

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

---



**ISO 9001:2015**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Nguyễn Đình Phúc**

**Giảng viên hướng dẫn: TS. Ngô Trường Giang**

**HẢI PHÒNG - 2018**

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

-----

**PHÂN ĐOẠN ẢNH DỰA TRÊN THUẬT TOÁN NỞ VÙNG**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**  
**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Nguyễn Đình Phúc**

**Giảng viên hướng dẫn : TS. Ngô Trường Giang**

**HẢI PHÒNG - 2018**

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

-----

**NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

**Sinh viên: Nguyễn Đình Phúc**

**Mã SV: 1112401048**

**Lớp: CT1701**

**Ngành: Công nghệ thông tin**

**Tên đề tài: Phân đoạn ảnh dựa trên thuật toán nở vùng**

## LỜI CẢM ƠN

Trước hết em xin cảm ơn các thầy cô giáo trong khoa Công nghệ thông tin – Trường đại học Dân Lập Hải Phòng đã tạo mọi điều kiện thuận lợi cho em trong quá trình học tập tại trường. Đặc biệt, em xin chân thành cảm ơn sự hướng dẫn tận tình của TS. Ngô Trường Giang – giảng viên khoa Công nghệ thông tin trường Đại học Dân Lập Hải Phòng đã tạo mọi điều kiện giúp đỡ em hoàn thành đồ án.

Mặc dù cố gắng hết sức cùng sự tận tâm của thầy giáo hướng dẫn xong do trình độ còn hạn chế, nội dung đề tài còn khá mới mẻ với em nên khó tránh khỏi những sai sót trong quá trình tiếp nhận kiến thức. Em rất mong chỉ dẫn của thầy cô và sự góp ý của bạn bè để em có thể hoàn thiện đồ án của em.

Cuối cùng em xin gửi lời cảm ơn đặc biệt nhất tới gia đình, bố, mẹ, những người động viên, khích lệ để giúp em hoàn thành đồ án này.

Em rất mong nhận được những sự góp ý của thầy cô giáo và các bạn sinh viên để đề tài của em được hoàn thiện hơn.

**Em xin chân thành cảm ơn!**

Hải Phòng, ngày 26 tháng 3 năm 2018

*Người thực hiện*  
Nguyễn Đình Phúc

## MỤC LỤC

<b>LỜI CẢM ƠN.....</b>	<b>4</b>
<b>MỤC LỤC .....</b>	<b>5</b>
<b>DANH MỤC HÌNH ẢNH.....</b>	<b>7</b>
<b>CHƯƠNG 1: MỘT SỐ VẤN ĐỀ CƠ BẢN TRONG XỬ LÝ ẢNH.....</b>	<b>9</b>
1.1 Một số khái niệm .....	9
1.1.1 Khái niệm về ảnh số.....	9
1.1.2 Tổng quan về một hệ thống xử lý ảnh .....	10
1.2 Kỹ thuật lọc nhiễu trong xử lý ảnh .....	11
1.2.1 Nhân chập.....	11
1.2.2 Lọc số là gì? .....	13
1.2.3 Một số kỹ thuật lọc nhiễu.....	14
1.3 Kỹ thuật tìm biên trong xử lý ảnh.....	17
1.3.1 Một số khái niệm.....	17
1.3.2 Các phương pháp phát hiện biên.....	17
1.4 Biến đổi khoảng cách.....	20
1.4.1 Giới thiệu.....	20
1.4.2 Thuật toán biến đổi khoảng cách đơn giản. ....	20
1.5 Phân đoạn ảnh .....	21
1.5.1 Giới thiệu.....	21
1.5.2 Một số hướng tiếp cận trong phân đoạn ảnh.....	22
1.5.3 Phân đoạn dựa trên ngưỡng .....	26
<b>CHƯƠNG 2: PHÂN ĐOẠN ẢNH VỚI THUẬT TOÁN NỞ VÙNG .....</b>	<b>29</b>
2.1 Thuật toán nở vùng .....	29
2.2 Một số thuật toán nở vùng .....	29
2.2.1 Thuật toán nở vùng cơ bản.....	29
2.2.2 Thuật toán nở vùng thống kê .....	30
2.2.3 Thuật toán nở vùng theo lưu vực .....	30
2.3 Phân đoạn ảnh theo biến đổi Watershed.....	31

2.3.1	Giới thiệu: .....	31
2.3.2	Thuật toán watershed dựa trên các thành phần liên thông.....	35
<b>CHƯƠNG 3: THỰC NGHIỆM .....</b>		<b>47</b>
3.1	Phát biểu bài toán ứng dụng.....	47
3.2	Khái quát về thư viện OpenCV.....	48
3.3	Các bước thực hiện .....	48
3.3.1	Bước 1: Tiền xử lý .....	49
3.3.2	Bước 2: Tách đối tượng .....	54
3.3.3	Xác định ranh giới giữa các đối tượng.....	56
<b>KẾT LUẬN.....</b>		<b>59</b>
<b>MỘT SỐ TÀI LIỆU THAM KHẢO .....</b>		<b>60</b>

## DANH MỤC HÌNH ẢNH

Hình 1-1. Tổng quan về hệ thống xử lý ảnh	10
Hình 1-2. Hình ảnh minh họa về phép lọc trung bình	14
Hình 1-3. Hình ảnh minh họa về phép lọc trung vị	15
Hình 1-4. Kết quả lọc ảnh theo 2 phương pháp trung bình và trung vị	16
Hình 1-5. Một số kiểu đường biên thông dụng	17
Hình 1-6. Toán tử Sobel	19
Hình 1-7. Toán tử Prewitt	19
Hình 1-8. Toán tử Robert	19
Hình 1-9. Ví dụ về phân đoạn ảnh	21
Hình 1-10. Một số hướng tiếp cận phân đoạn ảnh	22
Hình 1-11. Ví dụ ảnh gốc trước khi áp dụng phân ngưỡng	27
Hình 1-12. Ảnh sau khi phân ngưỡng toàn cục với mức ngưỡng 150	28
Hình 1-13. Ảnh phân ngưỡng với ngưỡng động	28
Hình 2-1. Mô tả hình ảnh lưu vực của 2 vùng	31
Hình 2-2. Hình ảnh minh họa thuật toán watershed	32
Hình 2-3. Mô tả thuật toán watershed theo nguyên lý nước dâng	33
Hình 2-4. Thuật toán dòng chảy và ngưỡng chìm	35
Hình 2-5. Ảnh gốc	36
Hình 2-6. Các pixel lân cận liên kết tới điểm cực tiểu	36
Hình 2-7. Nhãn được gán cho các điểm ảnh	36
Hình 2-8. Giá trị mức xám của ảnh đầu vào	37
Hình 2-9. Giá trị $v(p)$ của từng điểm ảnh sau khi chạy bước 1	38
Hình 2-10. Giá trị xám của ảnh đầu vào	39
Hình 2-11. Giá trị $v(p)$ của ảnh sau khi chạy bước 1	40
Hình 2-12. Giá trị $v(p)$ của các điểm ảnh sau khi chạy bước 2	40
Hình 2-13. Giá trị $v(p)$ sau khi đã hoàn tất các bước quét ảnh	40
Hình 2-14. Mức xám của điểm ảnh đầu vào	42
Hình 2-15. Nhãn mới được gán sau bước quét xuống lần 1	42
Hình 2-16. Nhãn thay đổi khi thực hiện phép quét từ dưới lên trên lần 1	42
Hình 2-17. Quét từ trên xuống dưới lần 2	43
Hình 2-18. Quét từ dưới lên lần 2	43
Hình 2-19. Quét ảnh từ trên xuống dưới lần 3	43
Hình 2-20. Hình ảnh gán nhãn cuối cùng	44
Hình 3-1. Ảnh thực nghiệm	47
Hình 3-2. Giao diện chương trình cài đặt	49
Hình 3-3. Hình ảnh đầu vào	50
Hình 3-4. Ảnh xám sau khi được chuyển đổi	52
Hình 3-5. Kết quả của bước làm mờ ảnh	53
Hình 3-6. Kết quả của quá trình phân ngưỡng	54
Hình 3-7. Hình ảnh sau khi sử dụng hàm biến đổi khoảng cách	55
Hình 3-8. Các đối tượng đã được tách	55
Hình 3-9. Hình ảnh phân đoạn bằng thuật toán watershed	58

Hình 3-10. Số lượng đồng xu đếm được

58



## CHƯƠNG 1: MỘT SỐ VẤN ĐỀ CƠ BẢN TRONG XỬ LÝ ẢNH

### 1.1 Một số khái niệm

#### 1.1.1 Khái niệm về ảnh số

Ảnh số là tập hợp hữu hạn các điểm ảnh với mức xám phù hợp dùng để mô tả gần nhất với ảnh thật. Số điểm ảnh xác định độ phân giải của ảnh, độ phân giải càng cao thì càng thể hiện rõ nét các đặc điểm của tấm hình, càng làm cho tấm ảnh trở nên thực và sắc nét hơn.

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

Mức xám là kết quả của sự biến đổi tương ứng một giá trị độ sáng của một điểm ảnh với một giá trị nguyên dương. Thông thường nó xác định trong khoảng từ 0 đến 255 tùy thuộc vào giá trị mà mỗi điểm ảnh được biểu diễn.

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

Ảnh đen trắng là ảnh chỉ bao gồm 2 màu: màu đen và màu trắng. Người ta phân mức đen trắng đó thành  $L$  mức. Nếu sử dụng số bit  $B$  để mã hóa mức đen trắng (hay mức xám) thì  $L$  được xác định:  $L = 2^B$ . Nếu  $B=1$ , thì chỉ có 2 mức: mức 0 và mức 1, còn gọi là ảnh nhị phân. Mức 1 ứng với màu sáng, còn mức 0 ứng với màu tối. Nói cách khác, với ảnh nhị phân mỗi điểm ảnh được mã hóa trên 1 bit. Nếu  $L$  lớn hơn 2 ta có ảnh đa cấp xám. Nếu dùng 8 bit để biểu diễn mức xám, số các mức xám có thể biểu diễn được là 256. Mỗi mức xám được biểu diễn dưới dạng là một số nguyên nằm trong khoảng từ 0 đến

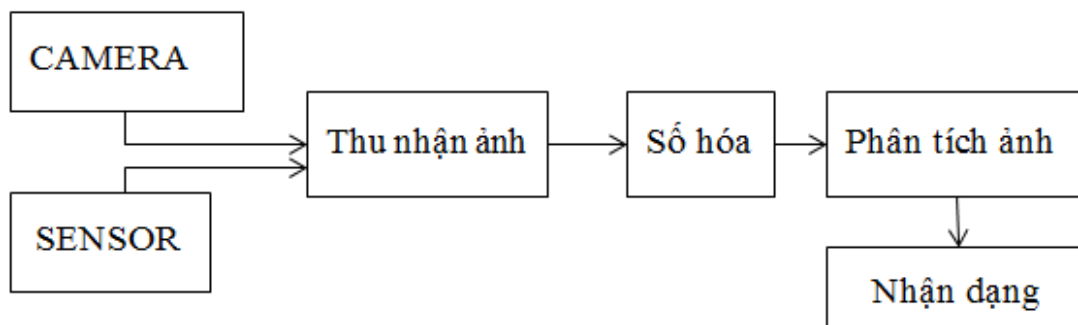
255, với mức 0 biểu diễn cho mức cường độ đen nhất và 255 biểu diễn cho mức cường độ sáng nhất.

Ảnh màu là ảnh được tổ hợp từ 3 màu cơ bản: đỏ (R) , lục (G) , lam (B) . Với ảnh màu, người ta lưu trữ thành từng màu riêng biệt, mỗi màu được lưu trữ như một ảnh đa cấp xám nên không gian nhớ dành cho một ảnh màu lớn hơn 3 lần một ảnh đa cấp xám cùng kích cỡ.

### 1.1.2 Tổng quan về một hệ thống xử lý ảnh

Xử lý ảnh là một ngành khoa học tương đối mới mẻ so với các ngành khoa học khác, nhất là trên quy mô công nghiệp. Tuy nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó.

Tổng quan về một hệ thống xử lý ảnh được thể hiện bằng hình ảnh bên dưới:



**Hình 1-1. Tổng quan về hệ thống xử lý ảnh**

Trước hết là quá trình thu nhận ảnh, ảnh có thể được thu nhận qua camera, thường ảnh được nhận qua camera là tín hiệu tương tự (loại camera ống kính CCIR) , nhưng cũng có thể là tín hiệu số hóa (loại CCD-Charge Coupled Device) . Ảnh cũng có thể được thu nhận qua các bộ cảm ứng (sensor) , ảnh được quét trên scanner.

Số hóa (Digitalizer) là quá trình biến đổi tín hiệu tương tự sang tín hiệu rời rạc (lấy mẫu) và số hóa bằng lượng hóa trước khi chuyển sang giai đoạn xử lý, phân tích hay lưu trữ lại.

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

Cuối cùng, tùy theo mục đích của người dùng sẽ là giai đoạn nhận dạng, phân lớp hoặc là các quyết định khác.

## 1.2 Kỹ thuật lọc nhiễu trong xử lý ảnh

### 1.2.1 Nhân chập

Là quá trình thay đổi giá trị pixel ban đầu của ảnh đầu vào sang một giá trị mới bằng cách sử dụng một ma trận hay một cửa sổ nhân chập. Toàn bộ các điểm ảnh (Pixel) trên ảnh sẽ được tiến hành nhân chập với ma trận, tâm của ma trận sẽ được đặt trùng vào vị trí của điểm ảnh (Pixel) đang được tính nhân chập làm thay đổi các giá trị của pixel ban đầu.

