

# Mục lục

<b>1</b>	<b>Xử lý ảnh và các vấn đề cơ bản của xử lý ảnh</b>	<b>4</b>
1.1	Xử lý ảnh là gì? . . . . .	4
1.2	Một số vấn đề trong xử lý ảnh . . . . .	6
1.2.1	Một số khái niệm trong xử lý ảnh . . . . .	6
1.2.2	Tăng cường khôi phục ảnh . . . . .	7
1.2.3	Biến đổi ảnh . . . . .	7
1.2.4	Phân tích ảnh . . . . .	8
1.2.5	Nhận dạng ảnh . . . . .	8
1.2.6	Nén ảnh . . . . .	9
<b>2</b>	<b>Một số phương pháp tiếp cận phân vùng ảnh</b>	<b>10</b>
2.1	Phân vùng ảnh theo ngưỡng biên độ . . . . .	10
2.2	Phân vùng ảnh theo miền đồng nhất . . . . .	11
2.2.1	Phương pháp tách cây tứ phân . . . . .	12
2.2.2	Phân vùng ảnh dựa vào phát triển vùng cục bộ . . . . .	13
2.2.3	Phân vùng ảnh dựa trên hợp và tách vùng . . . . .	14
2.3	Phân vùng ảnh dựa trên phân tích kết cấu . . . . .	14
2.4	Phân vùng ảnh dựa trên sự phân lớp điểm ảnh . . . . .	14
2.5	Phân vùng ảnh dựa vào lý thuyết đồ thị . . . . .	16
2.5.1	Cách biểu diễn ảnh dưới dạng đồ thị . . . . .	16
2.5.2	Các định nghĩa và tính chất liên quan . . . . .	17

2.6	Phân vùng ảnh dựa trên biểu diễn và xử lý đa phân giải . . . . .	18
<b>3</b>	<b>Phương pháp phân vùng ảnh sử dụng biên động</b>	<b>19</b>
3.1	Phương pháp Chan-Vese(CV) . . . . .	20
3.1.1	Mô hình . . . . .	20
3.1.2	Giải quyết vấn đề sử dụng phương pháp tập mức . . . . .	22
3.1.3	Ưu nhược điểm của mô hình . . . . .	25
3.2	Phương pháp Local binary fitting energy (LBF) . . . . .	26
3.2.1	Mô hình . . . . .	26
3.2.2	Giải quyết vấn đề sử dụng phương pháp tập mức . . . . .	29
3.2.3	Ưu nhược điểm của mô hình . . . . .	31
3.3	Phương pháp Local image fitting(LIF) . . . . .	33
3.3.1	Mô hình và phương trình với tập mức . . . . .	33
3.3.2	Ưu nhược điểm của mô hình . . . . .	35
3.4	Mô hình kết hợp Global và Local . . . . .	36
3.4.1	Mô hình . . . . .	36
3.4.2	Giải số . . . . .	38
3.4.3	Kết quả . . . . .	39
<b>4</b>	<b>Áp dụng mô hình GLIF trong theo dõi đối tượng trong video</b>	<b>42</b>
4.1	Theo dõi đối tượng trong video là gì? Tại sao phải theo dõi đối tượng trong video . . . . .	42
4.2	Biểu diễn đối tượng . . . . .	43
4.2.1	Lựa chọn đặc tính để thực hiện theo vết . . . . .	44
4.3	Theo vết đối tượng sử dụng mô hình GLIF . . . . .	45
4.4	Kết quả . . . . .	46
<b>5</b>	<b>Áp dụng GLIF trong phân vùng ảnh 3 chiều</b>	<b>47</b>
5.1	Ảnh 3 chiều trong thực tế . . . . .	47

5.2	Kết quả mô hình GLIF cho ảnh ba chiều . . . . .	48
-----	---	----

# Chương 1

## Xử lý ảnh và các vấn đề cơ bản của xử lý ảnh

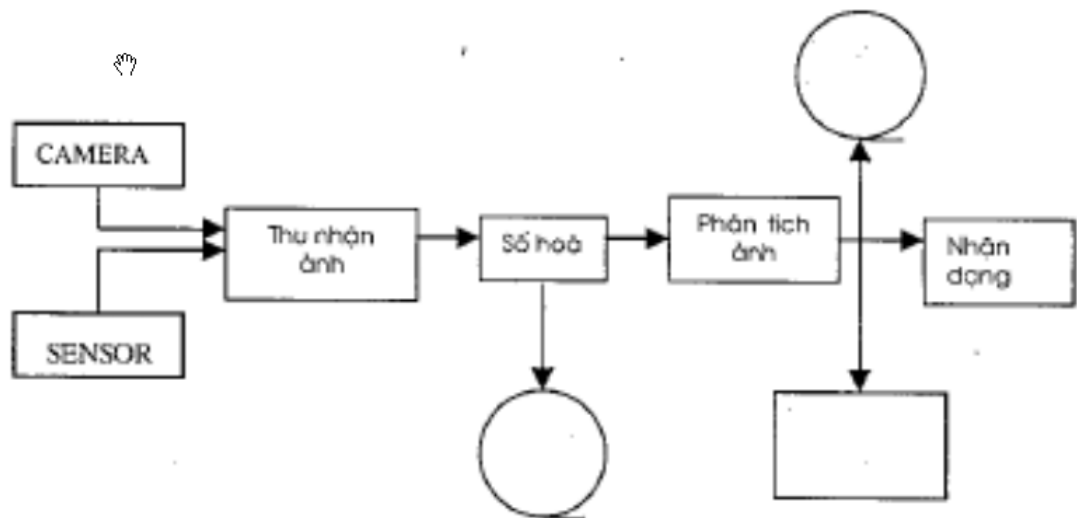
### 1.1 Xử lý ảnh là gì?

Một ảnh có thể được đại diện bởi một hàm 2 chiều  $f(x, y)$ .  $(x, y)$  là tọa độ của các điểm trong mặt phẳng, giá trị  $f$  tại các cặp tọa độ  $(x, y)$  được gọi là cường độ hay mức xám của ảnh tại điểm đó. Ảnh số là tập hợp hữu hạn các phần tử, mỗi phần tử có một tọa độ riêng và một giá trị  $f$  riêng. Các giá trị này gọi là *pixel* hay *imageelements*. Một trong các ứng dụng của ảnh số là trong ngành công nghiệp báo chí, khi ảnh lần đầu tiên được truyền qua cáp biển giữa London và NewYork. Việc truyền ảnh qua cáp biển này đã rút ngắn quá trình truyền ảnh qua Đại Tây Dương từ hơn một tuần xuống còn chưa đến 3 giờ. Một máy in đặc biệt mã hóa bức ảnh để truyền tải đi và khôi phục lại ảnh gốc ở đầu cuối. Sự phát triển của phần cứng máy tính đã giúp cho lĩnh vực đồ họa và xử lý ảnh phát triển một cách mạnh mẽ và ngày càng có nhiều ứng dụng trong cuộc sống.



