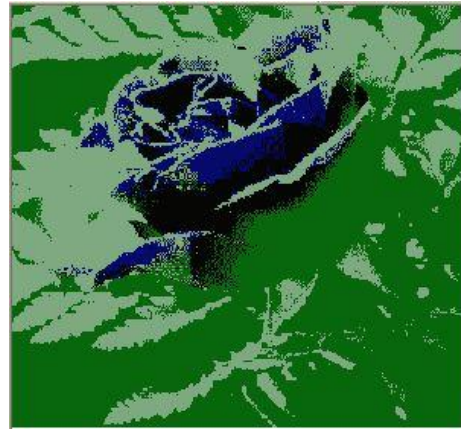
```
for(i=0;i<h;i++)
for(j=0;j<w;j++)
{
    BYTE vtri = p[i*w+j];
    if(vtri>nguong) vtri=255;
    else vtri=0;
    p[i*w+j]=(BYTE)vtri;
}

::SetDIBColorTable(dcMem,0,255,pRGB);
pDoc->m_bmBitmap.SetBitmapBits(bm.bmWidth*bm.bmHeight,p);
pDoc->UpdateAllViews(NULL);
}
```

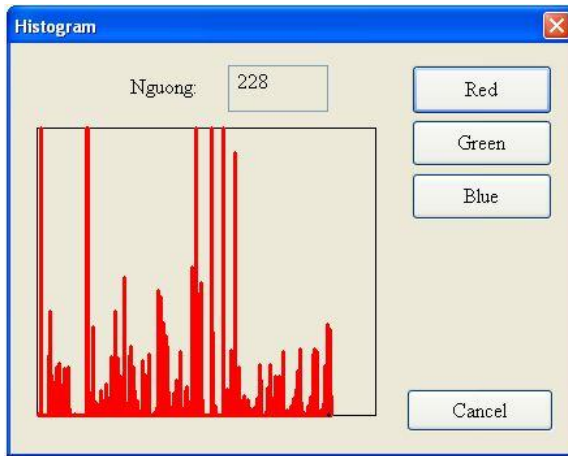

✓ Một số kết quả



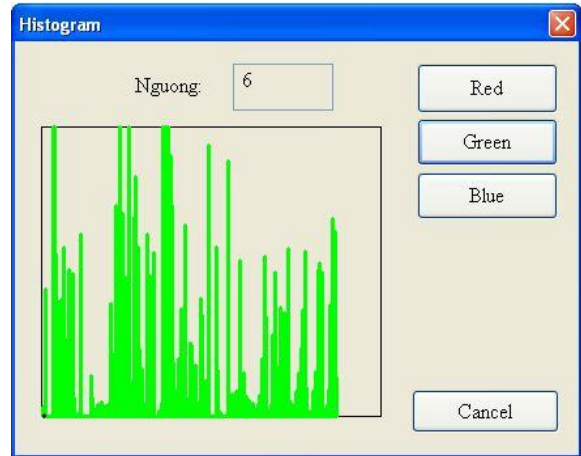
a)



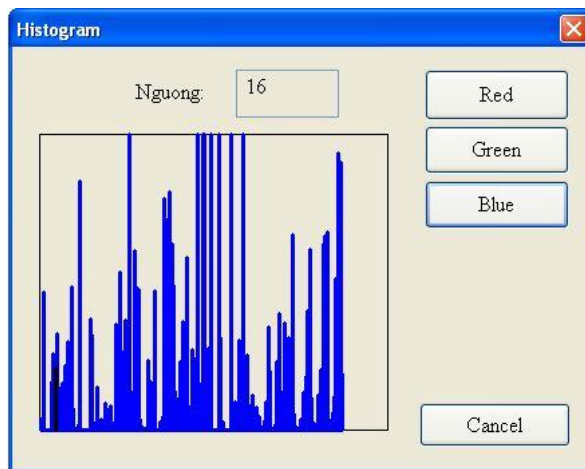
b)



c)



d)



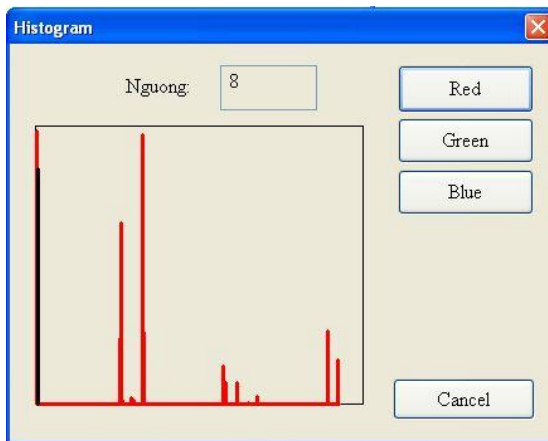
e)



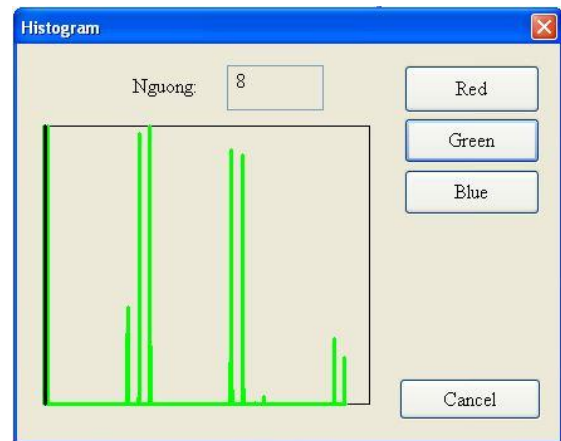
f)