Giá trị của pixel ban đầu được thay đổi theo công thức (1. 1)

$$I_{dst}(x, y) = I_{src}(x, y) * M(u, v) = \sum_{u=-n}^n \sum_{v=-n}^n I_{src}(x+u, y+v) * M(u, v) \quad (1. 1)$$

Trong đó :



Sau khi thực hiện phép nhân chập, giá trị điểm ảnh  $I_{src}(2, 2)$  từ giá trị 7 sẽ được thay đổi thành 201.

### 1.2.2 Lọc số là gì?

Một hệ thống dùng để làm biến dạng sự phân bố tần số của các thành phần tần số theo các chỉ tiêu đã cho được gọi là bộ lọc số. Lọc số có ý nghĩa quan trọng trong việc tạo ra các hiệu ứng trong ảnh, một số hiệu ứng nhờ sử dụng các bộ lọc làm mờ (Blur), làm trơn (Smooth).

Nguyên tắc chung của các phương pháp lọc số là cho ma trận ảnh nhân chập với một ma trận lọc (Kernel) hay còn được gọi là các phép tính nhân chập trên ảnh. Với mỗi phép lọc ta có những ma trận lọc khác nhau, không có một quy định cụ thể nào cho việc xác định M. Kích thước ma trận M là một số lẻ (ví dụ: 3x3, 5x5 ...). Ma trận lọc còn có thể được gọi là cửa sổ chập, cửa sổ lọc, mặt nạ...

Nguyên tắc lọc ảnh được thể hiện qua công thức (1. 2)

$$I_{dst} = M * I_{src} \quad (1. 2)$$

Trong đó:

- $I_{src}$ : Là ảnh gốc được sử dụng để lọc số ảnh.
- $I_{dst}$ : Là ảnh ra sau khi thực hiện xong phép lọc số ảnh.
- M: Là ma trận lọc (Mask, kernel).
- \*: Là phép toán nhân chập

$T_{pt}$  là tổng các phần tử trong ma trận M. Khi đó tổng  $T_{pt}$  các phần tử trong ma trận M thường là 1.

- Nếu  $T_{pt} > 1$ : Ảnh sau khi thực hiện xong phép lọc số ảnh ( $I_{dst}$ ) có độ sáng lớn hơn so với ảnh ban đầu ( $I_{src}$ ).

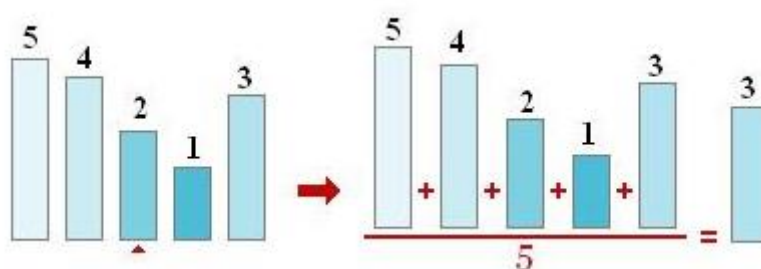
- Nếu  $T_{pt} < 1$ : Ảnh sau khi thực hiện xong phép lọc số ảnh ( $I_{dst}$ ) có độ sáng nhỏ hơn so với ảnh ban đầu ( $I_{src}$ ).

### 1.2.3 Một số kỹ thuật lọc nhiễu

#### Lọc trung bình

Lọc trung bình là kỹ thuật lọc tuyến tính, hoạt động như một bộ lọc thông thấp. Ý tưởng chính của thuật toán lọc trung bình như sau: Sử dụng một cửa sổ lọc (ma trận  $3 \times 3$ ) quét qua lần lượt từng điểm ảnh của ảnh đầu vào input. Tại vị trí mỗi điểm ảnh lấy giá trị của các điểm ảnh tương ứng trong vùng  $3 \times 3$  của ảnh gốc lắp vào ma trận lọc. Giá trị điểm ảnh của ảnh đầu ra là giá trị trung bình của tất cả các điểm ảnh trong cửa sổ lọc. Có thể chia việc tính toán này làm hai bước gồm tính tổng các thành phần trong cửa sổ lọc và sau đó chia tổng này cho số các phần tử của cửa sổ lọc.

Bộ lọc này được minh họa trong hình 1-2.



**Hình 1-2. Hình ảnh minh họa về phép lọc trung bình**

Các bước cơ bản của kỹ thuật lọc trung bình:

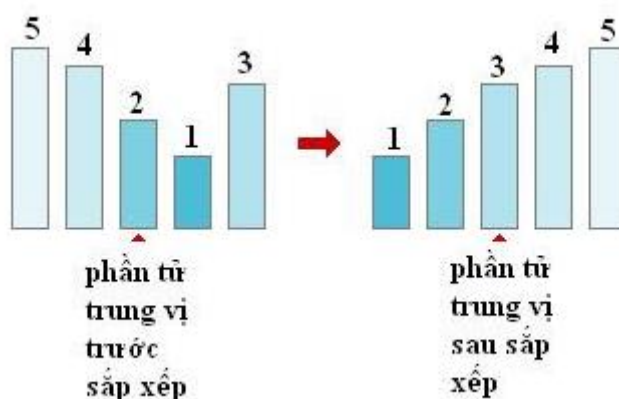
- Quét cửa sổ lọc lần lượt lên các thành phần của ảnh đầu vào; điền các giá trị được quét vào cửa sổ lọc.
- Xử lý bằng cách thao tác trên các thành phần của cửa sổ lọc.
- Tính giá trị trung bình các thành phần trong cửa sổ lọc.
- Gán giá trị trung bình này cho ảnh đầu ra.

#### Lọc trung vị

Lọc Trung vị là một kỹ thuật lọc phi tuyến, nó khá hiệu quả đối với hai loại nhiễu: nhiễu đốm (speckle noise) và nhiễu muối tiêu (salt-pepper noise). Kỹ thuật lọc trung vị này là một bước rất phổ biến trong xử lý ảnh.

Ý tưởng chính của thuật toán lọc Trung vị đó là sử dụng một cửa sổ lọc (ma trận 3x3) quét qua lần lượt từng điểm ảnh của ảnh đầu vào input. Tại vị trí mỗi điểm ảnh lấy giá trị của các điểm ảnh tương ứng trong vùng 3x3 của ảnh gốc "lấp" vào ma trận lọc. Sau đó sắp xếp các điểm ảnh trong cửa sổ này theo thứ tự (tăng dần hoặc giảm dần tùy ý). Cuối cùng, gán điểm ảnh nằm chính giữa (trung vị) của dãy giá trị điểm ảnh đã được sắp xếp ở trên cho giá trị điểm ảnh đang xét của ảnh đầu ra output.

Bộ lọc này được minh họa trong hình 1-3.



**Hình 1-3. Hình ảnh minh họa về phép lọc trung vị**

Các bước cơ bản của kỹ thuật lọc trung vị

- Quét cửa sổ lọc lên các thành phần của ảnh gốc; điền các giá trị được quét vào cửa sổ lọc.
- Lấy các thành phần trong cửa sổ lọc để xử lý.
- Sắp xếp theo thứ tự các thành phần trong cửa sổ lọc.
- Lưu lại thành phần trung vị, gán cho ảnh output

Kết quả của hai bộ lọc trung vị và lọc trung bình được thể hiện qua hình 1-4. Cả hai bộ lọc đều sử dụng cửa sổ lọc  $3 \times 3$ . Có thể thấy bộ lọc trung vị làm việc tốt hơn bộ lọc trung bình, nó làm mờ các cạnh của hình ảnh rất ít. Hình ảnh sau khi lọc cũng trở nên sắc nét hơn bộ lọc trung bình khá nhiều.



(a) Noiseless pepper image (256x256 pixels)



(b) Noisy image



(c) Mean filtered image



(d) Median filtered image

**Hình 1-4. Kết quả lọc ảnh theo 2 phương pháp trung bình và trung vị**

- (a) Ảnh gốc-không bị nhiễu
- (b) Ảnh gốc sau khi có thêm nhiễu
- (c) Kết quả lọc trung bình
- (d) Kết quả lọc trung vị



### 1.3 Kỹ thuật tìm biên trong xử lý ảnh

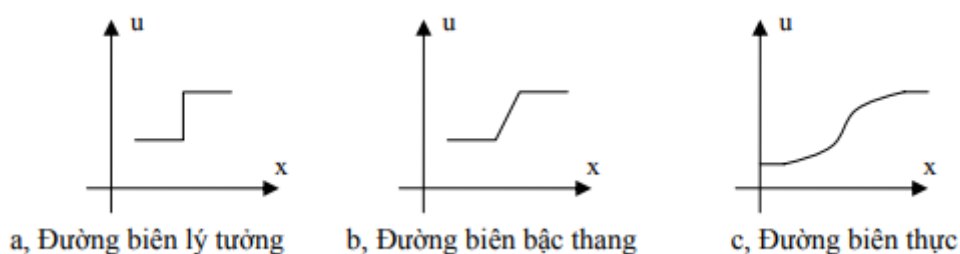
#### 1.3.1 Một số khái niệm

**Điểm biên:** Một điểm ảnh được coi là điểm biên nếu có sự thay đổi nhanh hoặc đột ngột về mức xám (hoặc màu). Ví dụ trong ảnh nhị phân, điểm đen gọi là điểm biên nếu lân cận nó có ít nhất một điểm trắng.

**Đường biên:** là tập hợp các điểm biên liên tiếp tạo thành một đường biên.

Ý nghĩa của đường biên trong xử lý: Thứ nhất, đường biên là một loại đặc trưng cục bộ tiêu biểu trong phân tích, nhận dạng ảnh. Thứ hai, người ta sử dụng biên làm phân cách các vùng xám (màu) cách biệt. Ngược lại, người ta cũng sử dụng các vùng ảnh để tìm đường phân cách.

Đường biên là nơi mà các điểm ảnh lân cận nhau có cường độ thay đổi mạnh một cách đột ngột. Một số kiểu đường biên hay gặp trên thực tế được minh họa trên hình 1-5.



**Hình 1-5. Một số kiểu đường biên thông dụng**

#### 1.3.2 Các phương pháp phát hiện biên

Các phương pháp phát hiện biên truyền thống thường dựa trên kết quả của phép nhân chập giữa bức ảnh cần nghiên cứu  $f(x, y)$  và một bộ lọc 2D  $h(x, y)$  thường được gọi là mặt nạ (mask). Công thức phát hiện biên được trình bày như bên dưới:

$$h(x, y) * f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(k_1, k_2) f(x - k_1, y - k_2) dk_1 dk_2 \quad (1.3)$$

Cấu trúc và giá trị của các toán tử phát hiện biên sẽ xác định hướng đặc trưng mà toán tử nhạy cảm với biên. Có một số toán tử thích hợp cho các đường biên có hướng nằm ngang, một số toán tử lại thích hợp cho việc tìm kiếm biên dạng thẳng đứng hay theo hướng đường chéo. Có nhiều phương pháp phát hiện biên đang được áp dụng, tuy nhiên ta có thể phân thành hai nhóm cơ bản là phát hiện biên dùng Gradient và phương pháp Laplacian.

### Phương pháp Gradient.

Phương pháp phát hiện biên dùng Gradient (sử dụng các toán tử Roberts, Prewitt, Sobel, Canny) dựa vào tính giá trị cực đại và cực tiểu của đạo hàm bậc nhất của ảnh.

Đạo hàm bậc nhất theo hướng ngang và dọc được tính theo công thức (1. 4)

$$\Delta f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1.4)$$

Biên độ của vector gradient hay độ lớn tổng cộng của giá trị đạo hàm nằm tại biên là kết hợp của cả hai giá trị này theo công thức (1. 5)

$$|\Delta f| = \sqrt{G_x^2 + G_y^2} \quad (1.5)$$

Hướng của vector gradient được xác định theo công thức (1. 6)

$$\nabla f = \tan^{-1} \theta \left( \frac{G_x}{G_y} \right) \quad (1.6)$$

Hướng của biên sẽ vuông góc với hướng của vector gradient này.

- Toán tử Sobel

Trên thực tế Sobel sử dụng hai mặt nạ có kích thước [3 x 3] trong đó một mặt nạ chỉ đơn giản là sự quay của mặt nạ kia đi một góc 90° như ở hình 1-6. Các mặt nạ này được thiết kế để tìm ra các đường biên theo chiều đứng và chiều ngang một cách tốt nhất. Khi thực hiện phép tích chập giữa ảnh và

các mặt nạ này ta nhận được các gradient theo chiều đứng và chiều ngang  $G_x$ ,  $G_y$ . Toán tử Sobel có dạng như hình 1-6.

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

**Hình 1-6. Toán tử Sobel**

- Toán tử Prewitt

Phương pháp Prewitt gần giống với Sobel. Đây là phương pháp lâu đời nhất, cổ điển nhất. Toán tử Prewitt được mô tả trên hình 1-7.

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

**Hình 1-7. Toán tử Prewitt**

- Toán tử Robert

Tương tự như Sobel, ta tính đường biên ngang và dọc một cách riêng rẽ dùng 2 mặt nạ như hình 1-8, sau đó tổng hợp lại để cho đường biên thực của ảnh.