**Hình 1.1:** Một bức ảnh được tạo ra bởi máy điện ấn năm 1921

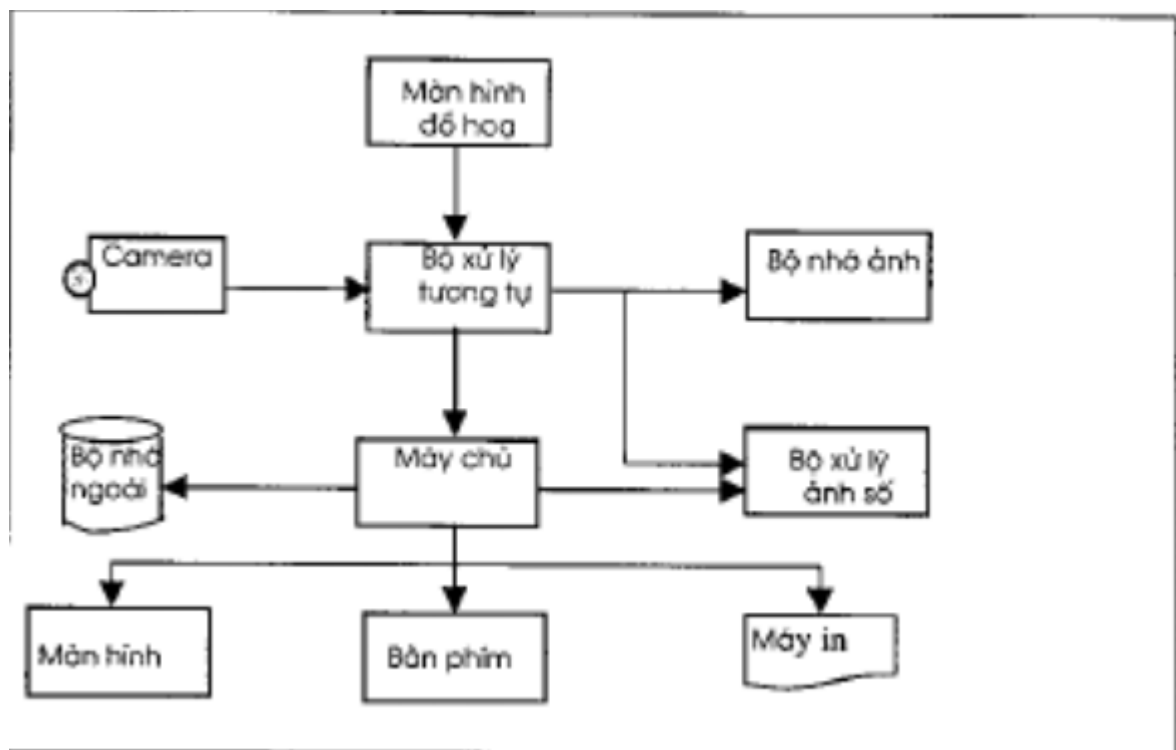
Quá trình xử lý ảnh được xem là quá trình thao tác với ảnh đầu vào để đưa ra một kết quả mong muốn. Quá trình xử lý ảnh có thể được mô tả theo sơ đồ sau:



**Hình 1.2:** Các quá trình của xử lý ảnh

Đầu tiên là quá trình thu nhận ảnh. Ảnh có thể được thu nhận qua camera. Đầu ra có thể là dạng tín hiệu tương tự hoặc cũng có thể là tín hiệu số hóa. Ngoài ra ảnh còn có thể được thu nhận qua bộ cảm ứng sensor, tranh, ảnh được quét thông qua máy quét. Quá trình số hóa thực hiện biến đổi tín hiệu tương tự thành tín hiệu rời rạc sau đó được đưa vào quá trình phân tích, xử lý hoặc thực hiện lưu trữ ảnh.

Quá trình phân tích ảnh bao gồm nhiều bước nhỏ. Với một số ảnh, do thiết bị thu nhận ảnh, nguồn sáng hay nhiễu mà ảnh có thể bị suy biến. Do vậy chúng ta cần thực hiện quá trình tăng cường ảnh nhằm tăng chất lượng hoặc làm nổi bật một số đặc tính chính của ảnh. Tiếp theo là quá trình phát hiện các đặc tính như biên ảnh (Edge Detection), phân vùng ảnh (Segmentation)... Cuối cùng tùy theo ứng dụng mà có thể thực hiện quá trình nhận dạng, phân lớp hay ra quyết định. Một hệ xử lý ảnh có thể được mô tả gồm các thành phần như bên dưới



**Hình 1.3:** Các thành phần chính của hệ thống xử lý ảnh

## 1.2 Một số vấn đề trong xử lý ảnh

### 1.2.1 Một số khái niệm trong xử lý ảnh

- **Điểm ảnh:** là một phần tử nhỏ nhất của ảnh. Một điểm ảnh có thể xem là tọa độ và cường độ sáng tại một điểm trong không gian của đối tượng.

- Mức xám, màu: là kết quả của sự mã hóa cường độ sáng của một điểm ảnh với một giá trị số.

### 1.2.2 Tăng cường khôi phục ảnh

Tăng cường và khôi phục ảnh là hai quá trình có mục đích khác nhau. Tăng cường ảnh bao gồm các thao tác nhằm làm hoàn thiện trạng thái quan sát của một ảnh như: tăng cường độ tương phản, khử nhiễu. Trong khi đó khôi phục ảnh nhằm khôi phục ảnh gần với ảnh thực nhất do nó bị biến dạng do nhiều nguyên nhân khác nhau.

Việc khôi phục ảnh nhằm tối thiểu hóa ảnh hưởng của các yếu tố môi trường bên ngoài và các hệ thống thu nhận và lưu trữ ảnh. Quá trình khôi phục ảnh có thể coi là quá trình xây dựng mô hình toán học cho các ảnh hưởng sau đó dùng ánh xạ ngược để xác định lại ảnh

### 1.2.3 Biến đổi ảnh

Trong xử lý ảnh, một số trường hợp có phức tạp tính toán cao hoặc là số lượng điểm ảnh lớn. Việc dùng các phương pháp thông thường trở nên không khả thi. Vì vậy người ta thực hiện xây dựng một không gian khác đơn giản hơn và thực hiện bước chuyển đổi từ không gian gốc sang không gian này để dễ tính toán hơn. Sau khi thực hiện xử lý trên không gian mới xong, dùng phép biến đổi ngược để đưa về miền xác định ban đầu. Một số biến đổi thường được hay dùng là:

- Biến đổi Fourier, Sin, Cosin
- Biến đổi bằng tích chập, tích Kronecker
- Các biến đổi như Kahumen Loeve, Hadamard

Ngoài ra người ta còn sử dụng một số công cụ thống kê trong biến đổi ảnh.

### 1.2.4 Phân tích ảnh

Việc phân tích ảnh nhằm đưa ra các mô tả đặc trưng về ảnh. Các kỹ thuật được sử dụng ở bước này nhằm xác định biên của ảnh. Ngoài ra người ta cũng sử dụng các kỹ thuật để phân vùng ảnh. Các kỹ thuật thường hay dùng là tách, hợp thành phần dựa theo các tiêu chuẩn đánh giá như màu sắc, cường độ ,... Các phương pháp thường được dùng như phân tích cây tứ phân, phân vùng dựa theo cấu trúc, phân vùng dựa theo đồ thị.

### 1.2.5 Nhận dạng ảnh

Nhận dạng là quá trình phân loại các đối tượng được biểu diễn theo một mô hình nào đó và gán cho chúng vào các lớp dựa theo các quy luật. Quá trình nhận dạng theo các mẫu có trước được gọi là học có thầy (supervised learning), trong trường hợp ngược lại gọi là học không có thầy. Có ba cách tiếp cận khác nhau trong nhận dạng ảnh:

- Nhận dạng dựa vào phân hoạch không gian
- Nhận dạng cấu trúc
- Nhận dạng theo mạng nơron

Với hai cách tiếp cận đầu, sau khi thu nhận, ảnh cần trải qua các bước tiền xử lý làm tăng chất lượng và làm nổi bật các chi tiết, sau đó là quá trình biểu diễn và trích chọn các đặc trưng, cuối cùng mới là quá trình nhận dạng.

Trong thực tế, người ta đã áp dụng thành công nhận dạng cho nhiều đối tượng khác nhau như:

- Nhận dạng vân tay phục vụ cho mục đích bảo mật và phát hiện tội phạm
- Nhận dạng chữ in phục vụ cho quá trình số hóa tài liệu, tự động đọc tài liệu
- Nhận dạng chữ viết tay với các ràng buộc về cách viết, kiểu chữ...



### 1.2.6 Nén ảnh

Việc truyền ảnh qua mạng là một thao tác được dùng thường xuyên, trong khi lượng thông tin dùng để biểu diễn ảnh lại là khá lớn. Do vậy việc nén ảnh để giảm lượng thông tin truyền đi là một nhu cầu cần thiết.