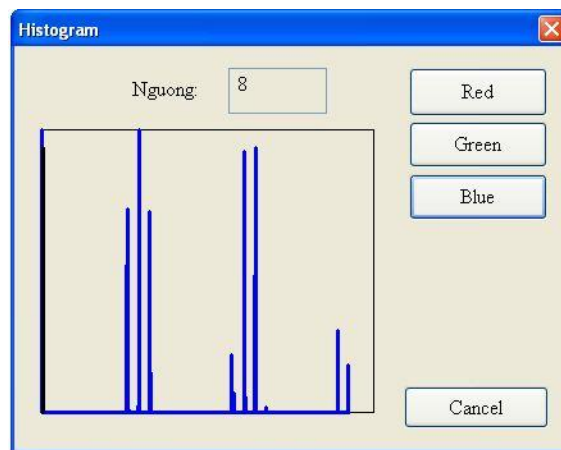
g)



h)



i)



k)

Hình 2.2. Phân đoạn theo thuật toán độ lệch nhỏ nhất.

a) và f). Ảnh gốc. b) và g). Ảnh sau khi phân đoạn.

c), d), e), h), i), k) là biểu đồ histogram và ngưỡng của các ảnh được phân đoạn (vạch đen là ngưỡng).

✓ *Đánh giá hai thuật toán:*

- Thuật toán đơn giản, cài đặt dễ dàng, thời gian tính toán nhanh.
- Tuy nhiên, cũng như nhược điểm của hầu hết các phương pháp phân đoạn dựa trên biểu đồ histogram là không quan tâm tới vị trí điểm ảnh trong không gian, dễ bị ảnh hưởng bởi nhiễu.

CHƯƠNG 3: PHƯƠNG PHÁP CỰC TIỂU NĂNG LƯỢNG DỰA TRÊN ĐỘ ĐỒNG NHẤT VÀ ĐỘ KHÔNG ỔN ĐỊNH CHO PHÂN ĐOẠN ẢNH

Hầu hết các phương pháp phân đoạn ảnh hiện nay dựa vào histogram, do đó, không khai thác được thông tin tạo bởi cường độ của ảnh. Ở đây, em xin giới thiệu một phương pháp tối ưu ngưỡng và gradient mới bằng cách tổ chức cường độ thành các đối tượng khác nhau của một ảnh, đó là phương pháp cực tiểu năng lượng dựa trên độ đồng nhất và độ không ổn định cho phân đoạn ảnh.

3.1 Giới thiệu

Phương pháp cực tiểu năng lượng dựa trên độ đồng nhất và độ không ổn định cho phân đoạn ảnh dựa trên độ không ổn định của lớp đối tượng và đặc trưng của histogram để xây dựng nên một hàm năng lượng cho gradient cùng đặc trưng của của đối tượng và hình khối trong một ảnh nhất định. Cuối cùng hàm năng lượng này được sử dụng để xác định một ngưỡng và gradient tối ưu. Phương pháp mới đồng thời xác định giá trị tối ưu cho cả ngưỡng và gradient của đối tượng khác nhau

3.1.1 Cơ sở lý thuyết

Cơ sở lý thuyết cho phương pháp này là các đối tượng được biểu diễn có biên mờ (*fuzzy boundaries*) trong hình ảnh thu được và trong cảm nhận bằng giác quan, cường độ có độ không ổn định cao kết hợp với biên ảnh thông thường sẽ cho biết được đối tượng.

3.1.2 Tối ưu và tự động ngưỡng

Thông thường, ngưỡng tối ưu cùng với chọn gradient là các vấn đề khó khăn trong cách tiếp cận phân đoạn nâng cao hoặc ít nhất là hướng tới phương pháp tự động. Tự động lựa chọn một ngưỡng chắc chắn và chính xác vẫn là một thách thức trong ảnh phân đoạn. Nhiều phương pháp lựa chọn ngưỡng tự động đã được báo cáo trong năm thập kỷ qua. Tuy nhiên, thiếu sót phổ biến của các cách tiếp cận là hoàn toàn dựa trên histogram mà không sử dụng đến số lượng đáng kể các thông tin nhúng trong các phân bố không gian của cường độ và hình thái trong ảnh. Thông thường, không thể cho một người quan sát để chọn ngưỡng cho một ảnh chỉ từ histogram của nó mà không thấy hình ảnh gốc. Mặt khác, ảnh chỉ chứa các phân vùng rõ ràng về đối tượng khác nhau thì việc chọn ngưỡng là quá dễ dàng. Chính vì thế, phương pháp cực tiểu năng lượng dựa trên độ đồng nhất và độ không ổn định sử dụng trực tiếp các hiển thị được tạo ra trên ảnh của đối tượng.

3.2 Lý thuyết

3.2.1 Cường độ dựa trên độ không ổn định

Ảnh số được biểu diễn bởi công thức: $\mathcal{C} = (\mathcal{C}, f)$

Trong đó: \mathcal{C} : xác định miền ảnh

$f|\mathcal{C} \rightarrow [I_{MIN}, I_{MAX}]$ với I_{MIN} và I_{MAX} biểu thị cường độ nhỏ nhất và lớn nhất.

\mathcal{C} thể hiện các điểm với tọa độ tách rời thuộc bên trong một hình hộp chữ nhật. Một phần tử của \mathcal{C} , thường ký hiệu là một vector \mathbf{p} , \mathbf{q} hoặc \mathbf{r} được gọi là một pixel 2 chiều (2D), một voxel trong không gian ba chiều (3D) và spel một trong không gian n chiều.