0	0	0
0	-1	0
0	0	1

0	0	0
0	0	-1
0	1	0

**Hình 1-8. Toán tử Robert**

## 1.4 Biến đổi khoảng cách

### 1.4.1 Giới thiệu

Biến đổi khoảng cách (Distance tranform) là thuật toán biến đổi giá trị của điểm ảnh thông qua khoảng cách của nó so với điểm biên trong bức ảnh. Thông thường một bức ảnh sẽ được chia thành các thành phần như sau:

- Điểm đối tượng là những điểm trực tiếp cấu thành lên đối tượng. Trong ảnh nhị phân, điểm đối tượng là những điểm có mức xám bằng 1.
- Điểm nằm ngoài đối tượng, hay còn gọi là điểm nền, trong ảnh nhị phân, điểm nền là những điểm có giá trị mức xám bằng 0
- Giao của các điểm đối tượng và các điểm nền được gọi là biên ảnh viết tắt là B.

### 1.4.2 Thuật toán biến đổi khoảng cách đơn giản.

Thuật toán biến đổi khoảng cách đơn giản được trình bày như sau:

Đầu tiên, chỉ định mỗi phần tử trên đường biên một giá trị khoảng cách là 0,  $I(s) = 0$

Trong đó điểm “s” là những điểm nằm trong B (B là đường biên) . Sau đó với mỗi điểm “t” không nằm trong B thì chúng ta gán  $I(t) = \min\{d(s, t)\}$ , trong đó  $d(s, t)$  là khoảng cách Euclid từ điểm s đến điểm t.

Giải thuật của thuật toán này được trình bày bên dưới:

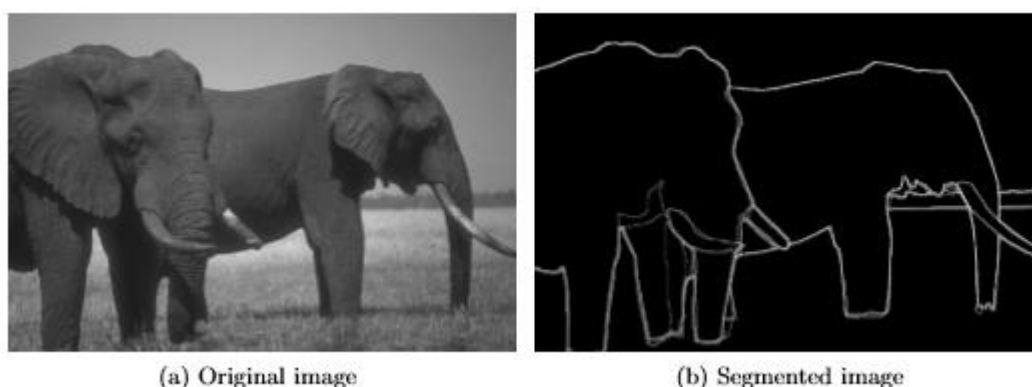
```
for (y = 1; y <ySize-1; y++) {
    for (x = 1; x <xSize-1; x++) {
        if (I'(x, y) !=0) {
            for (y1 = 0; y1 <ySize; y1++) {
                for (x1 = 0; x1 <xSize; x1++) {
                    if (I'(x1, y1) == 0) {
                        d = sqrt((x-x1)*(x-x1) + (y-y1)*(y-y1));
                        if (d < I'(x, y)) {
                            I'(x, y) = d;
                        }
                    }
                }
            }
        }
    }
}
```

```
        } // end for x1  
    } // end for y1  
} // end if  
} // end for x  
} // end for y
```

## 1.5 Phân đoạn ảnh

### 1.5.1 Giới thiệu

Phân đoạn ảnh (hay còn gọi là phân vùng ảnh) là bước then chốt trong xử lý ảnh. 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 mức xám, cùng màu hay cùng độ nhám.



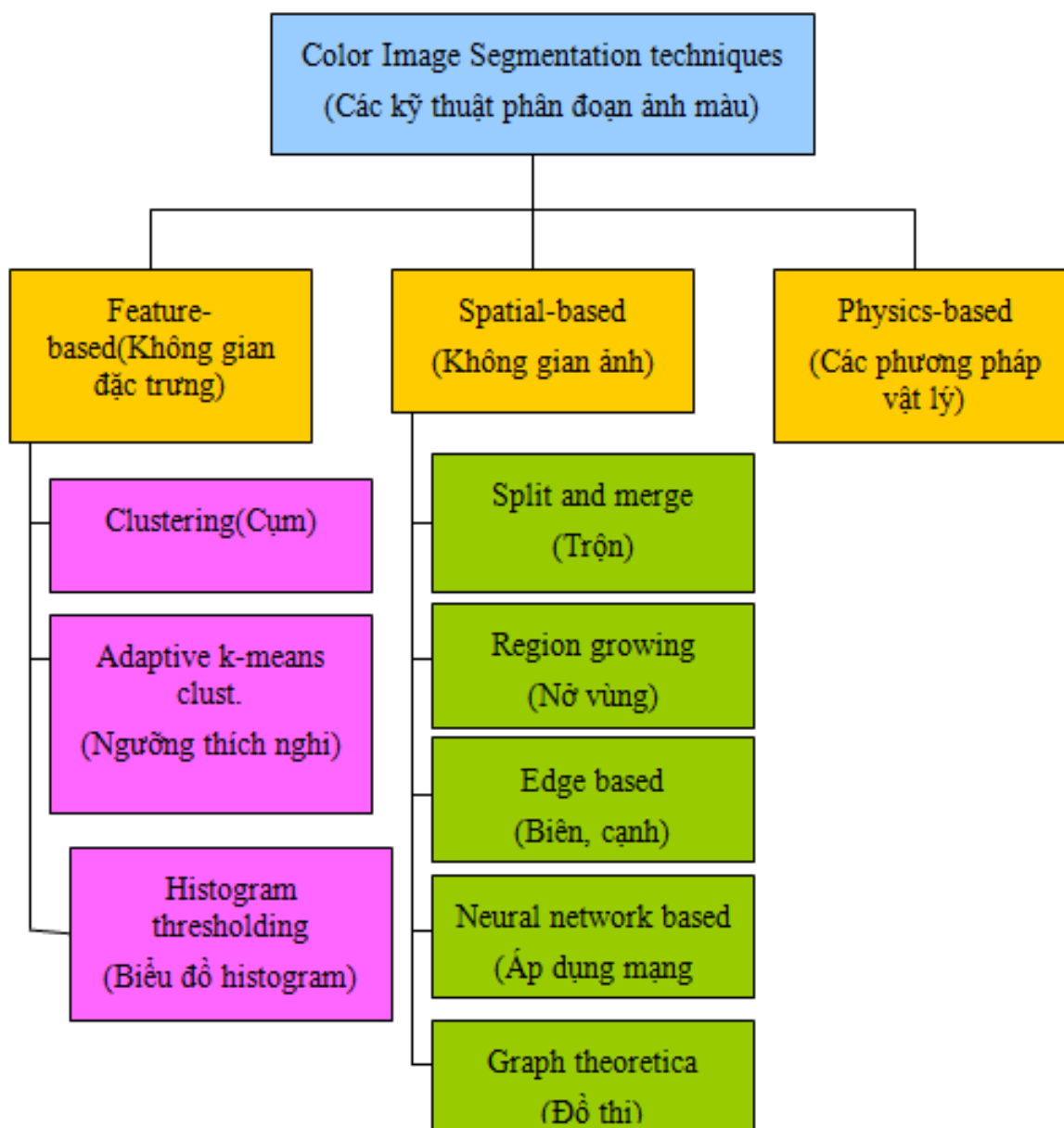
**Hình 1-9. Ví dụ về phân đoạn ảnh**

(a) Ảnh gốc trước khi phân đoạn

(b) Ảnh sau khi phân đoạn

Vùng ảnh là một chi tiết, một thực thể trong toàn cảnh. Nó là một tập hợp các điểm có cùng hoặc gần cùng một tính chất nào đó : mức xám, mức màu, độ nhám. Vùng ảnh là một trong những thuộc tính cơ bản của ảnh. Nói đến vùng ảnh là nói đến tính chất bề mặt. Đường bao quanh một vùng ảnh (Boundary) là biên ảnh. Các điểm trong một vùng ảnh có độ biến thiên giá trị mức xám tương đối đồng đều hay tính kết cấu tương đồng.

### 1.5.2 Một số hướng tiếp cận trong phân đoạn ảnh



**Hình 1-10. Một số hướng tiếp cận 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. Tổng quan về một số hướng tiếp cận trong

phân đoạn ảnh được mô tả như trong hình 1-10. Các ưu nhược điểm của các phương pháp phân đoạn ảnh được minh họa trong bảng 1-1

**Bảng 1-1. Ưu nhược điểm của các phương pháp phân đoạn ảnh**

Phương pháp phân vùng	Ưu điểm	Khuyết điểm
Featured-based techniques (Các phương pháp dựa trên không gian đặc trưng)		
Clustering (Phân đoạn dựa trên cụm)	<ul style="list-style-type: none"> <li>▪ Phân loại không giám sát.</li> <li>▪ Tồn tại các phương pháp kinh nghiệm cải tiến (heuristic) và hữu hạn.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Không quan tâm đến các thông tin trong không gian ảnh.</li> <li>▪ Có vấn đề trong việc xác định số lượng các cụm ban đầu.</li> <li>▪ Khó khăn trong việc điều chỉnh các cụm sao cho phù hợp với các vùng trong ảnh.</li> </ul>
Adaptive Clustering (Phân đoạn dựa vào ngưỡng thích nghi)	<ul style="list-style-type: none"> <li>▪ Sở hữu tính liên tục trong không gian ảnh và tính thích nghi cục bộ đối với các vùng ảnh.</li> <li>▪ Sử dụng các ràng buộc về không gian ảnh.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Cực đại hoá một xác suất hậu nghiệm có thể bị sai do các cực trị địa phương.</li> <li>▪ Hội tụ chậm.</li> </ul>
Histogram thresholding (Phân đoạn dựa vào biểu đồ histogram)	<ul style="list-style-type: none"> <li>▪ Không cần biết trước bất kỳ thông tin nào từ ảnh.</li> <li>▪ Các giải thuật nhanh</li> </ul>	<ul style="list-style-type: none"> <li>▪ Bỏ qua các thông tin về không gian ảnh.</li> <li>▪ Lấy ngưỡng trong các histogram đa chiều là</li> </ul>

	và dễ dàng cài đặt.	<p>một quá trình phức tạp.</p> <ul style="list-style-type: none"> <li>▪ Dễ ảnh hưởng bởi nhiễu xuất hiện trong ảnh.</li> </ul>
Spatial-based techniques (Các phương pháp trên không gian ảnh)		
<p>Spit and Merge (Chia và hợp vùng)</p>	<ul style="list-style-type: none"> <li>▪ Sử dụng các thông tin về không gian ảnh là chính.</li> <li>▪ Cho kết quả tốt với các ảnh chứa nhiều vùng màu đồng nhất.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Định nghĩa mức độ đồng nhất về màu sắc có thể phức tạp và khó khăn.</li> <li>▪ Quadtree có thể gây ra các kết quả không như mong muốn.</li> </ul>
<p>Region growing (Thuật toán nở vùng)</p>	<ul style="list-style-type: none"> <li>▪ Các vùng ảnh đồng nhất và liên thông.</li> <li>▪ Có một số thuật giải có tốc độ thực thi khá nhanh.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Tốn kém chi phí sử dụng bộ nhớ và tính toán.</li> <li>▪ Gặp khó khăn trong việc thu thập tập các điểm mầm và xác định các điều kiện đồng nhất đầy đủ.</li> <li>▪ Chịu ảnh hưởng bởi các đặc tính tự nhiên của kỹ thuật này.</li> </ul>
<p>Graph theories (Đồ thị)</p>	<ul style="list-style-type: none"> <li>▪ Thể hiện tốt không gian ảnh bằng đồ thị.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Một vài thuật giải mất khá nhiều thời gian</li> </ul>



	<ul style="list-style-type: none"> <li>▪ Một số thuật toán có tốc độ thực hiện nhanh.</li> </ul>	<p>thực hiện.</p> <ul style="list-style-type: none"> <li>▪ Các đặc trưng cục bộ đôi khi được sử dụng nhiều hơn các đặc trưng toàn cục.</li> </ul>
Neural networks (Mạng Neural)	<ul style="list-style-type: none"> <li>▪ Mức độ song song hoá cao và có tốc độ thực thi nhanh.</li> <li>▪ Khả năng chống chịu tốt trước các thay đổi.</li> <li>▪ Một công cụ hữu hiệu cho các ứng dụng nhận dạng và xử lý ảnh y khoa.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Màu sắc có thể làm tăng độ phức tạp của mạng.</li> <li>▪ Quá trình học cần phải biết trước số lượng các phân lớp/cục.</li> </ul>
Edge-based (Tìm biên)	<ul style="list-style-type: none"> <li>▪ Là phương pháp được hỗ trợ mạnh bởi các toán tử dò biên.</li> <li>▪ Có hiệu năng tốt với các ứng dụng dò biên đối tượng theo đường cong.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Khó khăn trong việc định nghĩa một hàm gradient cho các ảnh màu.</li> <li>▪ Nhiều hoặc các ảnh có độ tương phản kém ảnh hưởng xấu đến kết quả phân vùng.</li> </ul>
Physics-based techniques (Các phương pháp vật lý)		
	<ul style="list-style-type: none"> <li>▪ Khẳng định tính chắc chắn đối với các vùng bóng sáng/tối, và vùng bóng chuyển tiếp</li> </ul>	<ul style="list-style-type: none"> <li>▪ Bị giới hạn vào một số lượng nhất định các loại vật chất hình</li> </ul>

	<p>(diffuse hoặc shade)</p> <ul style="list-style-type: none"> <li>▪ Phân vùng các đối tượng dựa vào thành phần vật liệu cấu tạo</li> </ul>	<p>thành nên đối tượng.</p> <ul style="list-style-type: none"> <li>▪ Khó khăn trong việc xác định vùng bóng sáng và bóng chuyển tiếp trong các ảnh thực.</li> <li>▪ Một vài giải thuật đòi hỏi các thông tin về hình dạng đối tượng (không luôn luôn đáp ứng được) .</li> <li>▪ Chi phí tính toán khá cao.</li> </ul>
--	---	---

### 1.5.3 Phân đoạn dựa trên ngưỡng

Phương pháp phân đoạn dựa trên ngưỡng mục đích để phân chia hình ảnh dựa trên phân vùng ảnh thành các vùng khác nhau tùy vào giá trị cường độ của các điểm ảnh và các điểm ảnh lân cận. Đây là thao tác đơn giản nhất để phân đoạn hình ảnh.