Nén ảnh là kỹ thuật lợi dụng sự dư thừa thông tin của ảnh nhằm biến đổi dữ liệu ảnh thành dạng từ mã. Mỗi phương pháp nén ảnh dựa trên các định nghĩa dư thừa khác nhau. Các định nghĩa dư thừa hay được dùng như: sự phân bố mức xám, sự lặp lại mức xám, những mẫu sử dụng tần suất cao hoặc độ dư thừa vị trí. Tùy vào thuật toán và loại ảnh mà ta có tỷ lệ nén khác nhau:

$$\text{Tỷ lệ nén} = \frac{\text{Dữ liệu sau nén}}{\text{Dữ liệu trước khi nén}} \quad (1.1)$$

Dựa theo các cách phân loại khác nhau người ta có các phương pháp nén ảnh khác nhau:

- Phân loại theo lý thuyết nén
  - Nén không mất mát thông tin
  - Nén mất mát thông tin
- Phân loại dựa theo cách thức thực hiện nén:
  - Phương pháp không gian
  - Phương pháp dựa trên biến đổi
- Phân loại dựa trên phương pháp mã hóa:
  - Phương pháp nén thể hệ thứ nhất
  - Phương pháp nén thể hệ thứ hai

## Chương 2

# Một số phương pháp tiếp cận phân vùng ảnh

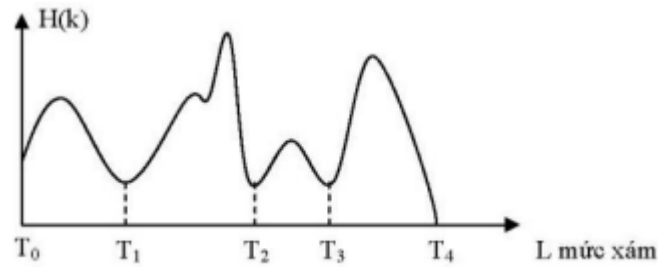
Phân vùng ảnh là một bước quan trọng trong xử lý ảnh, quá trình này nhằm phân tích ảnh thành các thành phần có cùng tính chất nào đó dựa theo biên hay các vùng liên thông. Vùng ảnh là tập hợp các điểm ảnh có cùng hay gần cùng một đặc điểm nào đó. Ví dụ như mức xám, mức màu .. Đường bao quanh vùng ảnh được gọi là biên ảnh

### 2.1 Phân vùng ảnh theo ngưỡng biên độ

Một ảnh được đặc trưng bởi các tính chất như: mức xám, độ tương phản, màu sắc,.... Ta có thể dùng ngưỡng biên độ với các đặc trưng trên để phân đoạn ảnh trong trường hợp ngưỡng biên độ đủ lớn đặc trưng cho ảnh. Kỹ thuật phân ngưỡng theo biên độ có lợi thế với ảnh nhị phân như bản in, đồ họa, ảnh màu Việc chọn ngưỡng sẽ bao gồm các bước sau:

- Xác định các đỉnh và khe của ảnh dựa vào việc phân tích lược đồ xám. Các khe có thể dùng để chọn ngưỡng
- Chọn ngưỡng  $t$  sao cho một phần xác định trước của toàn bộ mẫu là thấp hơn  $t$

- Điều chỉnh ngưỡng dựa trên lược đồ xám của các điểm lân cận
- Chọn ngưỡng theo lược đồ xám của các điểm thỏa mãn tiêu chuẩn chọn
- Khi có mô hình phân lớp xác suất, ta thực hiện xác định ngưỡng dựa vào tiêu chuẩn xác suất nhằm cực tiểu hóa xác suất sai
- Ta có ví dụ minh họa cho việc phân vùng dựa trên ngưỡng như sau:



**Hình 2.1:** Ví dụ minh họa cách chọn ngưỡng

Ở đây 5 ngưỡng được chọn và ảnh sẽ được phân vùng thành 4 vùng. Ta ký hiệu các vùng đó là  $C_k$  trong đó  $k = 1, 2, 3, 4$ . Các vùng sẽ được phân như sau:

$$P(m, n) \in C_k \text{ nu } T_{k-1} < P(m, n) < T_k; k = 1, 2, 3, 4 \quad (2.1)$$

Nếu sau khi phân vùng theo ngưỡng vừa mà chọn ảnh rõ nét thì kết thúc, ngược lại phải điều chỉnh ngưỡng và phân vùng lại cho đến khi đạt được kết quả mong muốn

## 2.2 Phân vùng ảnh theo miền đồng nhất

Kỹ thuật này sử dụng tính chất đồng nhất của các đặc trưng nào đó của ảnh. Cách thức phân vùng sẽ phụ thuộc vào việc lựa chọn đặc trưng dùng để thực hiện phân vùng. Các đặc trưng thường được dùng là mức xám với ảnh đen trắng và màu với ảnh màu. Với cách tiếp cận này, các phương pháp thường được sử dụng là :

- Phương pháp tách cây tứ phân

- Phương pháp cục bộ
- Phương pháp tổng hợp

### 2.2.1 Phương pháp tách cây tứ phân

Một tiêu chuẩn đồng nhất sẽ được chọn. Quá trình phân vùng sẽ thực hiện như sau: Đầu tiên ta thực hiện kiểm tra tính chất đồng nhất trên toàn bộ miền ảnh. Nếu tính đồng nhất thỏa mãn thì kết thúc thuật toán. Ngược lại miền ảnh sẽ được chia làm 4 miền con. Tiếp tục kiểm tra tính chất đồng nhất của các miền con và thực hiện phân tách khi tính chất đồng nhất không thỏa mãn. Quá trình kết thúc khi tất cả các miền con trong miền ảnh ban đầu thỏa mãn tính đồng nhất Phương pháp này được mô tả như sau

---

```

Func Phanvung(MienAnh)
{
    dsmiencon.Push(MienAnh)
    while(dsmiencon.Count()!=0)
    {
        miendangxet=dsmiencon.Pop();
        if(!ktdongnhat(miendangxet))
        {
            m1,m2,m3,m4=miendangxet.chiamien();
            for(int i=0;i<4;i++)
            {
                dsmiencon.Push(m_i)
            }
        }
    }
}

```

---

Tiêu chuẩn đồng nhất có thể dựa vào mức xám. Một cách đơn giản ta có thể chọn giá

trị chênh lệch giữa giá trị mức xám lớn nhất và giá trị mức xám nhỏ nhất trong miền đang xét. Hàm kiểm tra mức xám có thể được viết như sau: Giả sử  $(m_1, n_1), (m_2, n_2)$  là tọa độ điểm đầu và điểm cuối của miền đang xét

---

```
Func ktdongnhatch(miendangxet)
{
    int min=0;
    int max=255
    for(i=n1;i<n_2;i++)
    for(j=m1;j<m_2;j++)
    {
        if(I(i,j)<min)
            I(i,j)=min;
        if(I(i,j)>max)
            I(i,j)=max;
    }
    if((max-min)<nguong) return 1;
    return 0;
}
```

---

### 2.2.2 Phân vùng ảnh dựa vào phát triển vùng cục bộ

Ý tưởng của phương pháp này ngược lại với phương pháp cây tứ phân. Phương pháp thực hiện xét từ các miền ảnh nhỏ nhất của ảnh rồi nối chúng lại nếu chúng thỏa mãn điều kiện đồng nhất. Thực hiện tiếp tục khi không thể tiếp tục nối các miền lại với nhau nữa. Trong cách này, hai vùng sẽ được nối lại với nhau nếu chúng thỏa mãn 2 điều kiện :

- Kế cận nhau
- Có mức xám đồng nhất

Để xác định tính kế cận giữa 2 vùng người ta sử dụng tính liên thông. Có 2 quan niệm

về liên thông là 4 liên thông và 8 liên thông. Với 4 liên thông, một điểm ảnh sẽ có 4 điểm ảnh kề cận theo 2 hướng  $x, y$ . Còn với 8 liên thông, một điểm ảnh sẽ có 4 điểm ảnh theo 2 hướng  $x, y$  là 4 điểm ảnh theo các hướng chéo 45 độ