Cho $F_o \subset \mathcal{C}$ và $F_B \subset \mathcal{C}$ là các đối tượng giả định thực và lớp nền, tương ứng trong ảnh \mathcal{C} . Cho $p_o(g)$ biểu thị một xác suất tiêu nghiệm mà một đối tượng pixel có giá trị cường độ g . Ta có:

$$p_o(g) = P(f(\mathbf{p}) = g \mid \mathbf{p} \in F_o) \quad (3.1)$$

Khi đó, P là xác suất. Tương tự cho xác suất tiêu nghiệm $p_B(g)$ cho nền pixel có giá trị cường độ g .

$$p_B(g) = P(f(\mathbf{p}) = g \mid \mathbf{p} \in F_B) \quad (3.2)$$

Gọi θ là xác suất của pixel thuộc lớp đối tượng F_o , độc lập với cường độ của một pixel, sao cho $(1 - \theta)$ là xác suất của pixel các thuộc lớp nền F_B . Theo đó, θ được gọi là hàm mật độ. Vì vậy, xác suất mà pixel bất kỳ có giá trị cường độ g , thể hiện bởi $p(g)$, được tính như sau:

$$p(g) = \theta p_o(g) + (1 - \theta) p_B(g) \quad (3.3)$$

Sử dụng các phương trình trên, ta có xác suất hậu nghiệm mà một pixel với giá trị cường độ g , thuộc lớp đối tượng được xác định bằng cách sử dụng quy tắc Bayes [3].

$$P(\mathbf{p} \in F_o \mid f(\mathbf{p}) = g) = \frac{\theta p_o(g)}{p(g)} \quad (3.4)$$

Tương tự, ta có xác suất hậu nghiệm mà một pixel với giá trị cường độ g thuộc lớp nền được cho bởi công thức:

$$P(\mathbf{p} \in F_B \mid f(\mathbf{p}) = g) = \frac{(1 - \theta) p_B(g)}{p(g)} \quad (3.5)$$

Thước đo độ không ổn định để phân loại một pixel $\mathbf{p} \in C$ với giá trị cường độ g thuộc đối tượng hay lớp nền là entropy của hai giá trị xác suất hậu nghiệm như quy định tại phương trình (3.4) và (3.5). Biện pháp này được gọi là độ không ổn định và được ước tính theo phương trình entropy của Shannon và Weaver [4] như sau:

$$h(g) = -\frac{\theta p_o(g)}{p(g)} \log \frac{\theta p_o(g)}{p(g)} - \frac{(1-\theta)p(g)}{p(g)} \log \frac{(1-\theta)p(g)}{p(g)} \quad (3.6)$$

Ở đây, ý tưởng này là để mô hình một phân bố xác suất tiên nghiệm $p_o(g)$ và $p_B(g)$ và hàm mật độ θ như là một hàm chọn ngưỡng t và tham số gradient σ . Như vậy, bản đồ độ không ổn định của ảnh thay đổi như một hàm của ngưỡng t và tham số gradient σ , và chúng ta sử dụng $h_{t,\sigma}(g) \mid g \in [I_{MIN}, I_{MAX}]$ để biểu diễn ngưỡng và gradient phụ thuộc hàm độ không ổn định. Các phương pháp tính toán phân bố xác suất tiên nghiệm $p_o(g)$ và $p_B(g)$ và hàm mật độ θ như một hàm chọn các ngưỡng t và tham số gradient σ được giới thiệu trong phần sau.

3.2.2 Bề mặt năng lượng và tối ưu ngưỡng

Từ phương trình (3.6), ta có phạm vi tính toán của phương pháp độ không ổn định là $[0, 1]$. Do đó ta cần chuẩn hóa tham số gradient σ và có thể thay đổi. Ở đây, ta sử dụng mô hình Gauss để tìm ∇_σ chuẩn của cường độ gradient:

$$\nabla_\sigma(\mathbf{p}) = 1 - e^{-\frac{(\nabla(\mathbf{p}))^2}{2\sigma^2}} \quad (3.7)$$

Trong đó, ∇ là toán tử cường độ gradient và ∇_σ là toán tử gradient chuẩn. Lưu ý, đầu ra của toán tử ∇ phụ thuộc vào phạm vi cường độ ảnh, trong khi hiệu suất ∇_σ phụ thuộc vào đánh giá gradient trong phạm vi chuẩn hóa $[0,1]$. Cuối cùng hàm năng lượng E được tính như sau:

$$E(t,\sigma) = \sum_{\mathbf{p} \in C} \left(h_{t,\sigma}(f(\mathbf{p})) \times (1 - \nabla_\sigma(\mathbf{p})) + (1 - h_{t,\sigma}(f(\mathbf{p}))) \times \nabla_\sigma(\mathbf{p}) \right) \quad (3.8)$$

Theo phương trình trên, mỗi pixel \mathbf{p} tham gia vào năng lượng theo 2 cách

- Độ không ổn định nh cao và gradient thấp
- Độ không ổn định nh thấp và gradient cao

3.3 Phương pháp

Như giới thiệu ở phần trên, E là hàm năng lượng biến thiên theo hai thông số là ngưỡng t và gradient σ . Hai thông số này biểu diễn bề mặt năng lượng cho hàm E , đồng thời hàm E giúp tối ưu hai thông số này trên bề mặt năng lượng. Để triển

khai một thuật toán tối ưu dựa trên ngưỡng t và gradient σ , ta cần thực hiện các bước sau:

Cho trước các giá trị ngưỡng t và gradient σ , chúng ta cần tính:

- Phân bố xác suất tiên nghiệm đối tượng $p_o(g)$ và nền $p_B(g)$.
- Hàm mật độ θ
- Bản đồ gradient chuẩn ∇_σ .
- Tối ưu giá trị của t và σ trên bề mặt năng lượng E .

Trong các phần sau, em xin giới thiệu phương pháp để hoàn thành các nhiệm vụ trên.

Ban đầu, ảnh gốc được làm mờ bằng một hạt nhân Gaussian Blur

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.9)$$

Điều này nhằm mục đích:

- Tăng vùng có độ không chắc chắn và mịn nhiều trong vùng đồng nhất.
- Bản đồ độ không ổn định tăng phù hợp với việc tăng bản đồ gradient do làm mờ.