Nguyên lý cơ bản của kỹ thuật phân ngưỡng đó là chọn một giá trị tối ưu, phù hợp để chia các điểm ảnh trong hình ảnh thành các lớp khác nhau và phân biệt được đâu là đối tượng, đâu là nền.

Giá trị ngưỡng trong phân đoạn dựa trên ngưỡng được người dùng đặt phù hợp. Bất kỳ điểm nào trên hình ảnh mà có giá trị pixel lớn hơn giá trị ngưỡng này thì được gọi là điểm đối tượng và ngược lại ta gọi đó là điểm nền.

Có 3 kỹ thuật chọn ngưỡng chính được gọi là:

- Ngưỡng toàn cục
- Ngưỡng cục bộ
- Ngưỡng động.

Ở ngưỡng toàn cục, giá trị ngưỡng chỉ phụ thuộc vào giá trị mức xám của pixel của toàn vùng mà ta đang xét tới.

Ở ngưỡng cục bộ, ngưỡng được chọn phụ thuộc vào giá trị độ xám của các pixel thuộc một vùng cục bộ đang xét, các thuộc tính cục bộ của pixel như giá trị xám trung bình của một vùng đang xét.

Giá trị ngưỡng ở ngưỡng động được tính bằng các thuộc tính cục bộ, cường độ điểm ảnh và không gian tọa độ. Ngưỡng động thay đổi giá trị ngưỡng chủ động trên hình ảnh. Trong từng vùng ảnh, giá trị ngưỡng sẽ thay đổi sao cho phù hợp với các điều kiện của vùng ảnh đó. Giá trị ngưỡng được tính cho mỗi pixel dựa trên mối quan hệ của các điểm ảnh cục bộ. Ý nghĩa giá trị cường độ của mỗi điểm ảnh lân cận là rất quan trọng để xác định ngưỡng cho vùng.

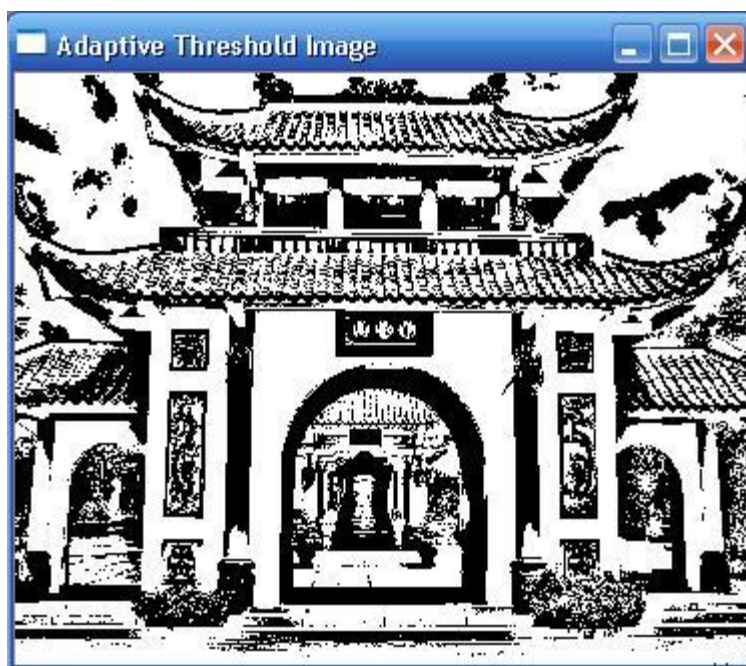
Dưới đây là 1 một số ví dụ về phân đoạn ảnh dựa trên ngưỡng:



**Hình 1-11. Ví dụ ảnh gốc trước khi áp dụng phân ngưỡng**



**Hình 1-12.** Ảnh sau khi phân ngưỡng toàn cục với mức ngưỡng 150



**Hình 1-13.** Ảnh phân ngưỡng với ngưỡng động

Ngoài ra chúng ta còn một số kỹ thuật phân đoạn ảnh dựa trên nhiều mảng khác nhau, các kỹ thuật này đều được nói khái quát ở bảng trên

## CHƯƠNG 2: PHÂN ĐOẠN ẢNH VỚI THUẬT TOÁN NỞ VÙNG

### 2.1 Thuật toán nở vùng

Thuật toán nở vùng (region growing) là một thuật toán phân đoạn ảnh được sử dụng để phân chia các vùng khác nhau trên một ảnh. Đặc điểm của các thuật toán nở vùng phụ thuộc vào thông số để kết thúc quá trình tìm kiếm trong vùng. Thông thường, quá trình nở vùng sẽ được dừng khi không có pixel thỏa mãn tiêu chuẩn của vùng đó. Khi thông tin ban đầu không thể tìm kiếm được, quá trình nở vùng sẽ dựa vào những pixel có cùng đặc tính để quyết định xem pixel có nằm trong vùng cần tính hay không. Việc lựa chọn các tiêu chuẩn tương đồng phụ thuộc đặc điểm của đối tượng cần xét trên ảnh và loại dữ liệu ảnh. Tiêu chuẩn có thể bao gồm giá trị cường độ xám, đặc điểm cấu trúc hoặc chỉ số thống kê và không tiến hành tính toán lại các pixel đã tính trong vùng. Việc lựa chọn tiêu chuẩn chính xác sẽ làm tăng khả năng xác định của thuật toán nở vùng cả về kích thước của vùng xét và hình dạng của vùng.

### 2.2 Một số thuật toán nở vùng

#### 2.2.1 Thuật toán nở vùng cơ bản

Thuật toán nở vùng được mô tả như một quá trình nhóm các pixel hoặc các vùng nhỏ vào một vùng lớn hơn dựa trên các tiêu chuẩn đã định trước. Thuật toán cơ bản là bắt đầu từ các điểm gieo mầm và từ đó mở rộng vùng tìm kiếm phụ thuộc vào các điểm lân cận có cùng đặc điểm với điểm gieo mầm như cùng mức độ xám. Công thức (2. 1) mô tả thuật toán nở vùng dựa vào giá trị độ xám của các điểm gieo mầm và của pixel đang xét. Xét công thức (2. 1) thì pixel  $R_i$  sẽ được chọn vào vùng P nếu thỏa mãn điều kiện là giá trị tuyệt đối của hiệu giá trị độ xám của pixel  $R_i$  và giá trị độ xám của điểm gieo mầm nhỏ hơn một ngưỡng T được lựa chọn. Ngưỡng T được người dùng chọn tùy theo mục đích.

$$P(R_i) = \text{True} : \text{if } |z - z_{seed}| < T \quad (2.1)$$

### 2.2.2 Thuật toán nở vùng thống kê

Thuật toán nở vùng thống kê về bản chất vẫn là thuật toán nở vùng theo nguyên tắc lan tỏa từ một điểm gieo mầm bên trong vùng. Quy tắc đặt ra cho việc lấy thêm một điểm vào vùng là dựa trên việc so sánh giá trị điểm ảnh mới với chỉ số thống kê được tính từ các điểm đã được phân loại vào vùng đang xét [1]. Thuật toán nở vùng thống kê được thực hiện dựa trên giá trị trung bình của các điểm ảnh trong vùng dựa vào công thức (2. 2) và độ lệch chuẩn công thức (2. 3) [2].

$$M = \frac{1}{n} \sum_{(r,c) \in R(i)} I(r, c) \quad (2.2)$$

$$s.d = \sqrt{\frac{1}{n} \sum_{(r,c) \in R(i)} [I(r, c) - M]^2} \quad (2.3)$$

trong đó:  $I(r, c)$  giá trị cường độ xám của các pixel có trong vùng. Một pixel sẽ được phân loại vào vùng đang xét nếu giá trị cường độ xám của pixel đó gần với giá trị cường độ xám trung bình của vùng xét dựa vào công thức (2. 4) [2]

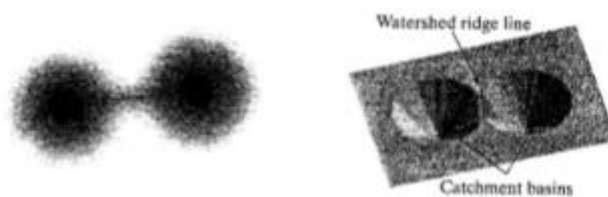
$$| I(r, c) - M(i) | \leq T(i) \quad (2.4)$$

Trong đó, giá trị ngưỡng  $T(i)$  của vùng xét có thể được xác định theo công thức (2. 5)

$$T(i) = 1 - \frac{s.d(i)}{M(i)} \quad (2.5)$$

### 2.2.3 Thuật toán nở vùng theo lưu vực

Trong tài liệu [1], tác giả sử dụng thuật toán nở vùng theo lưu vực để tách vết dầu trên ảnh SAR. Do các vết dầu là vùng đối tượng và tối hơn so với khu vực lân cận trên ảnh nên nếu coi ảnh SAR như hình ảnh 3D của địa hình thì các vết dầu có thể được coi như các thung lũng so với các vùng xung quanh. Chính vì thế các vết dầu có thể được tách dựa trên các quy luật hình thành lưu vực.



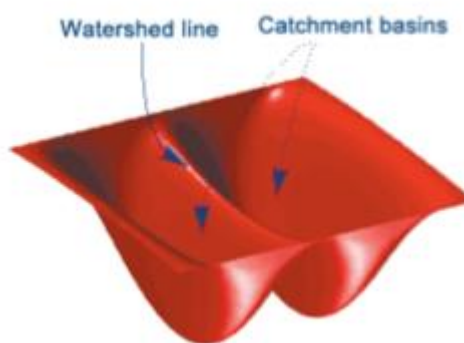
**Hình 2-1. Mô tả hình ảnh lưu vực của 2 vùng**

Như vậy, với một điểm gieo mầm được xác định trong lưu vực thì thuật toán nở vùng sẽ được xây dựng tương tự như hiện tượng nước sẽ được dâng lên và làm đầy lưu vực. Thuật toán tìm kiếm các điểm thuộc vùng xét vẫn tương tự như thuật toán nở vùng cơ bản. Tuy nhiên điều kiện để lựa chọn điểm pixel thuộc vào vùng xét sẽ dựa trên sự biến thiên giá trị của bản thân pixel. Trong thuật toán nở vùng theo lưu vực có rất nhiều phương pháp lựa chọn điều kiện để lựa chọn điểm pixel thuộc vào vùng xét như xác định các đường phân thủy (watershed ridge line) giữa các vùng; xác định giá trị gradient của các pixel có cường độ xám thấp ở đáy lưu vực và các pixel có cường độ xám cao ở đỉnh lưu vực.

## **2.3 Phân đoạn ảnh theo biến đổi Watershed**

### **2.3.1 Giới thiệu:**

Watershed là một dãy thuật toán, được sử dụng trong phân đoạn ảnh. Nó phổ biến trong các lĩnh vực như xử lý y sinh học và y tế, thị giác máy tính. Về địa lý, lưu vực có nghĩa là rặng núi phân chia các khu vực thoát nước bởi các hệ thống sông khác nhau. Nếu ảnh được xem là cảnh quan địa chất thì các đường phân thủy xác định ranh giới tách các vùng hình ảnh. Thuật toán watershed mục đích để tìm ra các lưu vực và các đường phân thủy, hay còn gọi là các đường phân cách các vùng trên ảnh, nhóm các vùng ảnh có cùng các thuộc tính liên quan lại thành một vùng đồng nhất.



**Hình 2-2. Hình ảnh minh họa thuật toán watershed**

Watershed line là đường phân thủy được giải thích như là đập để ngăn chặn sự hòa nhập lượng nước của các lưu vực chứa nước khác nhau, trong một bức ảnh thì nó được xem như đường phân cách các khu vực ảnh được nhóm vào thành một nhóm trước đó.

Catchment basins là các lưu vực chứa nước, áp dụng vào trong một bức ảnh thì nó được coi như là một vùng các pixel được nhóm lại thành một vùng nhất định có cùng một số thuộc tính nào đó (mức xám, độ sáng vv...)