### 2.2.3 Phân vùng ảnh dựa trên hợp và tách vùng

Để khắc phục được nhược điểm của 2 phương pháp cây tứ phân và phương pháp phát triển cục bộ người ta đưa ra phương pháp kết hợp ý tưởng hợp và tách vùng của 2 phương pháp trên. Với phương pháp tách, việc thực hiện chia quá chi tiết còn với phương pháp hợp mặc dù giảm được tối thiểu số vùng sau khi chia tuy nhiên không cho ta thấy rõ được mối liên hệ giữa các miền. Phương pháp kết hợp giữa hợp và tách thực hiện như sau. Đầu tiên, ta thực hiện tách miền ảnh theo cây tứ phân, thực hiện phân đoạn từ gốc đến lá, tiếp theo tiến hành duyệt cây theo chiều ngược lại để hợp các vùng đồng nhất. Trong thao tác hợp các vùng đồng nhất, có thể có nhiều vùng thỏa mãn điều kiện đồng nhất. Vì vậy ta phải xây dựng một hàm đánh giá giá trị đồng nhất của các vùng với nhau trả về các giá trị trong đoạn  $[0, 1]$  trong đó 0 là không đồng nhất, và 1 là hoàn toàn đồng nhất. Với hàm đánh giá kiểu này, trong quá trình thực hiện thao tác hợp, trong trường hợp có nhiều vùng thỏa mãn điều kiện đồng nhất, ta sẽ thực hiện chọn vùng mà giá trị hàm đồng nhất trả về lớn nhất để thực hiện hợp vùng.

## 2.3 Phân vùng ảnh dựa trên phân tích kết cấu

### 2.4 Phân vùng ảnh dựa trên sự phân lớp điểm ảnh

Cho một ảnh với các điểm ảnh:  $p_i$  với  $i = [1, N * M]$ ,  $M, N$  lần lượt là chiều rộng và chiều cao của ảnh. Với mỗi điểm ảnh ta chọn một thuộc tính để phục vụ cho việc phân vùng. Giả sử giá trị thuộc tính là  $A(p_i)$  với điểm ảnh thứ  $i$ . Thuật toán thực hiện chia tập hợp các điểm ảnh thành vào cá lớp và từ các lớp này tiến hành xây dựng vùng tương ứng. Tiêu chí dùng để phân vùng là độ tương đồng giữa các điểm ảnh trên thuộc tính phân

vùng được chọn. Ta hoàn toàn có thể sử dụng thuộc tính phân vùng nhiều chiều tùy vào yêu cầu của việc phân vùng. Ta có ví dụ về việc phân vùng với thuộc tính được chọn là một chiều, cụ thể là mức xám của điểm ảnh:

- Bước 1: Khởi tạo  $t = 0$ :

- Thực hiện phân tích lược đồ xám để dự đoán số lớp cần phân
- Giả sử số lớp chọn được là  $K$ , ta khởi tạo các ngưỡng đều nhau. Cụ thể là:

$$T_i(t) = i \frac{l_{max} - l_{min}}{K} + l_{min}; i = 0, \dots, K. \quad (2.2)$$

Với  $l_{max}, l_{min}$  lần lượt là giá trị mức xám lớn nhất và giá trị mức xám nhỏ nhất của các điểm ảnh trong ảnh.

- Bước 2: Thực hiện bước lặp theo thời gian:

- Phân lớp theo các ngưỡng  $T_i(k-1)$  với tiêu chí

$$p_i \in C_k \text{ nu } T_{k-1}(t-1) < A(p_i) < T_k(t-1), i = 1 \dots K \quad (2.3)$$

- Tính giá trị trung bình mức xám của các lớp

$$m_k(t-1) = \text{avarage}_{p_i \in C_k} A(p_i) \quad (2.4)$$

- Tính lại giá trị ngưỡng dựa trên giá trị trung bình mức xám của các lớp như sau:

$$T_i(t) = \frac{m_{i+1}(t-1) - m_i(t)}{2} \quad (2.5)$$

- Kiểm tra điều kiện dừng Nếu  $T_i(t) \approx T_i(t-1)$  hoặc  $m_i(t-1) \approx m_i(t)$  Thì dừng

- Thực hiện xây dựng vùng theo các lớp vừa phân

## 2.5 Phân vùng ảnh dựa vào lý thuyết đồ thị

Đây là một hướng tiếp cận mới được đưa ra gần đây. Với cách tiếp cận này, hình ảnh sẽ được biểu diễn bởi một đồ thị, trong đó các điểm ảnh là các đỉnh của đồ thị còn các cạnh trên đồ thị sẽ nối các điểm lân cận với nhau. Ngoài việc sử dụng cách phân vùng thông qua lựa chọn cạnh còn có thể phân vùng dựa trên mức độ khác nhau trong nội tại vùng và giữa các vùng với nhau.

### 2.5.1 Cách biểu diễn ảnh dưới dạng đồ thị

Ảnh được biểu diễn như là một đồ thị  $G = (V, E)$ . Trong đó  $V$  là tập hợp các đỉnh,  $E$  là tập hợp các cạnh của đồ thị. Mỗi đỉnh  $v_i \in V$  đại diện cho một điểm ảnh. Mỗi cạnh  $(v_i, v_j) \in E$  tương ứng với cạnh kết nối hai cặp điểm lân cận. Mỗi cạnh  $(v_i, v_j)$  sẽ có một trọng số tương ứng với độ chênh lệch về màu sắc hay mức xám giữa hai điểm ảnh tương ứng với hai đỉnh  $v_i, v_j$ . Trọng số này được ký hiệu là  $w(v_i, v_j)$

- Đỉnh kề: Hai đỉnh  $v_i, v_j$  trong đồ thị được gọi là kề nhau nếu  $(v_i, v_j)$  là cạnh của đồ thị. Bậc của một đỉnh  $deg(v)$  là số cạnh nối với đỉnh  $v$
- Đồ thị con và đồ thị riêng: Đồ thị  $G' = (V', E')$  là đồ thị con của đồ thị  $G = (V, E)$  nếu  $V' \subseteq V$  và  $E' = E \cap (V', V')$  Đồ thị  $G'' = (V, E')$  được gọi là đồ thị riêng của đồ thị  $G = (V, E)$  nếu  $E' \subseteq E$
- Cây bao trùm tối thiểu: Một cây được gọi là cây bao trùm tối thiểu của đồ thị  $G$  nếu nó chứa tất cả các đỉnh của  $G$  và trọng số nối mỗi đỉnh với một đỉnh bất kỳ là cạnh có trọng số nhỏ nhất trong các cạnh nối với đỉnh đó trong  $G$ .
- Vùng ảnh trên đồ thị: Với một phân vùng  $S$ , một vùng  $C \in S$  tương ứng với một thành phần liên thông  $G' = (V', E')$  trong đó  $V' \subseteq V, E' \subseteq E$  sao cho trong một vùng, trọng số nối hai cạnh bất kỳ nhỏ hơn trọng số ứng với cạnh nối hai vùng có giá trị lớn.