Ta có thể chỉ ra rằng tối ưu ngưỡng và gradient được áp dụng trên những ảnh gốc. Vì vậy, làm mờ được sử dụng trong quá trình tối ưu hóa ngưỡng và gradient mà không có bất kỳ ảnh hưởng về cấu trúc hay việc làm mờ ở phân đoạn cuối cùng.

3.3.1 Phân bố xác suất tiên nghiệm đối tượng $p_o(g)$ và nền $p_B(g)$

Theo giới thiệu ở trên, với giá trị bất kỳ cho trước của thông số ngưỡng t và gradient σ , ta cũng tính được đối tượng tiên nghiệm và phân phối cường độ nền theo công thức:

$$p_o(g) = \begin{cases} 1 & \text{if } g > t + 1.5\sigma \\ e^{-\frac{(g-(t+1.5\sigma))^2}{2\sigma^2}} & \text{if } g \leq t + 1.5\sigma \end{cases} \quad (3.10)$$

Và

$$p_B(g) = \begin{cases} 1 & \text{if } g < t - 1.5\sigma \\ e^{-\frac{(g-(t-1.5\sigma))^2}{2\sigma^2}} & \text{if } g \geq t - 1.5\sigma \end{cases} \quad (3.11)$$

3.3.2 Hàm mật độ θ

Sau khi làm mờ bằng hạt nhân Gaussian Blur [6], các hàm cường độ f được thay thế bằng hàm cường độ ảnh mờ f_{Blur} trong khi tính giá trị độ không chắc chắn. Hàm mật độ θ được tính bằng tỉ lệ số pixel của đối tượng trên tổng số pixel.

$$\theta = \frac{\sum_t^0 f_{Blur}(\mathbf{p})}{\sum f_{Blur}} \quad (3.12)$$

3.3.3 Bản đồ gradient chuẩn ∇_σ

Bản đồ gradient chuẩn được tính theo công thức sau:

$$\begin{aligned} \nabla_x(\mathbf{p}) &= f_{Blur}(\mathbf{p} + \mathbf{i}_x) - f_{Blur}(\mathbf{p} - \mathbf{i}_x) \\ \nabla_y(\mathbf{p}) &= f_{Blur}(\mathbf{p} + \mathbf{i}_y) - f_{Blur}(\mathbf{p} - \mathbf{i}_y) \\ \nabla(\mathbf{p}) &= \sqrt{\nabla_x^2(\mathbf{p}) + \nabla_y^2(\mathbf{p})} \end{aligned}$$

Trong đó, \mathbf{i}_x và \mathbf{i}_y là 2 vectơ đơn vị theo trục x và trục y.

Cuối cùng, bản đồ chuẩn hóa gradient được tính toán từ cường độ gradient sử dụng phương trình (3.7).

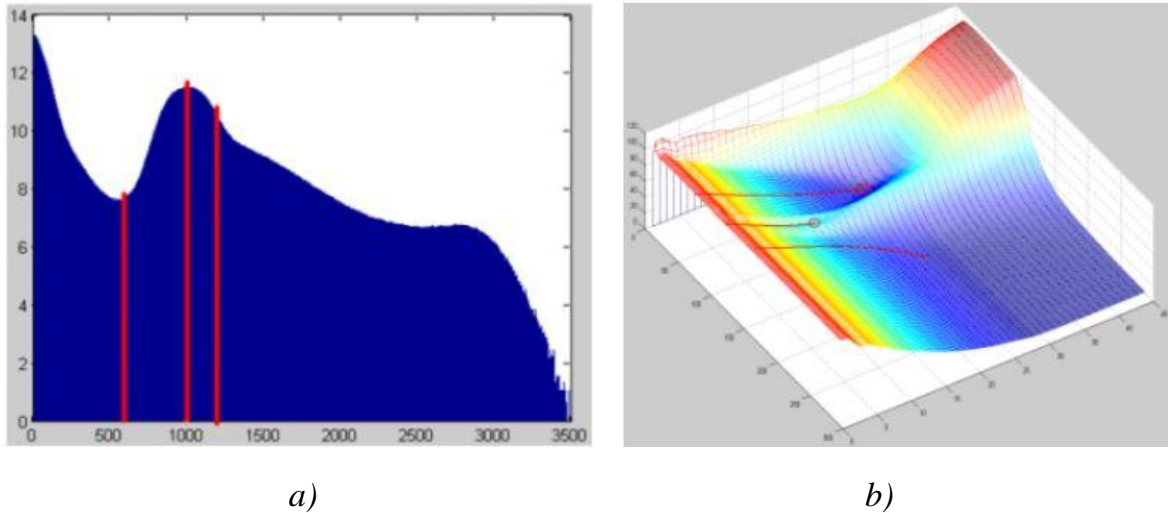
3.3.4 Tối ưu giá trị của t và σ trên bề mặt năng lượng E

Bây giờ, em xin giới thiệu các phương pháp tối ưu thông số ngưỡng t và gradient σ . Ở đây, chúng ta áp dụng kỹ thuật tìm kiếm vét cạn. Vì vậy, yếu tố quan trọng nhất ở đây là xác định hình học của các điểm tối ưu trên bề mặt năng lượng. Đối với thông số ngưỡng t , dải cường độ là $[I_{MIN}, I_{MAX}]$, được sử dụng để tìm kiếm các vị trí tối ưu. Mặt khác, tìm kiếm không gian cho các tham số gradient σ được thiết lập trong dải $[1\% \times (I_{MIN} - I_{MAX}), 40\% \times (I_{MIN} - I_{MAX})]$. Chúng ta xác định hai vị trí tối ưu trên bề mặt năng lượng (xem hình 3.2):

- Vị trí tối ưu loại I tạo thành hố (pit) trên bề mặt năng lượng E .
- Vị trí tối ưu loại II tạo thành thung lũng (valley) có ý nghĩa của một dòng năng lượng.