Thuật toán watershed được phát triển theo 2 hướng:

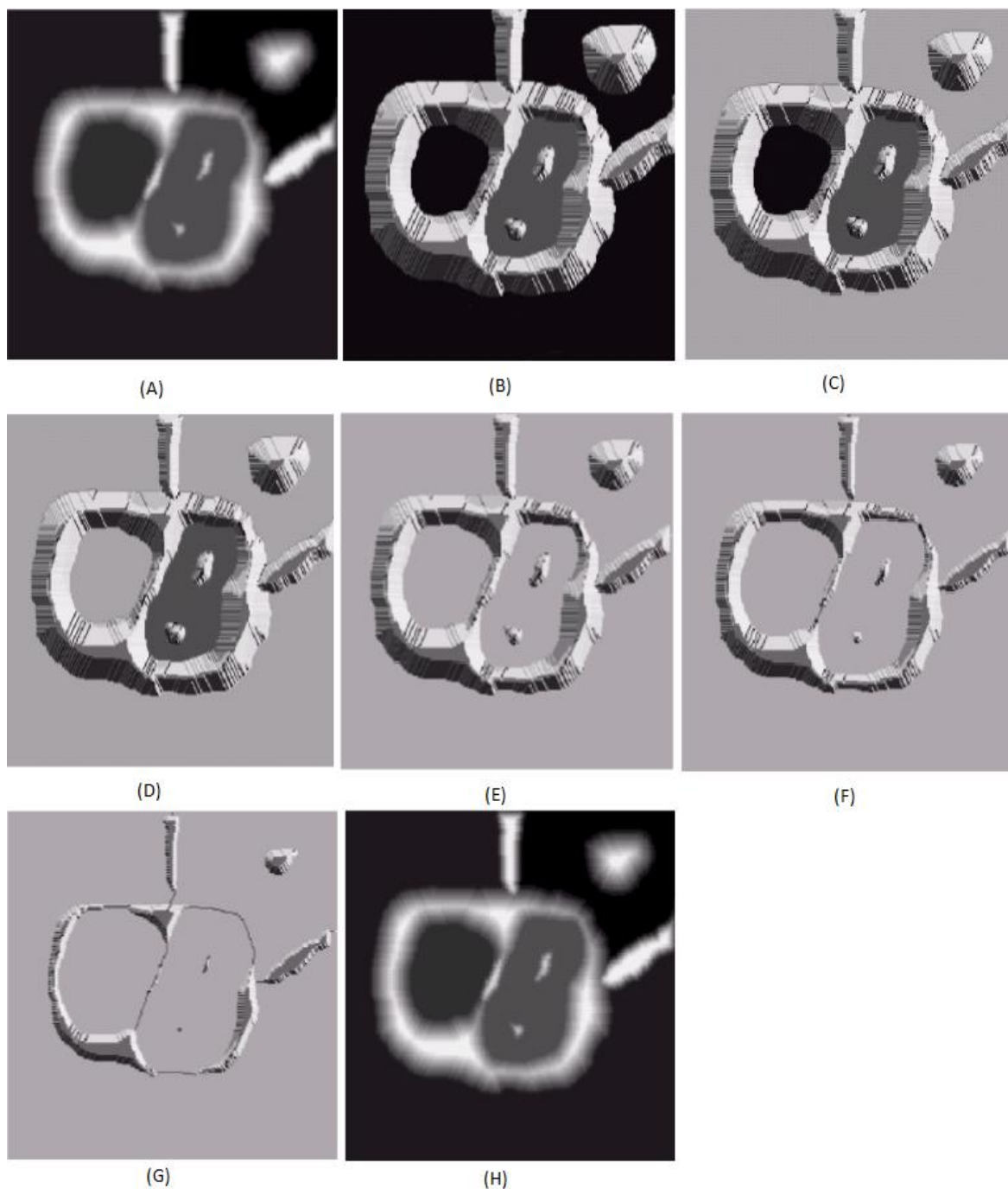
- Hướng đầu tiên được trình bày theo nguyên lý nước dâng và nó là một hướng tiếp cận truyền thống.
- Hướng thứ hai được trình bày theo nguyên lý dòng chảy.

**Thuật toán watershed dựa theo nguyên lý nước dâng:**

Ảnh đầu vào được coi như một vùng địa hình với các vùng trũng nhất là tập hợp những điểm có giá trị mức xám nhỏ nhất. Hãy tưởng tượng, nước được ngập bắt đầu từ những điểm này với tốc độ không đổi, mực nước sẽ tăng dần bề mặt, khi mực nước dâng lên cao, các lưu vực sẽ kết hợp với nhau. Để ngăn chặn sự sát nhập này, một đập được xây dựng đúng chỗ giao nhau của 2 lưu vực. Cứ thế, nước dâng sẽ đạt đến đỉnh điểm mà chỉ đầu đập mới có thể nhìn thấy trên đường nước. Những ranh giới đập liên tục này được gọi là các đường phân thủy.



Để hiểu thuật toán này rõ ràng hơn, những hình ảnh dưới đây sẽ mô tả chi tiết thuật toán nêu trên.



**Hình 2-3. Mô tả thuật toán watershed theo nguyên lý nước dâng**

Trong đó:

(A) Là ảnh gốc

(B) Là giao diện địa hình của ảnh

- (C) Là hình ảnh bắt đầu quá trình nước dâng
- (D) Là giai đoạn tiếp theo của quá trình, nước được dâng cao hơn
- (E) Tiếp tục việc nước dâng
- (F) Bắt đầu quá trình sát nhập nước từ các lưu vực, các đập chắn nước bắt đầu được hình thành
- (G) Đập chắn nước được hình thành dài hơn do việc sát nhập nước liên tục.
- (H) Lưu vực cuối cùng, các đường phân đoạn được hình thành

### **Thuật toán watershed dựa theo nguyên lý dòng chảy:**

Trong cách tiếp cận thuật toán watershed dựa theo nguyên lý dòng chảy, hình ảnh được coi như một bề mặt địa hình với ba loại điểm khác nhau:

- Những điểm chỉ ra cho ta biết đó là điểm tối thiểu
- Những điểm trên sườn dốc, đó là những điểm mà nước chảy vào tối thiểu có xác suất cao nhất.
- Điểm có nước chảy xuống vùng tối thiểu có xác suất cao hơn.

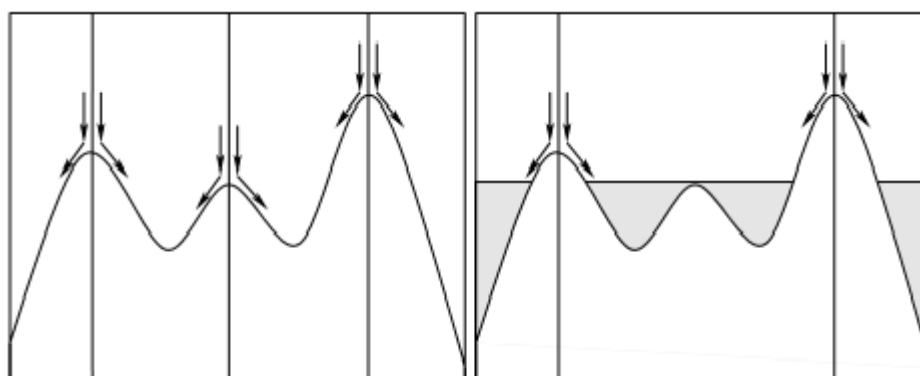
Các nhóm điểm thỏa mãn nhóm thứ hai sẽ được gọi là Catchment basin (tức là vùng chứa các điểm ảnh có chung một tính chất hay còn gọi là lưu vực) .

Các điểm ảnh thỏa mãn nhóm thứ ba, nơi mà nước rơi xuống và chảy vào nhiều vùng tối thiểu ta gọi đó là các điểm Watershed line (tức là tập hợp các điểm tạo ra đường ngăn cách sự hòa nhập nước, áp dụng vào phân đoạn ảnh ta có thể gọi nó là đường phân thủy).

Thuật toán watershed theo nguyên lý dòng chảy khai thác các khái niệm hơi khác một chút so với thuật toán phân đoạn theo nguyên lý nước dâng truyền thống. Về cơ bản, dòng chảy sẽ chảy từ trên đỉnh núi (bề mặt của định hình) và di chuyển xuống phía bên dưới do ảnh hưởng của trọng lực cho

đến khi đạt đến bề mặt tối thiểu của vùng địa hình. Thuật toán sẽ theo dõi con đường mà từ khi nước bắt đầu chảy qua các điểm trên bề mặt về phía vùng địa hình tối thiểu nếu như nước chảy qua điểm đó. Tất cả các điểm ảnh sẽ tạo thành một phân đoạn khi lượng nước chảy qua chúng cùng chảy về một vị trí sâu nhất. Khi một điểm mà nước chảy xuống có nhiều hơn một con đường để chảy về các điểm cực tiểu thì nó có thể được xác định cho một vùng trũng bất kỳ.

Ngưỡng chìm được sử dụng để loại bỏ ngọn núi thấp nhất (tương ứng với cạnh yếu nhất trong hình ảnh) . Ngọn núi sẽ không được xem xét nếu như chiều cao của chúng nằm dưới ngưỡng chìm này. Đường ngưỡng dưới này được minh họa như hình 2-4 phía bên phải khi lượng nước đã dâng cao.



**Hình 2-4. Thuật toán dòng chảy và ngưỡng chìm**

### 2.3.2 Thuật toán watershed dựa trên các thành phần liên thông

Khái niệm cơ bản của thuật toán watershed dựa trên các thành phần liên thông được giải thích như hình 2-5, 2-6 và 2-7. Tư tưởng của thuật toán watershed dựa trên các thành phần liên thông tương tự như thuật toán dựa trên nguyên lý dòng chảy.

Ma trận điểm ảnh ban đầu  $F(5, 6)$  có ba giá trị tối thiểu cục bộ được biểu thị bằng các hộp màu xám được thể hiện qua hình 2-5.

Các thành phần (pixel) nếu không phải là giá trị tối thiểu cục bộ thì nó được kết nối với điểm có giá trị tối thiểu cục bộ như được chỉ ra bởi các mũi tên như hình 2-6.

Tất cả các pixel không phải là giá trị tối thiểu kết nối tới một điểm có giá trị tối thiểu sẽ làm thành một phân đoạn và sẽ được gán một nhãn cho tất cả các điểm đó như hình 2-7.

7	4	8	12	11	3
7	7	8	12	11	7
13	13	15	16	16	13
19	19	18	17	15	7
20	18	17	16	15	5

Hình 2-5. Ảnh gốc

→	m	←	←	→	m
↗	↑	↖	←	↗	↑
↑	↑	↖	↖	↗	↑
↑	↑	↑	→	↘	↓
→	→	→	→	→	m

Hình 2-6. Các pixel lân cận liên kết tới điểm cực tiểu

0	0	0	0	1	1
0	0	0	0	1	1
0	0	0	0	1	1
0	0	0	2	2	2
2	2	2	2	2	2

Hình 2-7. Nhãn được gán cho các điểm ảnh

**Chi tiết của thuật toán:**

Các ký hiệu được sử dụng trong thuật toán:

- P: biểu diễn một điểm ảnh,  $f(p)$  : là giá trị xám của điểm ảnh p.
- N: là điểm ảnh láng giềng của P,  $f(n)$  : là giá trị xám của điểm ảnh N.
- $V(p)$  là giá trị dùng để lưu khoảng cách từ thấp nhất đến cao nhất của điểm ảnh.
- $L[p]$  là mảng được sử dụng để lưu trữ các nhãn.
- LMAX và VMAX biểu thị giá trị tối đa cho nhãn và khoảng cách tối đa trong hệ thống tương ứng. VMAX xác định khoảng cách giữa pixel đầu tiên của hàng đầu tiên đến pixel cuối cùng của hàng cuối cùng.

Thuật toán watershed dựa trên các thành phần liên thông bao gồm 3 bước như sau:

- Bước 1: Tìm ra các điểm có giá trị độ xám nhỏ hơn giá trị độ xám của các điểm láng giềng.
- Bước 2: Tìm ra khoảng cách của các điểm ảnh nằm trên vùng bằng phẳng đến điểm ảnh có giá trị xám thấp nhất và gán giá trị cho nó.
- Bước 3: Gán nhãn cho các điểm ảnh có cùng thuộc tính và nhóm nó thành một vùng.

Sau đây là thuật toán chi tiết của các bước:

**Bước 1:**

Ban đầu, mảng  $v[p]$  được đặt là “0” cho tất cả các phần tử trong mảng. Ảnh đầu vào sẽ được quét từ phía trên cùng từ trái qua phải theo từng hàng. Giá trị  $v[p]$  sẽ được đặt là ‘0’ (không đổi) nếu giá trị độ xám của điểm ảnh đang xét thấp hơn hoặc bằng giá trị độ xám của các điểm ảnh lân cận. Các điểm ảnh có giá trị độ xám cao hơn giá trị độ xám của các điểm ảnh lân cận sẽ được đặt là ‘1’. Thuật toán bước 1 được trình bày bằng hình ảnh 2-8 và hình 2-9 phía dưới.