### 2.5.2 Các định nghĩa và tính chất liên quan

Để có được việc phân vùng chúng ta cần xây dựng một tính chất để biết được có hay không một ranh giới giữa hai vùng trong một phân vùng. Tính chất này được xây dựng thông qua việc so sánh sự khác nhau giữa các điểm ảnh trong cùng một vùng và sự khác nhau giữa các điểm ảnh giữa hai vùng dọc theo ranh giới giữa hai vùng. Sự khác nhau trong nội tại vùng được định nghĩa là giá trị trọng số lớn nhất của cây bao trùm tối thiểu ứng với phần đồ thị tương ứng với vùng cần xét:

$$Int(C) = \max_{e \in MST(C, E)} (w(e)) \quad (2.6)$$

Sự khác nhau giữa các vùng được định nghĩa là trọng số nhỏ nhất của cạnh kết nối hai đỉnh đại diện cho hai vùng:

$$Dif(C_1, C_2) = \min_{v_1 \in C_1, v_2 \in C_2, (v_1, v_2) \in E} (w(v_i, v_j)) \quad (2.7)$$

Trong trường hợp không có cạnh nối hai vùng thì  $Dif(C_1, C_2) = \infty$ . Giữa hai vùng sẽ có ranh giới nếu giá trị khác biệt giữa hai vùng lớn hơn giá trị khác biệt trong nội tại trong ít nhất 1 vùng. Chúng ta định nghĩa sự khác biệt nội bộ tối thiểu của hai vùng  $C_1, C_2$  được định nghĩa như sau:

$$Mint(C_1, C_2) = \min(Int(C_1) + T(C_1), Int(C_2) + T(C_2)) \quad (2.8)$$

với  $T = \frac{k}{|C|}$ ,  $k$  là hằng số,  $|C|$  là kích thước của vùng  $C$ . Lúc này ta có tính chất so sánh  $C_1, C_2$  được định nghĩa như sau:

$$D(C_1, C_2) = \begin{cases} 1, & \text{if } Dif(C_1, C_2) > Mint(C_1, C_2) \\ 0, & \text{if otherwise} \end{cases} \quad (2.9)$$

Trong thực tế  $k$  không cố định và được chọn tùy vào ý kiến chuyên gia. Trong trường hợp  $k$  lớn thì kích thước các vùng đầu ra sẽ lớn, ngược lại kích thước đầu ra sẽ nhỏ. Giá trị của  $k$  sẽ không nhỏ hơn kích thước của vùng nhỏ nhất

## 2.6 Phân vùng ảnh dựa trên biểu diễn và xử lý đa phân giải

## Chương 3

# Phương pháp phân vùng ảnh sử dụng biên động

Mô hình biên động là một mô hình thành công nhất trong phân vùng ảnh. Các mô hình biên động hiện tại đang có thể chia làm hai loại chính là: mô hình dựa trên cạnh (edge-based models), và mô hình dựa trên vùng (region-based models). Cả hai loại này đều có ưu, nhược điểm riêng và việc lựa chọn cách nào còn tùy thuộc vào ứng dụng

Ý tưởng của các phương pháp theo hướng tiếp cận này là xuất phát từ đường cong ban đầu, ta thực hiện biến đổi đường biên ban đầu theo thời gian và dừng lại ở biên của vật. Việc biến đổi đường biên được thực hiện thông qua xây dựng một vòng lặp nhằm đạt được giá trị cuối cùng của đường cong thỏa mãn cực tiểu hoặc cực đại một hàm năng lượng (energy function) nào đó.

Mô hình dựa trên cạnh sử dụng gradient ảnh để dừng đường biên tại biên của vật. Mô hình dựa trên cạnh thường gồm hai thành phần dừng dựa trên cạnh và thành phần lực cầu để kiểm soát chuyển động của đường biên. Thành phần dừng phục vụ cho việc dừng đường biên tại biên mong muốn. Thành phần lực cầu thực hiện co, mở rộng cạnh. Do vậy đường biên khởi tạo ban đầu có thể ở xa biên mong muốn. Khó khăn của mô hình này là việc lựa chọn lực cầu cho phù hợp. Nếu chọn lực cầu quá nhỏ, có thể quá trình phát triển biên sẽ không thể vượt qua một số vùng bằng phẳng của đối tượng. Ngược lại nó có thể

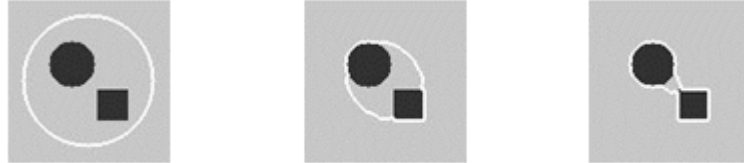
bỏ qua một số vùng biên yếu của vật.

Mô hình dựa trên miền có một số lợi thế so với mô hình dựa trên cạnh. Đầu tiên là việc mô hình dựa trên miền không sử dụng gradient ảnh do đó nó có hiệu suất tốt hơn đối với ảnh có biên yếu. Tiếp theo, nó ít ảnh hưởng bởi đường cong nhiễu tạo ban đầu hơn so với mô hình dựa trên cạnh. Phần dưới đây sẽ là ba mô hình biên động dựa trên miền điển hình và một mô hình cải tiến kết hợp các lợi thế của các mô hình trên để giải quyết tốt hơn với các đầu vào đa dạng.

## 3.1 Phương pháp Chan-Vese(CV)

### 3.1.1 Mô hình

Chúng ta định nghĩa đường cong  $C \in \Omega$  là biên của một tập con mở  $\omega$  của  $\Omega$ . Giả sử rằng ảnh được chia làm hai vùng được xấp xỉ bởi các hằng số cường độ khác nhau  $u_0^i$  và  $u_0^o$ . Đối tượng cần xác định trong ảnh được đại diện bởi vùng có cường độ  $u_0^i$ . Giả sử biên của nó là  $C_0$ . Vì vậy chúng ta có  $u_0 \approx u_0^i$  trong đối tượng, và  $u_0 \approx u_0^o$ .



Mô hình bao gồm 2 phần chính Fitting term và Regularization term:

- Fitting term: Đặt

$$F_1 = \iint_{inside(C)} |u_0 - c_1|^2 dx dy \quad (3.1)$$

$$F_2 = \iint_{outside(C)} |u_0 - c_2|^2 dx dy \quad (3.2)$$

Trong đó

$$c_1 = \text{average}(u_0 \text{inside}(C)) \quad (3.3)$$

$$c_2 = \text{average}(u_0 \text{outside}(C)) \quad (3.4)$$

Trong trường hợp đường cong  $C$  nằm ngoài đối tượng, ta có  $F_1 > 0, F_2 \approx 0$ . Nếu đường cong  $C$  nằm trong đối tượng ta có  $F_1 \approx 0, F_2 > 0$ . Nếu đường cong  $C$  một phần nằm trong và một phần nằm ngoài đối tượng ta có  $F_1 > 0, F_2 > 0$ . Nếu đường cong  $C$  khớp với đối tượng  $F_1 \approx 0, F_2 \approx 0$ :



- Regularization term:

$$\mu \cdot |C| + \nu \cdot \text{Area}(\text{inside}(C)) \quad (3.5)$$

Trong đó  $C$  là độ dài đường cong  $C$ ,  $\text{Area}(\text{inside}(C))$  là độ lớn phần diện tích giới hạn bởi đường cong. Trong hầu hết bài toán ta chọn  $\nu = 0$  và  $\mu$  là tham số thay đổi tùy theo đầu vào để được kết quả tốt. Tham số  $\mu$  được chọn càng nhỏ thì mô hình càng có khả năng phát hiện các đối tượng nhỏ. Trong trường hợp chỉ cần phát hiện các đối tượng có kích thước lớn,  $\mu$  được chọn lớn hơn.

Tổng kết lại ta có hàm năng lượng

$$\begin{aligned} \inf_{c_1, c_2, C} F(c_1, c_2, C) = & \mu \cdot |C| + \nu \cdot \text{Area}(\text{inside}(C)) \\ & + \lambda_1 \cdot \iint_{\text{inside}(C)} |u_0 - c_1|^2 dx dy + \lambda_2 \cdot \iint_{\text{outside}(C)} |u_0 - c_2|^2 dx dy \end{aligned} \quad (3.6)$$

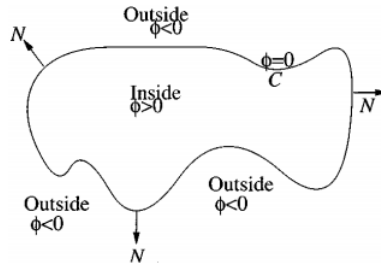
Bài toán cực tiểu đặt ra là

$$\inf_{c_1, c_2, C} F(c_1, c_2, C) \quad (3.7)$$

### 3.1.2 Giải quyết vấn đề sử dụng phương pháp tập mức

Ta coi biên  $C$  được đại diện bởi tập mức 0 của một hàm  $\phi : \omega \rightarrow \mathbb{R}$ . Đặt

$$\begin{cases} C = \omega = (x, y) \in \Omega : \phi(x, y) = 0 \\ \text{inside}(C) = \partial\omega = (x, y) \in \Omega : \phi(x, y) > 0 \\ \text{outside}(C) = \Omega\omega = (x, y) \in \Omega : \phi(x, y) < 0 \end{cases} \quad (3.8)$$