Gọi E_σ biểu diễn tham số gradient cố định tại giá trị σ cho trước và tham số ngưỡng thay đổi, khi đó E_σ tạo thành một đường cong năng lượng từ các giá trị tham số gradient σ . Cực tiểu địa phương trên bề mặt năng lượng E được gọi là hố,

còn cực tiểu của một dòng năng lượng E_σ được gọi là điểm thung lũng. Tùy thuộc vào độ phân giải của không gian tìm kiếm, cả E_σ và E có thể chứa một số lượng lớn các cực tiểu nhiều. Ở đây, em sử dụng ý tưởng của watershed, một ý tưởng tương tự như vực chứa nước sử dụng trong phương pháp phân đoạn watershed, để phân biệt giữa nhiều và cực tiểu có ý nghĩa.



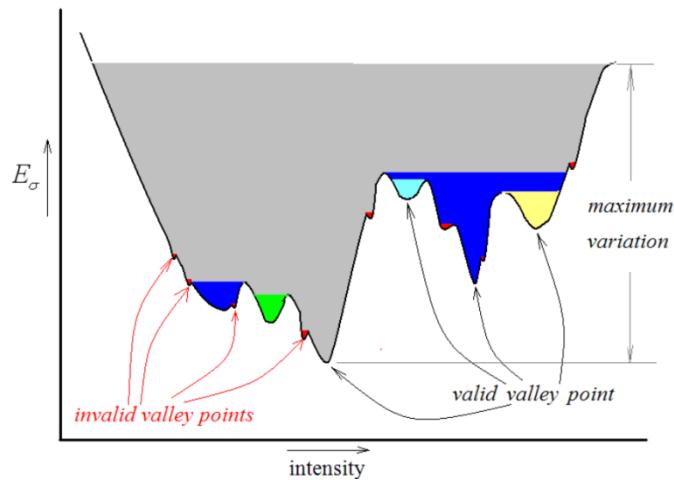
Hình 3.2. Ví dụ minh họa dòng năng lượng và bề mặt năng lượng.

a) Biểu một dòng năng lượng, các vạch đỏ biểu diễn các ngưỡng tối ưu hay các hố hợp lệ.

b) Biểu diễn bề mặt năng lượng, các vạch đỏ là các hố hợp lệ, khoanh tròn đỏ biểu diễn thung lũng ý nghĩa.

Gọi (t_1, σ_1) biểu thị hố, nghĩa là cực tiểu địa phương trên bề mặt năng lượng E . Các watershed của (t_1, σ_1) được biểu diễn bởi $B(t_1, \sigma_1)$, là tập hợp tất cả các vị trí (t, σ) , như vậy tồn tại một đường thẳng từ (t, σ) đến (t_1, σ_1) và tất cả các điểm có giá trị năng lượng lớn hơn hoặc bằng $E(t_1, \sigma_1)$. Về cơ bản, $B(t_1, \sigma_1)$ tương ứng với các vùng trên E các thể bị ngập bằng cách đổ nước từ trên tại $E(t_1, \sigma_1)$, hoặc không có nước bị rò rỉ tại một vị trí có giá trị năng lượng ít hơn $E(t_1, \sigma_1)$ (xem hình 3.1). Trong hình 3.1, các dòng màu đen biểu thị các dòng năng lượng của dải cường độ $[I_{MIN}, I_{MAX}]$ tại giá trị gradient σ_0 cho trước, mỗi màu cho thấy mỗi watershed sẽ cho một cực tiểu địa phương riêng, và độ sâu của vực là khoảng cách từ đỉnh của một màu đến đáy của nó, còn các vực màu đỏ là nhiều hay các điểm thung lũng không hợp lệ. Một hố (hoặc một điểm thung lũng) được xem là một hố hợp lệ (tương ứng một thung lũng hợp lệ) nếu chiều cao của $B(t_1, \sigma_1)$ (tương ứng $B_d(t)$) bao gồm ít

nhất 3% sự biến động tối đa của E (tương ứng E_σ). Ví dụ sự thay đổi lớn nhất của các đường cong năng lượng E_σ là các vực màu xám. Độ sâu của các vực màu đỏ nhỏ dưới 3% sự biến động tối đa và do đó không được coi là một thung lũng hợp lệ.



Hình 3.1. Minh họa vực bên trong.

Mỗi hố hợp lệ được xác định là một vị trí tối ưu loại I. Một thung lũng được định nghĩa là một con đường nối các điểm dọc theo thung lũng hợp lệ tham số gradient và một thung lũng được coi là ý nghĩa nếu chiều dài của nó bao gồm ít nhất 10% chiều dài tìm kiếm cùng các tham số gradient. Cuối cùng, một điểm tối ưu loại II được xác định ở trung tâm của một thung lũng có ý nghĩa.