3	5	5	2	8	8	8	11	10	10
5	5	11	11	8	11	11	8	10	10
11	5	11	11	9	9	9	9	8	10
11	11	11	7	7	7	7	9	9	8
11	11	11	11	11	9	7	10	8	10
11	10	11	9	7	7	9	9	10	8
11	10	11	9	11	9	10	10	8	10
11	11	11	8	8	8	8	8	10	10
11	11	11	11	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10
11	11	11	11	10	10	10	10	10	10

**Hình 2-8. Giá trị mức xám của ảnh đầu vào**

0	1	1	0	1	0	0	1	1	0
1	1	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	0	1
1	1	1	0	0	0	0	1	1	0
1	1	1	1	1	1	0	1	0	1
1	0	1	1	0	0	1	1	1	0
1	0	1	1	1	1	1	1	0	1
1	1	1	0	0	0	0	0	1	1
1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0

**Hình 2-9. Giá trị  $v(p)$  của từng điểm ảnh sau khi chạy bước 1**

Bước này được mô tả theo thuật toán sau:

```

function step1 (p)
    if  $v[p] \neq 1$  then
        for each n of p // n is neighbour pixel of p
            if  $f[n] < f(p)$  then  $v[p] \leftarrow 1$ 
            end if
        end for
    end if
end function
    
```

Thuật toán của bước 1 khá đơn giản, nó sẽ tiến hành thay đổi giá trị  $v(p)$  của điểm ảnh đang xét bằng cách so sánh giá trị độ xám của điểm ảnh đang xét với các điểm ảnh lân cận, giá trị  $v(p)$  của điểm ảnh đang xét sẽ được đặt là 1 nếu như nó thỏa mãn điều kiện là giá trị độ xám  $f(p)$  của nó lớn hơn một trong những giá trị độ xám của các điểm ảnh lân cận  $f(n)$ . Ngược lại thì giá trị  $v(p)$  của điểm ảnh đang xét sẽ được đặt là 0 (không đổi).

**Bước 2:**

**Vùng bằng phẳng** là vùng chứa ít nhất 2 điểm ảnh có giá trị độ xám  $f(p)$  giống nhau là điểm lân cận nằm cạnh nhau.

Xét ví dụ trong hình 2-8, Các điểm ảnh  $I(1;5)$ ,  $I(1;6)$ ,  $I(1;7)$  lần lượt có giá trị độ xám  $f(p) = 8, 8, 8$  lên nó được coi là một vùng bằng phẳng, được đánh dấu bằng màu xám nhạt như trong hình 2-8.

Trong bước này, chỉ có các điểm ảnh có giá trị  $v[p]=0$  tìm được ở bước 1 sẽ được xem xét, các điểm ảnh lân cận của điểm ảnh đang được xét phải nằm trên cùng một mặt phẳng giá trị không đổi (Vùng bằng phẳng) . Thuật toán duyệt từ trái qua phải theo từng hàng, từ trên xuống dưới.

Bước này được mô tả bởi thuật toán sau:

```

function step2 (p)
    if  $v[p] \neq 1$  then  $min \leftarrow VMAX$ ,
        for each n of p // n is neighbor pixel of p
            if  $f(n) = f(p)$  and  $v[n] > 0$  and  $v[n] < min$  then  $min \leftarrow v[n]$ 
            end if
        end for
        if  $min \neq VMAX$  and  $v[p] \neq (min+1)$  then  $v[p] \leftarrow min+1$ 
        end if
    end if
end function
    
```

Thuật toán bước 2 được minh họa bằng các hình ảnh bên dưới:

3	5	5	2	8	8	8	11	10	10
5	5	11	11	8	11	11	8	10	10
11	5	11	11	9	9	9	9	8	10
11	11	11	7	7	7	7	9	9	8
11	11	11	11	11	9	7	10	8	10
11	10	11	9	7	7	9	9	10	8
11	10	11	9	11	9	10	10	8	10
11	11	11	8	8	8	8	8	10	10
11	11	11	11	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10
11	11	11	11	10	10	10	10	10	10

Hình 2-10. Giá trị xám của ảnh đầu vào

0	1	1	0	1	0	0	1	1	0
1	1	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	0	1
1	1	1	0	0	0	0	1	1	0
1	1	1	1	1	1	0	1	0	1
1	0	1	1	0	0	1	1	1	0
1	0	1	1	1	1	1	1	0	1
1	1	1	0	0	0	0	0	1	1
1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0

Hình 2-11. Giá trị  $v(p)$  của ảnh sau khi chạy bước 1

0	1	1	0	1	2	3	1	1	2
1	1	1	1	1	1	1	4	1	1
1	2	1	1	1	1	1	1	5	1
1	1	1	0	0	0	0	1	1	6
1	1	1	1	1	1	0	1	7	1
1	0	1	1	0	0	1	1	1	8
1	0	1	1	1	1	1	1	9	1
1	1	1	0	0	0	0	10	1	1
1	1	1	1	1	1	1	1	1	2
0	0	0	2	2	2	2	2	2	2
1	1	1	1	3	3	3	3	3	3

Hình 2-12. Giá trị  $v(p)$  của các điểm ảnh sau khi chạy bước 2

0	1	1	0	1	2	3	1	1	2
1	1	1	1	1	1	1	4	1	1
1	2	1	1	1	1	1	1	5	1
1	1	1	0	0	0	0	1	1	6
1	1	1	1	1	1	0	1	7	1
1	0	1	1	0	0	1	1	1	8
1	0	1	1	1	1	1	1	9	1
1	1	1	14	13	12	11	10	1	1
1	1	1	1	1	1	1	1	1	2
5	4	3	2	2	2	2	2	2	2
1	1	1	1	3	3	3	3	3	3

Hình 2-13. Giá trị  $v(p)$  sau khi đã hoàn tất các bước quét ảnh

**Bước 3:**

Bước này được mô tả như sau:

**function** step3 (p)

lmin  $\leftarrow$  LMAX, fmin  $\leftarrow$  f(p)



```

if  $v[p] = 0$  then
    for each n of p
        if  $f(n) = f(p)$  and  $l[n] > 0$  and  $l[n] < l_{min}$  then  $l_{min} \leftarrow l[n]$ 
        end if
    end for
    if  $l_{min} = LMAX$  and  $l[p] = 0$  then  $l_{min} \leftarrow \text{New label} + 1$ 
    end if
else if  $v[p] = 1$  then
    for each n of p
        if  $f(n) < f_{min}$  then  $f_{min} \leftarrow f[n]$ 
        end if
    for each n of p
        if  $f(n) = f_{min}$  and  $l[n] > 0$  and  $l[n] < l_{min}$  then
             $l_{min} \leftarrow l[n]$ 
        end if
    else
        for each n of p
            if  $f(n) = f(p)$  and  $v[n] = v[p] - 1$  and  $l[n] > 0$  and  $l[n] < l_{min}$  then
                 $l_{min} \leftarrow l[n]$ 
            end if
        end if
        if  $l_{min} \neq LMAX$  and  $l(p) \neq l_{min}$  then  $l[p] \leftarrow l_{min}$ 
        end if
end function

```

**Ví dụ:**

Trong bước này, ảnh sẽ được quét từ trái trên cùng xuống dưới phải.

Ban đầu nhãn  $L(p)$  của tất cả các điểm ảnh đều được gán  $= 0$ .

Thuật toán sẽ duyệt ảnh đến khi tất cả các điểm ảnh đều được gán nhãn mới và duyệt đến khi các bước duyệt không có gì thay đổi thì thuật toán dừng lại.

Một số kết quả ảnh sau khi thực hiện bước 3:

3	5	5	2	8	8	8	11	10	10
5	5	11	11	8	11	11	8	10	10
11	5	11	11	9	9	9	9	8	10
11	11	11	7	7	7	7	9	9	8
11	11	11	11	11	9	7	10	8	10
11	10	11	9	7	7	9	9	10	8
11	10	11	9	11	9	10	10	8	10
11	11	11	8	8	8	8	8	10	10
11	11	11	11	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10
11	11	11	11	10	10	10	10	10	10

Hình 2-14. Mức xám của điểm ảnh đầu vào

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

Hình 2-15. Nhãn mới được gán sau bước quét xuống lần 1

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

Hình 2-16. Nhãn thay đổi khi thực hiện phép quét từ dưới lên trên lần 1

1	1	2	2	1	2	2	2	2	2
1	1	2	2	2	2	2	2	2	2
1	1	1	3	3	3	3	3	2	2
1	1	1	3	3	3	3	3	2	2
4	4	3	3	3	3	3	3	2	2
4	4	3	3	3	3	3	3	2	2
4	4	2	3	3	3	3	2	2	2
4	4	2	2	2	2	2	2	2	2
0	0	2	2	2	2	2	2	2	2
0	0	0	2	2	2	2	2	2	2
0	0	2	2	2	2	2	2	2	2

**Hình 2-17. Quét từ trên xuống dưới lần 2**

Sau bước quét ta thấy vẫn còn một số điểm chưa được dán nhãn, tiếp tục việc quét từ dưới lên trên

1	1	2	2	1	2	2	2	2	2
1	1	2	2	2	2	2	2	2	2
1	1	1	3	3	3	3	3	2	2
1	1	1	3	3	3	3	3	2	2
4	4	3	3	3	3	3	3	2	2
4	4	3	3	3	3	3	3	2	2
4	4	2	3	3	3	3	2	2	2
4	4	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
0	0	2	2	2	2	2	2	2	2

**Hình 2-18. Quét từ dưới lên lần 2**

Vẫn tồn tại 2 điểm ảnh chưa được dán nhãn, tiếp tục việc quét từ trên xuống dưới một lần nữa

1	1	2	2	1	2	2	2	2	2
1	1	2	2	2	2	2	2	2	2
1	1	1	3	3	3	3	3	2	2
1	1	1	3	3	3	3	3	2	2
4	4	3	3	3	3	3	3	2	2
4	4	3	3	3	3	3	3	2	2
4	4	2	3	3	3	3	2	2	2
4	4	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2

**Hình 2-19. Quét ảnh từ trên xuống dưới lần 3**

Như ta thấy, lúc này toàn bộ ảnh đều đã được dán nhãn, thuật toán sẽ dừng việc quét ảnh ở đây vì nhãn đã được gán cho tất cả các điểm ảnh. Sau bước 3, ta được một hình ảnh được phân nhãn đầy đủ như hình 2-26:

1	1	2	2	2	2	2	2	2	2
1	1	2	2	2	2	2	2	2	2
1	1	1	3	3	3	3	3	2	2
1	1	1	3	3	3	3	3	2	2
4	4	3	3	3	3	3	3	2	2
4	4	3	3	3	3	3	3	2	2
4	4	2	3	3	3	3	2	2	2
4	4	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2

**Hình 2-20. Hình ảnh gán nhãn cuối cùng**

Toàn bộ thuật toán được mô tả như sau:

Input : f , Output : l

$v[p] \leftarrow 0$ ,  $l[p] \leftarrow 0$ , New\_label  $\leftarrow 0$ , Scan\_Step2  $\leftarrow 1$ , Scan\_Step3  $\leftarrow 1$

Scan from top left to bottom right : step1 (p)

**while** Scan\_Step2 = 1 **do**

    Scan image from top left to bottom right : step2 (p)

**if**  $v[p]$  is not changed **then**

        Scan\_Step2  $\leftarrow 0$

**else**

        Scan image from bottom right to top left : step2 (p)

**if**  $v[p]$  is not changed **then**

            Scan\_Step2  $\leftarrow 0$

**end if**

**end if**

**end while**

**while** Scan\_Step3 = 1 **do**

    Scan image from top left to bottom right : step3 (p)

**if**  $l[p]$  is not changed **then**

        Scan\_Step3  $\leftarrow 0$

```

else
    Scan image from bottom right to top left : step3 (p)
    if l[p] is not changed then
        Scan_Step3 ← 0
    end if
end if
end while
function step1 (p)
    if v[p] ≠ 1 then
        for each n of p // n is neighbour pixel of p
            if f[n] < f (p) then v[p] ← 1
            end if
        end for
    end if
end function
function step2 (p)
    if v[p] ≠ 1 then min ← VMAX,
        for each n of p // n is neighbor pixel of p
            if f (n) = f (p) and v[n] > 0 and v[n] < min then min ←
v[n]
            end if
        end for
        if min ≠ VMAX and v[p] ≠ (min+1) then v[p] ← min+1
        end if
    end if
end function
function step3 (p)
    lmin ← LMAX, fmin ← f (p)
    if v[p] = 0 then
        for each n of p
            if f (n) = f (p) and l[n] > 0 and l[n] < lmin then lmin ←
l[n]

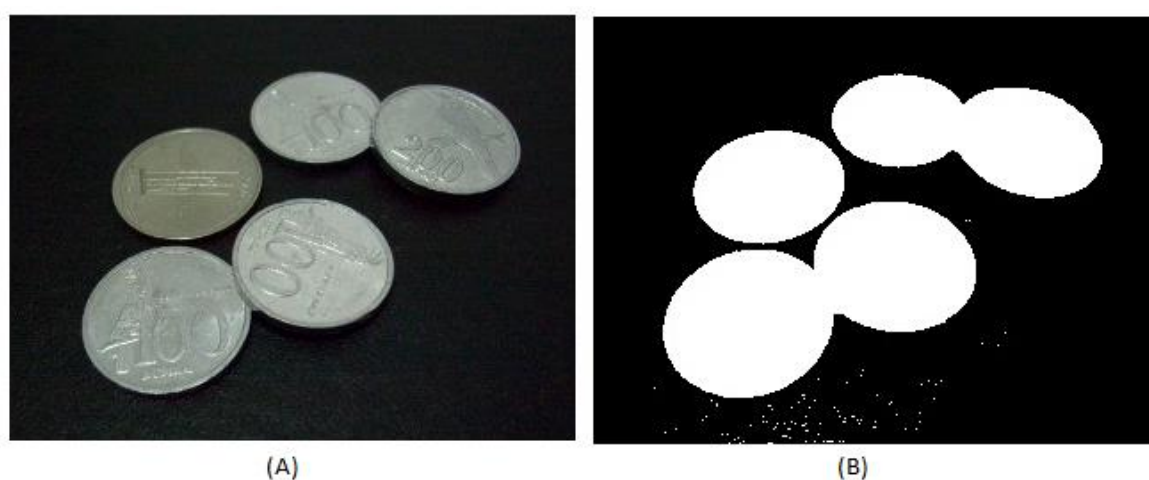
```

```
        end if
    end for
    if lmin = LMAX and l[p] = 0 then lmin ← New label + 1
    end if
    else if v[p] = 1 then
        for each n of p
            if f (n) < fmin then fmin ← f[n]
            end if
        for each n of p
            if f (n) = fmin and l[n] > 0 and l[n] < lmin then
                lmin ← l[n]
            end if
        end if
    else
        for each n of p
            if f (n) = f (p) and v[n] = v[p] - 1 and l[n] > 0 and l[n] <
                lmin then lmin ← l[n]
            end if
        end if
    end if
    if lmin ≠ LMAX and l (p) ≠ lmin then l[p] ← lmin
    end if
end function
```