**Hình 3.1:** Đường cong  $C = \{(x, y), \phi(x, y) = 0\}$

Thay biến chưa biết  $C$  bởi biến chưa biết  $\phi$ . Trong quá trình xây dựng lại phương trình theo  $\phi$  ta sử dụng hàm Heviside  $\phi$  và Hàm Delta Dirac  $\delta_0$

$$H(z) = \begin{cases} 1 & \text{nếu } z > 0 \\ 0 & \text{nếu } z < 0 \end{cases} \quad (3.9)$$

Hàm năng lượng sẽ được viết lại thành

$$\begin{aligned} \inf_{c_1, c_2, \phi} F(c_1, c_2, C) &= \mu \int_{\Omega} \delta(x, y) |\nabla \phi(x, y)| \, dx \, dy + \nu \int_{\Omega} H(\phi(x, y)) \, dx \, dy \\ &+ \lambda_1 \cdot \int_{\Omega} |u_0 - c_1|^2 H(\phi(x, y)) \, dx \, dy \\ &+ \lambda_2 \cdot \iint_{\Omega} |u_0 - c_2|^2 (1 - H(\phi(x, y))) \, dx \, dy \end{aligned} \quad (3.10)$$

Trong đó  $c_1, c_2$  có thể được tính theo  $\phi$  theo công thức:

$$c_1(\phi) = \frac{\int_{\Omega} u_0(x, y) H(\phi(x, y)) dx dy}{\int_{\Omega} H(\phi(x, y)) dx dy} \quad (3.11)$$

$$c_2(\phi) = \frac{\int_{\Omega} u_0(x, y) (1 - H(\phi(x, y))) dx dy}{\int_{\Omega} H(\phi(x, y)) (1 - H(\phi(x, y))) dx dy} \quad (3.12)$$

Để tính toán phương trình Euler-Lagrange cho hàm  $\phi$  ta thực hiện chuẩn hóa  $H$  và  $\delta_0$  tương ứng thành  $H_\epsilon$  và  $\delta_\epsilon$ ,  $\epsilon \rightarrow 0$ . Hàm mở rộng  $H_\epsilon$  có thể chọn là hàm  $H_{1,\epsilon}$  trên  $C^2(\bar{\Omega})$  của  $H$  và  $\delta_\epsilon = H'_\epsilon$ :

$$H_{1,\epsilon}(z) = \begin{cases} 1 & \text{nếu } z > \epsilon \\ 0 & \text{nếu } z < -\epsilon \\ \frac{1}{2} \left( 1 + \frac{z}{\epsilon} + \frac{1}{\pi} \sin\left(\frac{\pi z}{\epsilon}\right) \right) & \text{nếu } -\epsilon < z < \epsilon \end{cases} \quad (3.13)$$

Ngoài ra ta có thể sử dụng hàm mở rộng của  $H$  trên  $C^\infty(\bar{\Omega})$ :

$$H_{2,\epsilon}(z) = \frac{1}{2} \left( 1 + \frac{z}{\epsilon} + \frac{1}{\pi} \sin\left(\frac{\pi z}{\epsilon}\right) \right) \text{nếu } -\epsilon < z < \epsilon \quad (3.14)$$

Cả hai hàm  $H_{1,\epsilon}(z), H_{2,\epsilon}(z)$  đều tiến dần đến  $H$  và  $\delta_{1,\epsilon}(z), \delta_{2,\epsilon}(z)$  đều tiến dần đến  $\delta_0$  khi  $\epsilon \rightarrow 0$  khi  $\epsilon \rightarrow 0$ . Tuy vậy điều khác nhau là  $\delta_{1,\epsilon}$  hỗ trợ ít hơn, cụ thể là trong đoạn  $[-\epsilon, \epsilon]$  còn  $\delta_{2,\epsilon}$  lại khác 0 tại mọi điểm. Do hàm năng lượng đã xây dựng bên trên không lồi do vậy có thể có cực trị địa phương, và kết quả có thể phụ thuộc vào đường cong khởi tạo ban đầu. Với  $H_{1,\epsilon}(z), \delta_{1,\epsilon}(z)$  thuật toán có thể đưa ra kết quả là cực trị địa phương, trong khi với  $H_{2,\epsilon}(z), \delta_{2,\epsilon}(z)$  thuật toán có xu hướng đưa ra kết quả là cực trị toàn cục. Hàm mở rộng của  $F(c_1, c_2, C)$  thành

$$\begin{aligned} F(c_1, c_2, \phi) &= \mu \int_{\Omega} \delta_0(x, y) |\nabla \phi(x, y)| dx dy + \nu \int_{\Omega} H(\phi(x, y)) dx dy \\ &+ \lambda_1 \cdot \int_{\Omega} |u_0 - c_1|^2 H(\phi(x, y)) dx dy \\ &+ \lambda_2 \cdot \int_{\Omega} |u_0 - c_2|^2 (1 - H(\phi(x, y))) dx dy \end{aligned} \quad (3.15)$$

Cố định  $c_1, c_2$ , thực hiện cực tiểu hàm  $F_\epsilon$  theo  $\phi$ , ta thu được phương trình Euler-Lagrange theo  $\phi$ . Tham số hóa  $\phi$  theo thời gian và áp dụng phương pháp hướng giảm ta được:

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) [\mu - \text{div}\left(\frac{\nabla \phi}{|\nabla \phi|}\right) - \nu - \lambda_1 (u_0 - c_1)^2 - \lambda_2 (u_0 - c_2)^2] \text{trong } (0, \infty) \times \Omega \quad (3.16)$$

$$\phi(0, x, y) = \phi_0(x, y) \text{ trong } \Omega \quad (3.17)$$

$$\frac{\delta_\epsilon(\phi)}{|\nabla\phi|} \frac{\partial\phi}{\partial\vec{n}} = 0 \quad (3.18)$$

Trong đó  $\vec{n}$  là vector pháp tuyến hướng ra ngoài của biên  $\Omega$ , và  $\frac{\partial\phi}{\partial\vec{n}}$  là đạo hàm theo vector pháp tuyến của  $\phi$ . Trong quá trình tính toán xấp xỉ các giá trị  $\phi_x, \phi_y$  ở các điểm gần biên, ta cần biết được giá trị  $\phi(x, y)$  tại các điểm  $(x, y)$  nằm ngoài biên  $\partial\Omega$  trong khi ta chỉ có giá trị tại các điểm nằm trong và trên biên của  $\Omega$ , một cách tự nhiên người ta có thể chọn  $\phi(x, y)$  tại các điểm  $(x, y)$  nằm ngoài biên bằng với giá trị của  $\phi(x, y)$  tại điểm gần  $(x, y)$  nhất trên biên, hay  $\frac{\partial\phi}{\partial\vec{n}} = 0$ . Thuật toán theo mô hình này được mô tả cụ thể như sau: Khi làm việc với hàm Level set, thủ tục cần thiết là khởi tạo lại hàm  $\phi$  bằng hàm dấu khoảng cách ứng với  $\phi$ . Điều này giúp ngăn việc hàm  $\phi$  ko trở nên quá lớn dẫn đến việc thuật toán không hội tụ.