3.4 Tiến trình giải thuật:

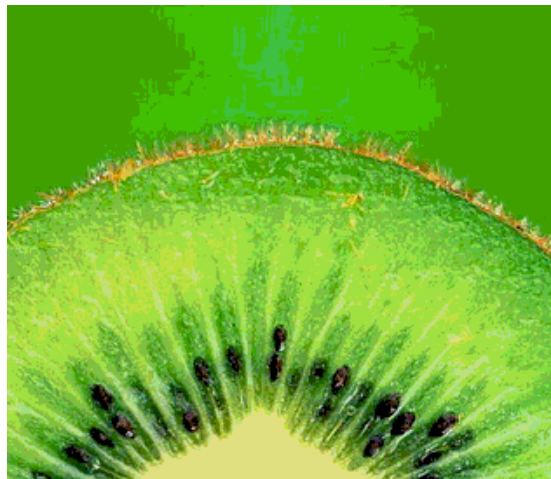
- ✓ Làm mờ ảnh gốc bởi hạt nhân Gaussian Blur (3.9).
- ✓ Cho trước một ngưỡng t_0 và gradient σ_0 . Xác định:
 - Hàm mật độ θ theo phương trình (3.12).
 - Phân bố xác suất tiên nghiệm của đối tượng $p_o(g)$ theo phương trình (3.10) và của nền $p_B(g)$ theo phương trình (3.11).
 - Gradient chuẩn theo phương trình (3.7).
 - Hàm năng lượng E theo phương trình (3.8) với tham số gradient σ thuộc dải cường độ $[1\% \times (I_{MIN} - I_{MAX}), 40\% \times (I_{MIN} - I_{MAX})]$ và ngưỡng t biến thiên trong dải $[I_{MIN}, I_{MAX}]$.
- ✓ Xác định các vị trí tối ưu loại I trên bề mặt năng lượng E và loại II của dòng năng lượng E_σ .

CHƯƠNG 4: CÀI ĐẶT CHƯƠNG TRÌNH VÀ ĐÁNH GIÁ

4.1 Cài đặt chương trình

4.1.1 Định dạng ảnh BMP

Trong đồ họa máy tính, BMP là một định dạng tệp tin hình ảnh khá phổ biến. Các tệp tin đồ họa lưu dưới dạng BMP thường có đuôi là .bmp (hình 4.1) hoặc .dib



Hình 4.1. Ảnh lưu dưới dạng BMP đuôi .bmp

Các thuộc tính tiêu biểu của một tệp tin ảnh BMP (cũng như file ảnh nói chung) là:

✓ Số pixel trên mỗi điểm ảnh, thường được ký hiệu là n . Một ảnh BMP n -bit có 2^n màu. Giá trị n càng lớn thì ảnh càng có nhiều màu và càng rõ nét hơn. Giá trị tiêu biểu của n là 1 (ảnh đen trắng), 4 (ảnh 16 màu), 8 (ảnh 256 màu), 16 (ảnh 65536 màu) và 24 (ảnh 16 triệu màu). Ảnh BMP 24-bit có chất lượng hình ảnh trung thực nhất.

- ✓ Chiều cao của ảnh (*height*), cho bởi điểm ảnh (*pixel*).
- ✓ Chiều rộng của ảnh (*width*), cho bởi điểm ảnh.

Cấu trúc một tệp tin ảnh BMP gồm 4 phần:

- ✓ *Bitmap Header* (14 bytes): giúp nhận dạng tệp tin bitmap.
- Lưu trữ thông tin cơ bản về tệp ảnh.

- Chứa chữ ký của ảnh có giá trị 0x4D42, cỡ tệp tin, lưu không, thông tin vùng bitmap information (lưu địa chỉ vùng dữ liệu).

- ✓ *Bitmap Information* (40 bytes): lưu một số thông tin chi tiết giúp hiển thị ảnh như chiều rộng ảnh, chiều cao ảnh, số màu, kiểu nén, độ phân giải ngang, dọc...

- ✓ *Color Palette* (4*n bytes), n là số màu của ảnh: định nghĩa các màu sẽ được sử dụng trong ảnh.

- Là mảng một chiều gồm n phần tử.

- Mỗi mảng gồm 4 phần tử 1 byte cho R, 1 byte cho G, 1 byte cho B, 1 byte cho cường độ (thường không dùng).

- ✓ *Bitmap Data*: lưu dữ liệu ảnh.

- Bố trí thành các dòng scanline.

- Lưu trữ từ dưới lên.

- Được dùng để XLA.

Đặc điểm nổi bật nhất của định dạng BMP là tệp tin hình ảnh thường không được nén bằng bất kỳ thuật toán nào. Khi lưu ảnh, các điểm ảnh được ghi trực tiếp vào tệp tin - một điểm ảnh sẽ được mô tả bởi một hay nhiều byte tùy thuộc vào giá trị n của ảnh. Do đó, một hình ảnh lưu dưới dạng BMP thường có kích cỡ rất lớn, gấp nhiều lần so với các ảnh được nén (chẳng hạn GIF, JPEG hay PNG).

4.1.2 Cài đặt thử nghiệm

- Giá trị gradient σ biến thiên trong dải $[1\% \times (I_{MIN} - I_{MAX}), 40\% \times (I_{MIN} - I_{MAX})]$ và với mỗi giá trị gradient σ thì ngưỡng t biến thiên trong dải $[I_{MIN}, I_{MAX}]$.

- Sử dụng các màu khác nhau cho các vùng đối tượng khác nhau.

- Đưa ra dòng năng lượng tại giá trị gradient σ hợp lệ lớn nhất.

```
void CImageSegmentationDoc::OnRunCuctieu(int ga)
```

```
{
```

```
    // TODO: Add your command handler code here
```

```
    CDC dcMem;
```

```

BITMAP bm;
m_bmBitmap.GetBitmap(&bm);
RGBQUAD *pRGB = new RGBQUAD[256];
dcMem.CreateCompatibleDC(NULL);
dcMem.SelectObject(m_bmBitmap);
::GetDIBColorTable(dcMem,0,256,pRGB );
BYTE* p = new BYTE[bm.bmWidth*bm.bmHeight];
BYTE* ptg = new BYTE[bm.bmWidth*bm.bmHeight];
m_bmBitmap.SetBitmapBits(bm.bmWidth*bm.bmHeight,pd);
m_bmBitmap.GetBitmapBits(bm.bmWidth*bm.bmHeight,p);
m_bmBitmap.GetBitmapBits(bm.bmWidth*bm.bmHeight,ptg);

int t=min; // nguong bien thien
int Thres[256]; // tap nguong
int gra=ga; // gradient
double he[256]; // Entropy
double Po[256],Pb[256],P[256]; // xs tien nghiem va hau nghiem
double E[256]; // nang luong
int mdS=0,mdo=0; // ham mat do
float tt=0.0;
int h,w;
int i,j,k,l;
double e=2.718281828;
double* grac = new double[bm.bmWidth*bm.bmHeight];
int dem1=0,dem2=0;

h=bm.bmHeight;
w=bm.bmWidth;