### CHƯƠNG 3: THỰC NGHIỆM

#### 3.1 Phát biểu bài toán ứng dụng

Đếm và tách các đối tượng có hình dạng đơn giản và không dính nhau khá là dễ. Chỉ cần lấy biên từ ảnh nhị phân thì chúng ta có thể biết được số lượng đường biên. Nhưng nếu đối tượng dính nhau hoặc đè chồng, thì khó hơn. Ví dụ như hình 3-1A, nếu lấy ngưỡng của nó trong hình 3-1B thì số lượng biên là 3 trong khi đúng là 5. Để được kết quả đúng, chúng ta cần sử dụng kết hợp các kỹ thuật khác nhau trong xử lý ảnh. Trong phần này, đồ án sẽ thực nghiệm kết hợp kỹ thuật biến đổi khoảng cách để tách và đếm các đối tượng, sau đó áp dụng kỹ thuật phân đoạn ảnh theo phép biến đổi Watershed để xác định ranh giới giữa các đối tượng. Thực nghiệm được cài đặt sử dụng visual studio 2015 với thư viện OpenCV3.0.



Hình 3-1. Ảnh thực nghiệm

### 3.2 Khái quát về thư viện OpenCV

Giới thiệu về thư viện OpenCV:

OpenCV (Open Source Computer Vision) là một thư viện mã nguồn mở về thị giác máy với hơn 500 hàm và hơn 2500 các thuật toán đã được tối ưu về xử lý ảnh, và các vấn đề liên quan tới thị giác máy. OpenCV được thiết kế một cách tối ưu, sử dụng tối đa sức mạnh của các dòng chip đa lõi... để thực hiện các phép tính toán trong thời gian thực, nghĩa là tốc độ đáp ứng của nó có thể đủ nhanh cho các ứng dụng thông thường.

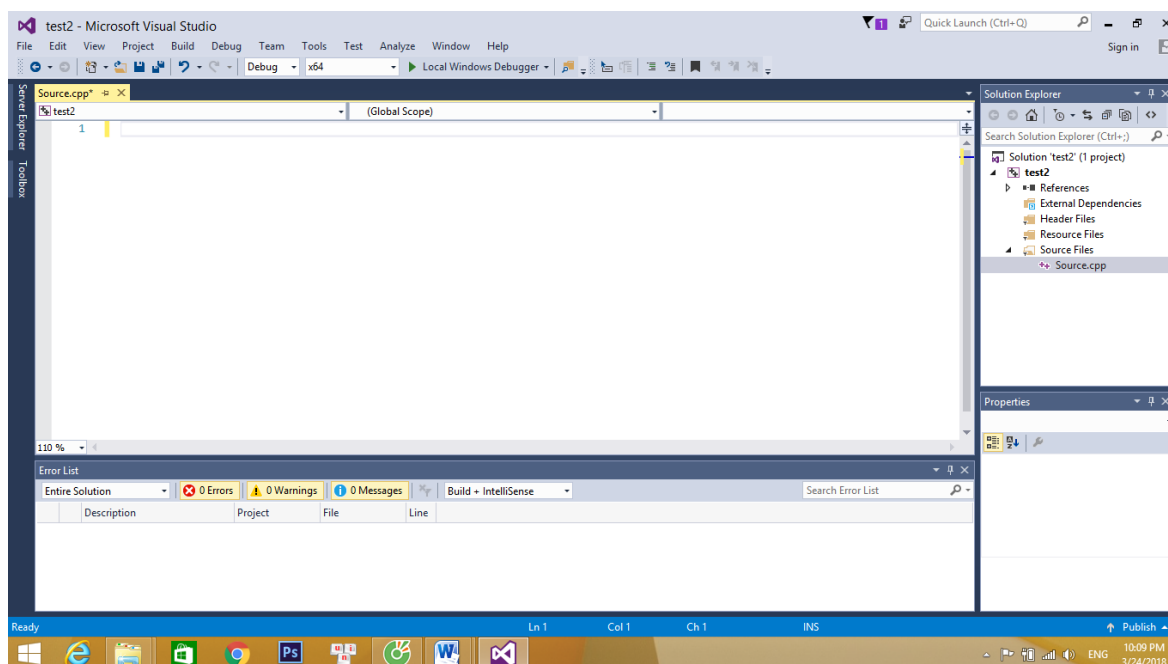
OpenCV là thư viện được thiết kế để chạy trên nhiều nền tảng khác nhau (cross-platform) , nghĩa là nó có thể chạy trên hệ điều hành Window, Linux, Mac, iOS ... Việc sử dụng thư viện OpenCV tuân theo các quy định về sử dụng phần mềm mã nguồn mở BSD do đó bạn có thể sử dụng thư viện này một cách miễn phí cho cả mục đích phi thương mại lẫn thương mại. Dự án về OpenCV được khởi động từ những năm 1999, đến năm 2000 nó được giới thiệu trong một hội nghị của IEEE về các vấn đề trong thị giác máy và nhận dạng, tuy nhiên bản OpenCV 1. 0 mãi tới tận năm 2006 mới chính thức được công bố và năm 2008 bản 1. 1 (pre-release) mới được ra đời. Tháng 10 năm 2009, bản OpenCV thế hệ thứ hai ra đời (thường gọi là phiên bản 2. x) , phiên bản này có giao diện của C++ (khác với phiên bản trước có giao diện của C) và có khá nhiều điểm khác biệt so với phiên bản thứ nhất. Thư viện OpenCV ban đầu được sự hỗ trợ từ Intel, sau đó được hỗ trợ bởi Willow Garage, một phòng thí nghiệm chuyên nghiên cứu về công nghệ robot. Cho đến nay, OpenCV vẫn là thư viện mở, được phát triển bởi nguồn quỹ không lợi nhuận (none -profit foundation) và được sự hưởng ứng rất lớn của cộng đồng[4].

### 3.3 Các bước thực hiện

Hình ảnh được sử dụng trong bài toán này là ảnh các đồng xu chồng lấp một phần lên nhau. Trong phần này đề án sẽ thực hiện đếm và xác định ranh



giới giữa những đối tượng đồng xu đè chồng lên nhau bằng cách sử dụng hàm biến đổi khoảng cách Distance transform và thuật toán Watershed.



**Hình 3-2. Giao diện chương trình cài đặt**

### 3.3.1 Bước 1: Tiền xử lý

Bước này gồm có 2 thao tác cần thực hiện: Lọc nhiễu và phân ngưỡng

#### **Đọc ảnh (Input Image) :**

Ta có ảnh đầu vào được đọc bằng hàm `Imread` có sẵn trong thư viện của OpenCV. Hàm `Imread` là một hàm dùng để đọc hình ảnh từ file với đối số

truyền vào là một input đầu vào chứa tên file, đối số thứ hai là một biến chỉ định cách mà OpenCV sẽ load hình.

Cấu trúc của hàm `imread` trong thư viện OpenCV được trình bày như sau:

**Mat image;**

**Image=imread (INPUT, CV\_LOAD\_IMAGE\_COLOR) ;**

Trong đó:

- INPUT là file ảnh đầu vào cần đọc.
- CV\_LOAD\_IMAGE\_COLOR: Hình ảnh load lên dưới định dạng 3 kênh màu RGB (Red, Green, Blue) .
- CV\_LOAD\_IMAGE\_GRAYSCALE: Hình ảnh load lên dưới dạng hình ảnh xám.
- CV\_LOAD\_IMAGE\_UNCHANGED: Hình ảnh load lên được giữ nguyên cấu trúc của nó.
- Mat là cấu trúc dùng để lưu giữ data của hình ảnh được sử dụng trong OpenCV.

Sau khi load ảnh bằng hàm `Imread` ta có được hình ảnh 3-4:



**Hình 3-3. Hình ảnh đầu vào**

### Lọc nhiễu:

Sau khi đã có được ảnh đầu vào, thao tác tiếp theo cần thực hiện trong bước tiền xử lý là chúng ta sẽ đi lọc nhiễu cho bức ảnh đầu vào. Mục đích của thao tác lọc nhiễu là để loại bỏ một số loại nhiễu không cần thiết có thể ảnh hưởng đến quá trình tìm kiếm đối tượng sau này.

Để thực hiện thao tác lọc nhiễu, đầu tiên ta cần chuyển hình ảnh đầu vào từ ảnh màu qua ảnh xám để có thể áp dụng được một số kỹ thuật lọc:

Trong OpenCV chúng ta có thể sử dụng hàm `CvtColor` để thực hiện chuyển ảnh đầu vào từ ảnh màu qua ảnh xám. Cấu trúc của hàm `CvtColor` trong OpenCV được trình bày như sau:

**Mat bw;**

**CvtColor (src, bw, CV\_BGR2GRAY) ;**

Trong đó:

- src là đối số đầu vào
- bw là đối số đầu ra
- CV\_BGR2GRAY là code quy định sẽ chuyển đổi từ mã màu nào sang mã màu nào. Ở đây ta sẽ thực hiện việc chuyển đổi từ mã ảnh màu sang ảnh xám.

Sau khi chuyển đổi hệ màu của ảnh gốc sang ảnh xám bằng hàm `CvtColor` trong OpenCV ta có được hình ảnh 3-4:



**Hình 3-4. Ảnh xám sau khi được chuyển đổi**

Thao tác tiếp theo cần thực hiện đó là lọc nhiễu cho ảnh. Để lọc nhiễu cho bức ảnh này, đề án sẽ sử dụng bộ lọc làm mờ Blur. Là một phép lọc làm cho trơn ảnh và khử nhiễu hạt và là một bộ lọc trung bình.

Trong OpenCV hàm Blur được trình bày như sau:

**blur (src, dst, Size ksize, (-1, -1) , 4)**

Trong đó:

- src là ảnh đầu vào
- dst là ảnh sau khi thực hiện phép lọc.
- ksize là kích thước của ma trận lọc.
- (-1, -1) là giá trị mặc định của ma trận lọc
- giá trị 4 là giá trị mặc định được dùng để ước lượng và căn chỉnh các điểm ảnh nếu phép lọc chúng vượt ra khỏi giới hạn của ảnh.

Sau khi thực hiện lọc ảnh bằng hàm Blur trong OpenCV ta được kết quả là bức ảnh đã được làm mờ:



**Hình 3-5. Kết quả của bước làm mờ ảnh**

### **Phân ngưỡng**

Tiếp theo là thao tác chuyển qua ảnh nhị phân, hay còn gọi là phân ngưỡng, bức ảnh sau khi được lọc bằng hàm Blur ở thao tác trên sẽ được loại bỏ nhiễu ở thao tác này bằng việc chọn một mức ngưỡng phù hợp. Mục đích của thao tác phân ngưỡng ảnh là dùng để phục vụ cho việc tách các đối tượng của các thuật toán tiếp theo.