- Bước 1: Khởi tạo  $\phi^0 = \phi_0, n = 0$
- Bước 2: Tính toán các giá trị  $c_1(\phi^n), c_2(\phi^n)$
- Bước 3: Giải phương trình đạo hàm riêng (1.14) để có  $\phi^{n+1}$
- Bước 4: Khởi tạo lại hàm  $\phi$  bằng hàm dấu khoảng cách tới đường cong
- Bước 5: Kiểm tra điều kiện dừng, nếu không thỏa mãn thì gán  $n = n + 1$  và quay lại bước 2

Các tính toán xấp xỉ được dùng trong quá trình giải số hệ phương trình đạo hàm riêng



(1.14) như sau:

$$\begin{aligned}
k &= \operatorname{div}\left(\frac{\nabla\phi}{|\phi|}\right) = \frac{\phi_{xx}\phi_y^2 - 2\phi_{xy}\phi_x\phi_y + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}} \\
\phi_x &= \frac{1}{2h}(\phi_{i+1,j} - \phi_{i-1,j}) \\
\phi_y &= \frac{1}{2h}(\phi_{i,j+1} - \phi_{i,j-1}) \\
\phi_{xx} &= \frac{1}{h^2}(\phi_{i+1,j} - \phi_{i,j} + \phi_{i-1,j}) \\
\phi_{yy} &= \frac{1}{h^2}(\phi_{i,j+1} - \phi_{i,j} + \phi_{i,j-1}) \\
\phi_{xy} &= \frac{1}{h^2}(\phi_{i+1,j+1} - \phi_{i-1,j+1} - \phi_{i,j+1} + \phi_{i,j-1})
\end{aligned} \tag{3.19}$$

trong đó  $h$  là khoảng cách lưới. Khi đó (7) có thể được rời rạc hóa thành thành

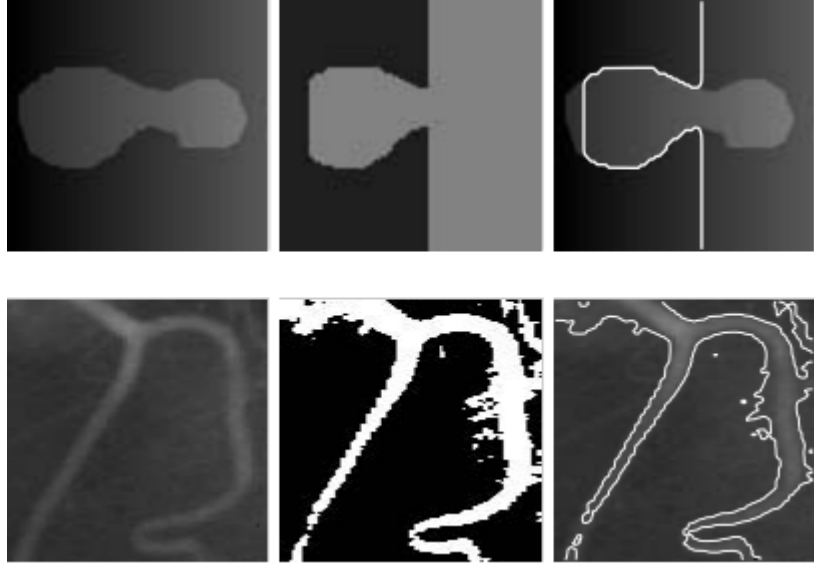
$$\frac{\phi_{n+1} - \phi_n}{\Delta t} = \delta_\epsilon(\phi_{n+1})[\mu k - \lambda_1(u_{i,j} - c_1(\phi^n))^2 + \lambda_2(u_{i,j} - c_2(\phi^n))^2] \tag{3.20}$$

Từ đó ta có thể tính được  $\phi^{n+1}$  theo  $\phi^n$  Chú ý là thành phần  $|C|$  có thể viết lại dưới dạng tổng quát hơn thành  $|C|^p$ . Nếu ta xét đến trường hợp số chiều  $N > 1 (\Omega \in \mathbb{R}^N)$ , khi đó ta có thể chọn  $p = 1$  hoặc  $p = \frac{N}{N-1}$

### 3.1.3 Ưu nhược điểm của mô hình

Ưu điểm của mô hình là có thể phát hiện được biên trơn, xử lý tốt với ảnh nhiễu, mô hình có thể phát hiện được biên từ duy nhất một đường cong khởi tạo ban đầu. Đường cong này có thể ở mọi vị trí trên ảnh mà không nhất thiết phải bao quanh đối tượng trong ảnh.

Nhược điểm của mô hình là độ phức tạp tính toán còn lớn do thao tác re-initialization và kết quả còn thiếu chính xác trong trường hợp ảnh không đồng nhất cường độ. Điều này có thể giải thích là do việc coi giá trị mức xám của các điểm trong đối tượng, ngoài đối tượng là đồng nhất, trong khi sự chênh lệch giữa mức xám giữa các điểm ảnh lại lớn trong trường hợp ảnh không đồng nhất cường độ.



**Hình 3.2:** Kết quả của mô hình C-V trong trường hợp ảnh không đồng nhất

## 3.2 Phương pháp Local binary fitting energy (LBF)

Phần trên đã đề cập đến việc mô hình CV tỏ ra không hiệu quả đối với ảnh không đồng nhất cường độ do mô hình CV chỉ sử dụng thông tin toàn cục của ảnh. Mô hình LBF được tạo ra để giải quyết vấn đề này. Mô hình LBF sử dụng thêm thông tin địa phương nhờ việc sử dụng cửa sổ Gauss thông qua sử dụng hàm nhân Gauss.

### 3.2.1 Mô hình

Cho ảnh  $I : \Omega \rightarrow \mathbb{R}^d$ . Trong đó  $\Omega \subset \mathbb{R}^2$  là miền ảnh,  $d > 1$  là bậc của vector  $I(x)$ . Với ảnh xám  $d=1$ , với ảnh màu  $d=3$ . Gọi  $C$  là biên của ảnh trong  $\Omega$ , với mỗi  $x$  ta định nghĩa một hàm năng lượng sau:

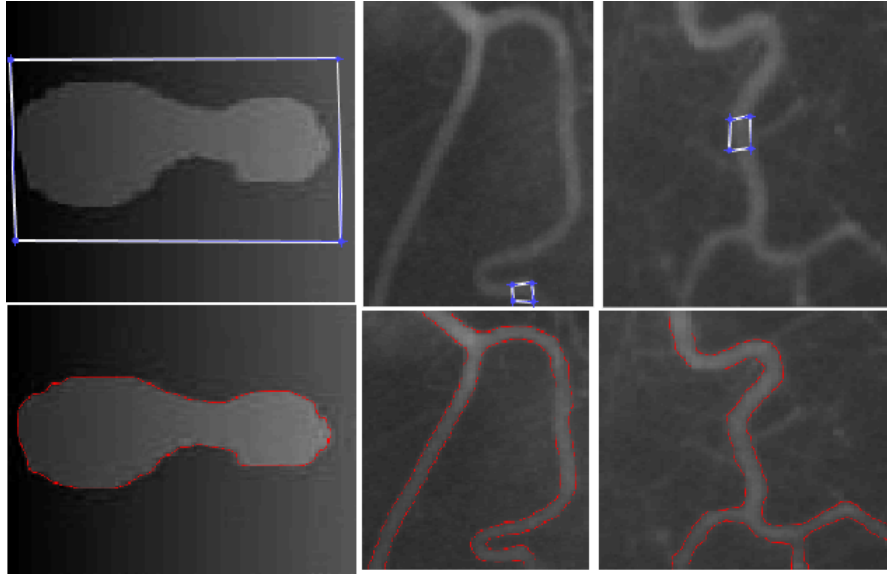
$$\epsilon_x^{LBF}(C, f_1(x), f_2(x)) = \lambda_1 \int_{in(C)} K(x-y) |I(y) - f_1(x)|^2 dy + \lambda_2 \int_{out(C)} K(x-y) |I(y) - f_2(x)|^2 dy \quad (3.21)$$

trong đó  $\lambda_1, \lambda_2$  là các hằng số dương,  $K$  là hàm nhân với thuộc tính địa phương  $K(u)$  giảm và dần về 0 khi  $|u|$  giảm,  $f_1(x), f_2(x)$  là hai số khớp với cường độ ảnh tại các điểm

gần  $x$ . Ta gọi  $x$  là điểm trung tâm của hàm năng lượng trên. Trong mô hình này  $K$  được chọn là hàm nhân Gaussian

$$K_\sigma(x) = \frac{1}{(2\pi)^{n/2}\sigma^n} e^{-|x|^2/2\sigma^2} \quad (3.22)$$

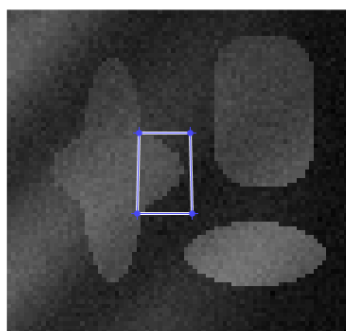
với  $\sigma$  là tham số dương có thể tùy chỉnh được. Nhấn mạnh rằng  $f_1(x), f_2(x)$  thay đổi theo  $x$ .  $f_1(x), f_2(x)$  làm phương pháp này trở nên khác các phương pháp khác.