```

```

for(i=0;i<256;i++)
{
    Thres[i]=vitri[i]=0;
    he[i]=Po[i]=Pb[i]=P[i]=E[i]=0;
}

// Lam mo;

double kqr,kqg,kqb;
BYTE ByteRead;
int red,green,blue;
UINT r,g,b,th;

// Nhan chap
for(i=1;i<h;i++)
    for(j=1;j<w;j++)
    {
        kqr=kqg=kqb=0;

        for(k=-1;k<=1;k++)
            for(l=-1;l<=1;l++)
            {
                ByteRead=p[(i+k)*w+(j+l)];
                red=pRGB[ByteRead].rgbRed;
                green=pRGB[ByteRead].rgbGreen ;
                blue=pRGB[ByteRead].rgbBlue;
                kqr+=red;
                kqg+=green;
            }
    }

```

```

        kqb+=blue;
    }

    r=BYTE((double)kqr/9+0.5);
    g=BYTE((double)kqg/9+0.5);
    b=BYTE((double)kqb/9+0.5);
    th:::GetNearestPaletteIndex(hPal,RGB(r,g,b));

    ptg[i*w+j]=th;
}

// nguong bien thien tu min -> max
for(t=min-1;t<max;t++)
{
    mdS=mdo=0;

    // ham mat do voi nguong =10
    for(i=min;i<=max;i++)
    {
        mdS+=Histo[i];
        if(i<=t) mdo+=Histo[i];
    }
    tt=(float)mdo/mdS;

    // xs tien nghiem va hau nghiem
    for(i=min;i<=max;i++)
    {
        if (i>(t+1.5*gra)) Po[i]=1;
        else Po[i]=pow(e,(double)-(i-(t+1.5*gra))*(t+1.5*gra))/(2*gra*gra);
    }
}

```



```

if (i < (t - 1.5 * gra)) Pb[i] = 1;
else Pb[i] = pow(e, (double) - (i - (t - 1.5 * gra)) * (t - 1.5 * gra)) / (2 * gra * gra);
}

```

```

for (i = min; i <= max; i++)
{
    P[i] = (double) (tt * Po[i] + (1 - tt) * Pb[i]);
}

```

// Tinh gradient chuan

```

for (i = 0; i < h; i++)
    for (j = 0; j < w; j++)
    {
        double tg1, tg2, tg;
        double mu;
        if (i == 0 && j == 0)
        {
            tg1 = ptg[(i + 1) * w + j];
            tg2 = ptg[i * w + (j + 1)];
        }
        else if (i == 0)
        {
            tg1 = ptg[(i + 1) * w + j];
            tg2 = (ptg[i * w + (j + 1)] - ptg[i * w + (j - 1)]);
        }
        else if (j == 0)
        {
            tg1 = (ptg[(i - 1) * w + j] - ptg[(i + 1) * w + j]);
            tg2 = ptg[i * w + (j + 1)];
        }
    }

```

```

}
else if(i==(h-1) && j==(w-1))
{
    tg1=ptg[(i-1)*w+j];
    tg2=ptg[i*w+(j-1)];
}
else if(i==(h-1))
{
    tg1=ptg[(i-1)*w+j];
    tg2= (ptg[i*w+(j+1)] - ptg[i*w+(j-1)]);
}
else if(j==(w-1))
{
    tg1= (p[(i-1)*w+j] - p[(i+1)*w+j]);
    tg2=p[i*w+(j-1)];
}
else
{
    tg1= (p[(i-1)*w+j] - p[(i+1)*w+j]);
    tg2= (p[i*w+(j+1)] - p[i*w+(j-1)]);
}
tg=tg1*tg1+tg2*tg2;
mu=pow(e,(double)-tg/(2*gra*gra));
grac[i*w+j]=(double)(1-mu);
}

// pt entropy va nang luong
for(i=min;i<=max;i++)

```

```

{
    double tg1,tg2;
    tg1=(tt*Po[i])/P[i];
    tg2=((1-tt)*Pb[i])/P[i];
    he[i]=-tg1*log(tg1)-tg2*log(tg2);
}

for(i=0;i<h;i++)
for(j=0;j<w;j++)
{
    ByteRead=p[i*w+j];
    E[t]+=(he[ByteRead]*(1-grac[i*w+j])+(1-he[ByteRead])*grac[i*w+j]);
}
}

```

// Xac dinh vi tri toi uu loai I, II

```

for(i=min-1;i<max;i++)
{
    if(E[i]<E[i+1] && E[i]!=0)
    {
        Thres[dem1]=i;
        break;
    }
}

t=Thres[dem1];
while(t<=max)

```

```

{
    if(E[t]>E[t+1])
    {
        while(t<max)
        {
            if(E[t]<E[t+1] && E[i]!=0)
            {
                dem1++;
                Thres[dem1]=t;
                break;
            }
            t++;
        }
    }
    t++;
}

```

```

int gtri;
if(dem1!=0) gtri=255/dem1;
while(dem1>=0)
{
    for(i=0;i<h;i++)
        for(j=0;j<w;j++)
        {
            BYTE vtri = ptg[i*w+j];
            if(vtri<Thres[0]) vtri=255;
            else if(vtri>Thres[dem1])
            {