Để thực hiện việc phân ngưỡng trong OpenCV ta sẽ sử dụng hàm `threshold`, trong OpenCV hàm `threshold` được trình bày như sau:

**`Threshold (src, dst, 40, 255, CV_THRESH_BINARY)`**

Trong đó:

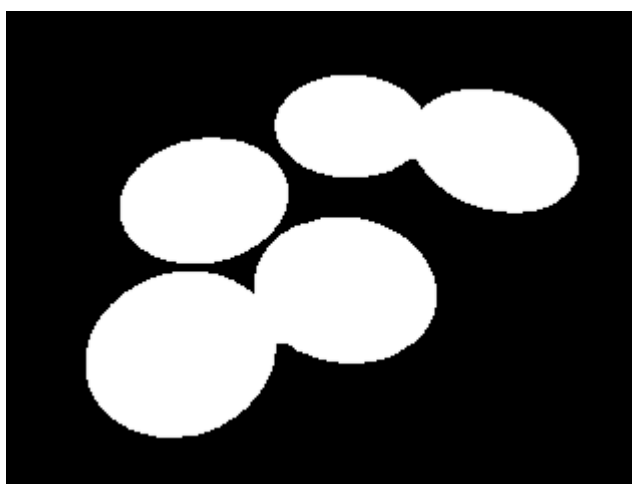
- `src` là ảnh xám đầu vào
- `dst` là ảnh đầu ra
- 40 là giá trị ngưỡng
- 255 là giá trị được gán nếu pixel giá trị lớn hơn giá trị ngưỡng
- `CV_THRESH_BINARY` là loại phân ngưỡng

Tùy theo các loại phân ngưỡng mà pixel được gán giá trị khác nhau:

- `THRESH_BINARY`: Nếu giá trị pixel lớn hơn ngưỡng thì gán bằng `maxval`. Ngược lại bằng gán bằng 0

- THRESH\_BINARY\_INV: Nếu giá trị pixel lớn hơn ngưỡng thì gán bằng 0. Ngược lại bằng gán bằng maxval
- THRESH\_TRUNC: Nếu giá trị pixel lớn hơn ngưỡng thì gán giá trị bằng ngưỡng. Ngược lại giữ nguyên giá trị

Sau khi thực hiện phân ngưỡng bằng hàm threshold trong OpenCV ta được kết quả là bức ảnh đã được phân ngưỡng như sau:



**Hình 3-6. Kết quả của quá trình phân ngưỡng**

### **3.3.2 Bước 2: Tách đối tượng**

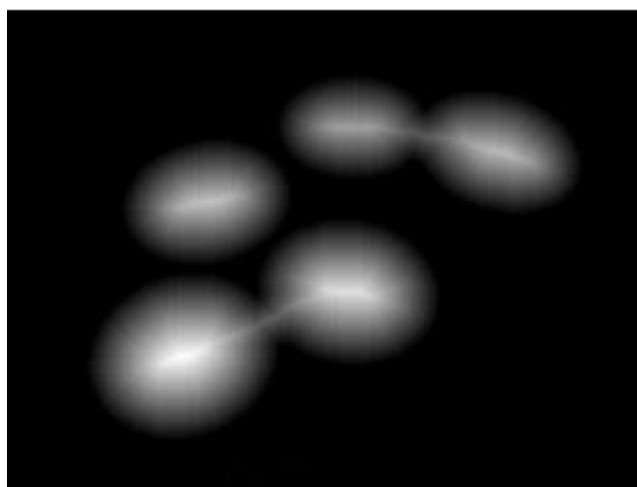
Hình ảnh được sử dụng là hình ảnh các đồng xu chồng lấp một phần lên nhau, nếu tìm biên và đếm số biên từ hình ảnh nhị phân thì kết quả sẽ không chính xác trong trường hợp này. Để đếm chính xác được số tượng đồng xu trong ảnh thì ta phải tách các đối tượng trong hình ảnh thành các đối tượng không trùng lặp.

Để thực hiện việc tách các đối tượng trong ảnh thành các vùng không trùng lặp, trong đồ án này sẽ sử dụng thuật toán biến đổi khoảng cách (Distance transform) . Trong trường hợp này, với mỗi điểm ảnh đối tượng sẽ được thay thế bằng khoảng cách của nó tới điểm ảnh nền gần nhất. Nghĩa là sau khi sử dụng thuật toán, các điểm ảnh ở gần tâm đối tượng sẽ có giá trị cao hơn, còn các điểm ảnh càng ở gần biên sẽ có giá trị nhỏ hơn. Trong OpenCV hàm biến đổi khoảng cách được trình bày như sau:

**distanceTransform (bw, dist, CV\_DIST\_L2, 3) ;**

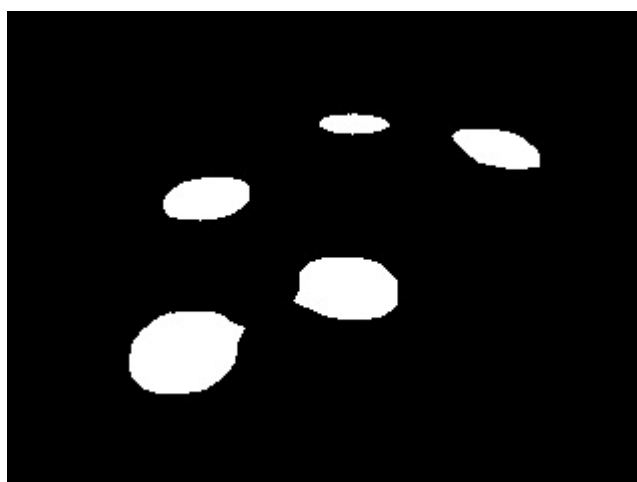
Sau khi thực hiện thuật toán biến đổi khoảng cách và thực hiện chuẩn hóa kết quả về khoảng  $[0, 1]$ , tức là những điểm sau khi thực hiện biến đổi khoảng cách sẽ có một thông số giá trị, điểm càng xa biên sẽ có giá trị càng lớn, điểm càng gần biên sẽ có giá trị càng nhỏ, giá trị này nằm trong khoảng  $[0, 1]$

Hình ảnh kết quả sau khi biến đổi khoảng cách và chuẩn hóa:



**Hình 3-7. Hình ảnh sau khi sử dụng hàm biến đổi khoảng cách**

Để hoàn thành việc tách đối tượng thành các vùng rời rạc ta sẽ thực hiện phân ngưỡng một lần nữa đối với ảnh đã thực hiện phép biến đổi khoảng cách:



**Hình 3-8. Các đối tượng đã được tách**

Sau khi các đối tượng đã được tách thành các vùng rời rạc, việc tiếp theo cần làm đó là đi tìm biên của chúng.

Để thực hiện được việc tìm biên của các đối tượng chúng ta sẽ sử dụng hàm **findContours**. Trong OpenCV hàm tìm biên được trình bày như sau:

```
findContours (dist_8u, contours, CV_RET_EXTRNAL,  
CV_CHAIN_APPROX_SIMPLE) ;
```

Trong đó:

- dist\_8u là hình ảnh cần tìm biên.
- Contours là biến lưu trữ các đường biên tìm được, mỗi đường biên được lưu trữ dưới dạng một vecto của các điểm.
- CV\_RET\_EXTRNAL khi sử dụng hàm này thì hàm sẽ chỉ lấy những đường biên bên ngoài, những đường biên bên trong của đối tượng bị loại bỏ.
- CV\_CHAIN\_APPROX\_SIMPLE hàm này sẽ có tác dụng nén đường viền trước khi lưu trữ, nén phân đoạn theo chiều ngang, chiều dọc và chéo.

Sau khi đếm số biên bằng hàm **findContours** chúng ta sẽ có được số lượng biên tương ứng với số đối tượng. Để biết có bao nhiêu đường biên tương ứng với đối tượng được tìm ra ta sử dụng câu lệnh sau trong OpenCv:

```
int ncomp = contours.size () ;
```

### 3.3.3 Xác định ranh giới giữa các đối tượng

Ở bước này chúng ta sẽ áp dụng thuật toán watershed để tìm ranh giới giữa các đối tượng riêng biệt. Để thực hiện được thuật toán watershed việc đầu tiên chúng ta phải làm đó là tạo ảnh marker. Ảnh marker là ảnh cho thuật toán watershed biết đâu là đối tượng đang xét và đâu là đối tượng nền.

Trong OpenCV, thuật toán tạo ảnh marker được trình bày như sau:



```
Mat markers = Mat zeros (dist. size () , CV_32SC1) ;
```

```
For (int i=0; i< ncomp; i++)
```

```
    drawContours (marker, coutour, i, cv::Scalar::all (i+1) , -1) ;
```

Trong thuật toán tạo ảnh marker phía trên có sử dụng một vòng lặp for nhằm mục đích truy cập đến từng đường biên của đối tượng đã được tìm ra ở thuật toán findContours để vẽ đường viền bằng hàm drawContours cho nó. Các đường viền này sẽ được dùng để nhận biết đó là những đối tượng trên mảnh marker.

Ngoài ra chúng ta cũng cần phải vẽ marker nền để thuật toán watershed phân biệt. Giả sử đối tượng được chỉ định ở giữa ảnh thì chúng ta cần vẽ marker gần cạnh bức ảnh.

Thuật toán vẽ marker nền được trình bày như sau trong OpenCV:

```
Circle (markers, Point (5, 5) , 3, CV_RGB (255, 255, 255) , -1)
```

Sau khi đã tạo được ảnh marker ta thực hiện thuật toán watershed:

```
watershed (src, markers) ;
```

Bước tiếp theo cần thực hiện đó là tạo màu ngẫu nhiên cho các đối tượng:

Tạo màu ngẫu nhiên cho các đối tượng trong OpenCV:

```
vector<Vec3b> colors;
```

```
for (int i = 0; i < ncomp; i++)
```

```
{
```

```
    int b = theRNG () . uniform (0, 255) ;
```

```
    int g = theRNG () . uniform (0, 255) ;
```

```
    int r = theRNG () . uniform (0, 255) ;
```

```
    colors. push_back (Vec3b ( (uchar) b, (uchar) g, (uchar) r) ) ;
```

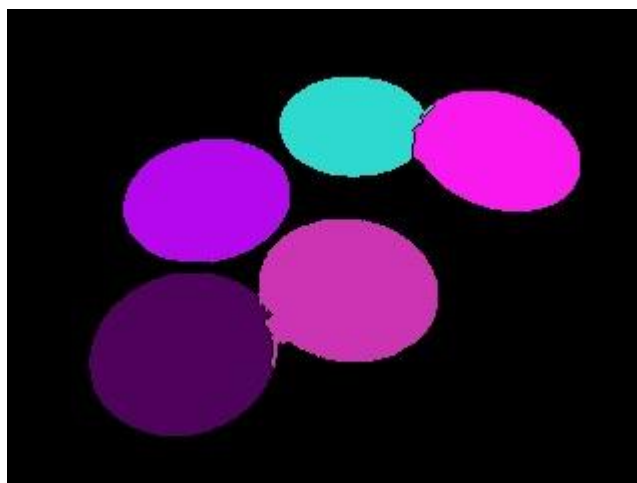
```
}
```

Tạo ảnh kết quả:

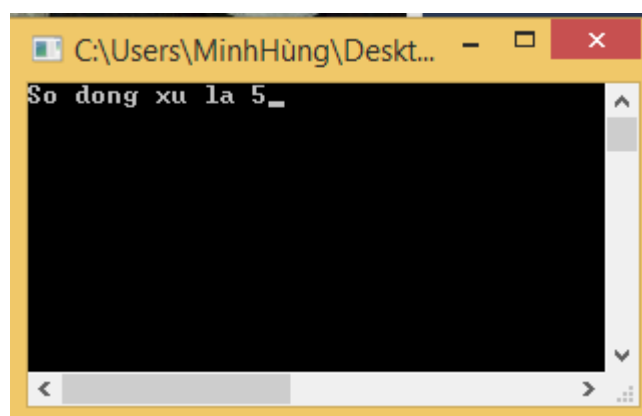
```
Mat dst = cv::Mat::zeros (markers. size () , CV_8UC3) ;
```

Điền các đối tượng được gán nhãn với màu sắc ngẫu nhiên.

```
for (int i = 0; i < markers. rows; i++)  
{  
    for (int j = 0; j < markers. cols; j++)  
    {  
        int index = markers. at<int> (i, j) ;  
        if (index > 0 && index <= ncomp)  
            dst. at<cv::Vec3b> (i, j) = colors[index-1];  
        else  
            dst. at<cv::Vec3b> (i, j) = cv::Vec3b (0, 0, 0) ;  
    }  
}
```



**Hình 3-9. Hình ảnh phân đoạn bằng thuật toán watershed**



**Hình 3-10. Số lượng đồng xu đếm được**

## KẾT LUẬN

Sau một thời gian tìm hiểu và nghiên cứu đề án, em đã đạt được một số kết quả sau:

- Tìm hiểu được các hướng và các phương pháp giải quyết bài toán tách và đếm đối tượng trong ảnh.
- Tìm hiểu về kỹ thuật phân đoạn dựa vào thuật toán watershed và ứng dụng vào tách đối tượng chồng lấp trong ảnh.
- Tìm hiểu được cách thức làm việc của một số thuật toán trong xử lý ảnh.
- Xây dựng được một chương trình áp dụng phép toán nở vùng.
- Mô tả và giải thích được các vấn đề cơ bản trong phân đoạn ảnh và tiền xử lý ảnh.

Do kiến thức còn hạn hẹp và đề tài còn quá mới mẻ đối với em lên trong quá trình thực thi báo cáo không tránh khỏi sai sót.

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

### MỘT SỐ TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Đình Dương, 2011. Ô nhiễm dầu trên biển và quan trắc bằng viễn thám siêu cao tần, NXB Khoa học và kỹ thuật, Hà Nội.
- [2]. Longin Jan Latecki, Image Segmentation Using Region Growing and Shrinking, Computer Graphics and Image Processing
- [3]. Sameer Ruparelia, Implementation of Watershed Based Image Segmentation Algorithm in FPGA, March 15, 2012
- [4]. Nguyễn Văn Long. Ứng dụng xử lý ảnh trong thực tế với thư viện OpenCV C/C++
- [5]. [Http://thigiacmaytinh.com/thuat-toan-watershed](http://thigiacmaytinh.com/thuat-toan-watershed)
- [6]. Rafael C. Gonzalez, Digital image processing using Matlab.
- [7]. George J. Grevera Saint Joseph's University Philadelphia, Pennsylvania, USA. distancetransformalgorithmsand their implementationand evaluation