```

```
        vtri=0;
    }
    else if(Thres[dem1]<=vtri && vtri<Thres[dem1+1])
    {
        vtri=254-gtri;
        vtri-=gtri;
    }
    p[i*w+j]=(BYTE)vtri;
}
dem1--;
}
::SetDIBColorTable(dcMem,0,255,pRGB);
m_bmBitmap.SetBitmapBits(bm.bmWidth*bm.bmHeight,p);
UpdateAllViews(NULL);
}
```

4.2 Một số kết quả và đánh giá

- ✓ Một số kết quả
- Hình ảnh phân đoạn ảnh bông hoa với $\sigma \in [2, 90]$ và $t \in [5, 230]$.



a)



b)



c)

Hình 4.2. Kết quả phân đoạn ảnh bông hoa và biểu đồ dòng năng lượng

a) Ảnh gốc.

b) Ảnh sau khi phân đoạn.

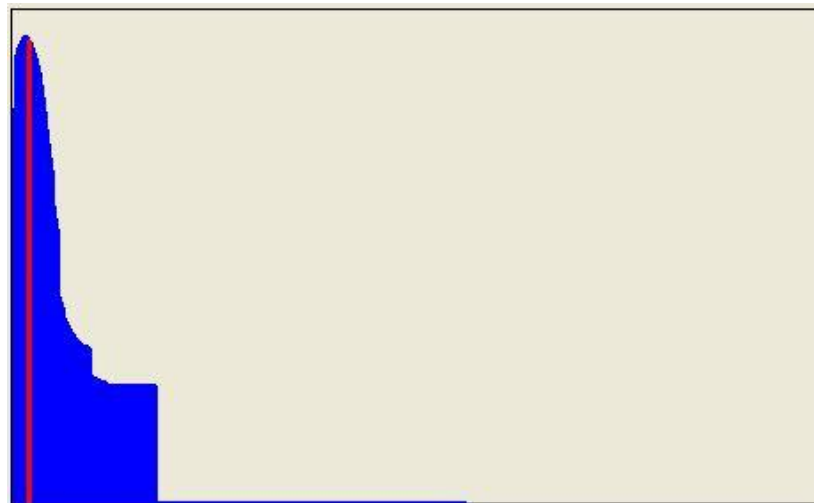
c) Biểu đồ dòng năng lượng tại $\sigma = 32$, màu đỏ biểu hiện giá trị năng lượng của tập ngưỡng tối ưu $t = \{7, 31, 40, 61, 67, 121\}$.

- Hình ảnh phân đoạn ảnh cô gái với $\sigma \in [2,96]$ và $t \in [7,247]$



a)

b)



c)

Hình 4.3. Kết quả phân đoạn ảnh cô gái và biểu đồ dòng năng lượng.

a) Ảnh gốc.

b) Ảnh sau khi phân đoạn.

c) Biểu đồ dòng năng lượng tại $\sigma = 74$, màu đỏ biểu hiện giá trị năng lượng của ngưỡng tối ưu $t=9$.

✓ *Đánh giá*

Phương pháp cực tiểu năng lượng dựa trên độ đồng nhất và độ không ổn định có thể tự động xác định nhiều hơn một ngưỡng cần thiết cho một ảnh. Tuy nhiên, thuật toán phức tạp, mất khá nhiều thời gian tính toán.

KẾT LUẬN

Trong quá trình nghiên cứu tài liệu và thực hiện đồ án dưới sự hướng dẫn của PGS TS. Ngô Quốc Tạo – Viện CNTT, Viện KH&CN Việt Nam, em thấy bản thân đã đạt được một số kết quả như sau:

✓ Tìm hiểu một cách tổng quan về XLA và phân đoạn ảnh, em đã có một cách nhìn hệ thống về các hướng tiếp cận chính trong phân đoạn ảnh và một số thuật toán. Đồng thời biết được ưu điểm cũng như nhược điểm của từng hướng tiếp cận để có thể đưa ra cách lựa chọn phù hợp với từng loại ảnh.

✓ Em đã tìm hiểu và cài đặt được phương pháp cực tiểu năng lượng dựa trên độ đồng nhất và độ không ổn định cho phân đoạn ảnh. Phương pháp bề mặt năng lượng để tối ưu đồng thời ngưỡng và gradient để tăng cường khả năng phân đoạn.

✓ 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 đề tài nghiên cứu khoa học.

Bên cạnh những kết quả đạt được em tự thấy bản đồ án vẫn còn một số hạn chế:

✓ 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 đầy đủ và khoa học.

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 đồ án. 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 phương pháp cực tiểu năng lượng dựa trên độ đồng nhất và độ không ổn định đối với ảnh động hoặc các đoạn video ngắn.

TÀI LIỆU THAM KHẢO

Tài liệu tiếng việt:

[1] TS Đỗ Năng Toàn, “Giáo trình môn học Xử lý ảnh”, Khoa CNTT – Trường đại học Thái nguyên.

[2] PGS. TS Nguyễn Quang Hoan, “Xử lý ảnh”, Học viện Bưu chính viễn thông.

[3] “Nhập môn xử lý ảnh”, Đại học bách khoa Hà Nội.

Tài liệu tiếng anh:

[4] P. K. Saha and J. K. Udupa, “A new Optimum Thresholding Method Using Region Homogeneity and Class Unvertainty”, Proc SPIE: Medical Imaging 2000, vol. 3979, phương pháp. 1.80-191, 2000.

Website:

[5] <http://www.fit.hcmup.edu.vn/~hais/Xu%20Ly%20Anh/>

[6] <http://www.ieev.org/2009/06/gaussian-blur.html#comment-form>

[7] <http://www.bic.mni.mcgill.ca/brainweb/>