**Hình 3.3:** Kết quả của mô hình LBF trong trường hợp ảnh không đồng nhất

Trong mô hình này, hàm năng lượng có tính đại phương với  $x$  hay  $f_1(x), f_2(x)$  chỉ khớp với cường độ ảnh tại các điểm gần  $x$ . Điều này có được là do tính chất của hàm  $K$  là  $K(x - y)$  có giá trị lớn hơn khi  $y$  gần  $x$ . Vì vậy cường độ ảnh tại các điểm  $y$  gần  $x$  ảnh hưởng nhiều hơn đến giá trị  $f_1, f_2$  làm cực tiểu hàm năng lượng  $\epsilon_x^{LBF}(C, f_1, f_2)$ , trong khi cường độ ảnh tại các điểm  $y$  xa  $x$  hầu như không ảnh hưởng tới các giá trị  $f_1, f_2$

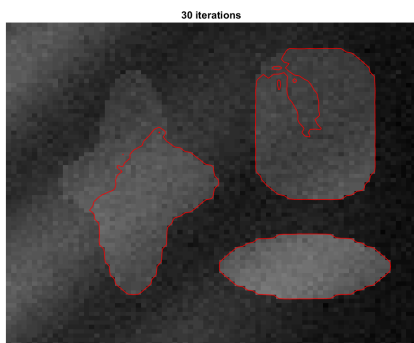
Tại mỗi điểm trung tâm  $x$ , hàm năng lượng  $\epsilon_x^{LBF}(C, f_1, f_2)$  có thể đạt cực tiểu khi đường con  $C$  khớp với biên của đối tượng và các giá trị  $f_1, f_2$  được chọn sao tối ưu. Tuy nhiên hàm năng lượng  $\epsilon_x^{LBF}(C, f_1, f_2)$  được định nghĩa một cách cục bộ theo điểm trung tâm  $x$ . Để tìm được toàn bộ biên của vật, ta cần cực tiểu hàm  $\epsilon_x^{LBF}$  với mọi  $x$  trong miền



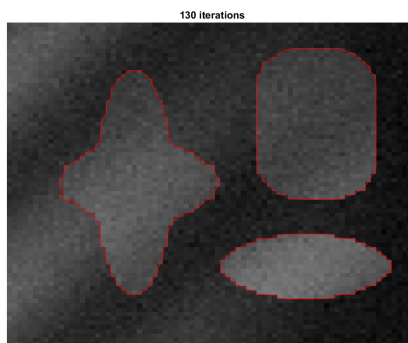
(a) Đường cong khởi tạo



(b) Bước lặp 10



(c) Bước lặp 30



(d) Bước lặp 130

**Hình 3.4:** Kết quả LBF với ảnh nhiễu

ảnh  $\Omega$ . Điều này có thể làm được bằng cách định nghĩa hàm năng lượng mới sau

$$\epsilon(C, f_1, f_2) = \int_{\Omega} \epsilon_x^{LBF}(C, f_1(x), f_2(x)) dx \quad (3.23)$$

### 3.2.2 Giải quyết vấn đề sử dụng phương pháp tập mức

Đường cong  $C \subset \Omega$  được đại diện bởi tập mức 0 của một hàm Lipschit  $\phi \rightarrow \mathbb{R}$ . Viết lại hàm năng lượng  $\epsilon_x^{LBF}(C, f_1, f_2)$  theo  $\phi$  ta được

$$\begin{aligned} \epsilon_x^{LBF}(\phi, f_1(x), f_2(x)) = & \lambda_1 \int_{\Omega} K_{\sigma}(x-y) |I(y) - f_1(x)|^2 H(\phi(y)) dy \\ & + \lambda_2 \int_{\Omega} K_{\sigma}(x-y) |I(y) - f_2(x)|^2 (1 - H(\phi(y))) dy \end{aligned} \quad (3.24)$$

trong đó  $H$  là hàm Heaviside. Theo đó hàm năng lượng  $\epsilon^{LBF}$  có thể được viết lại thành

$$\begin{aligned} \epsilon(C, f_1, f_2) &= \int_{\Omega} \epsilon_x^{LBF}(C, f_1(x), f_2(x)) dx \\ &= \lambda_1 \int_{\Omega} \int_{\Omega} K_{\sigma}(x-y) |I(y) - f_1(x)|^2 H(\phi(y)) dy dx \\ &+ \lambda_2 \int_{\Omega} \int_{\Omega} K_{\sigma}(x-y) |I(y) - f_2(x)|^2 (1 - H(\phi(y))) dy dx \end{aligned} \quad (3.25)$$

Để đảm bảo rằng  $\phi$  ổn định, chúng ta thêm hàm độ lệch giữa hàm tập mức  $\phi$  và hàm dấu khoảng cách.

$$P(\phi) = \int_{\Omega} \frac{1}{2} (|\nabla \phi| - 1)^2 dx \quad (3.26)$$

$$\mathcal{L}(\phi) = \int_{\Omega} \delta(\phi(x)) |\nabla \phi(x)| dx \quad (3.27)$$

Bây giờ ta có hàm energy cuối cùng

$$\mathcal{F}(\phi, f_1, f_2) = \epsilon(C, f_1, f_2) + \mu \mathcal{P}(\phi) + \nu \mathcal{L}(\phi) \quad (3.28)$$

trong đó  $\mu$  và  $\nu$  là các hằng số. Trong thực tế, các  $H$  được xấp xỉ bởi một hàm trơn  $H_{\epsilon}$

$$H_{\epsilon}(x) = \frac{1}{2} \left[ 1 + \frac{2}{\pi} \arctan\left(\frac{x}{\epsilon}\right) \right] \quad (3.29)$$

và  $\delta_{\epsilon}(x)$  được chọn là

$$\delta_{\epsilon}(x) = H'_{\epsilon}(x) = \frac{1}{\pi} \frac{\epsilon}{\epsilon^2 + x^2} \quad (3.30)$$

Hàm năng lượng này sẽ được cực tiểu hóa để tìm biên Với  $\phi$  cố định ta thực hiện cực tiểu hàm năng lượng  $\epsilon(C, f_1, f_2)$  theo  $f_1, f_2$  ta tìm được  $f_1, f_2$  như sau:

$$f_1(x) = \frac{K_\sigma * [(H_\epsilon(\phi(x))) * I(x)]}{K_\sigma * (H_\epsilon(\phi(x)))} \quad (3.31)$$

$$f_1(x) = \frac{K_\sigma * [(1 - H_\epsilon(\phi(x))) * I(x)]}{K_\sigma * (1 - H_\epsilon(\phi(x)))} \quad (3.32)$$

Theo công thức  $f_1, f_2$  ở trên luôn dương do  $H_\epsilon$  và  $1 - H_\epsilon$  luôn dương. Cố định  $f_1, f_2$  và cực tiểu hàm  $\epsilon^{LBF}(\phi, f_1(x), f_2(x))$  theo  $\phi$ , sử dụng phương pháp hướng giảm ta được:

$$\begin{aligned} \frac{\partial \phi}{\partial t} = & -\delta_\epsilon(\phi)(\lambda_1 e_1 - \lambda_2 e_2) + \nu \delta_\epsilon(\phi) \operatorname{div}\left(\frac{\nabla \phi}{|\nabla \phi|}\right) \\ & + \mu(\nabla^2 \phi - \operatorname{div}\left(\frac{\nabla \phi}{|\nabla \phi|}\right)) \end{aligned} \quad (3.33)$$

trong đó  $\delta_\epsilon(x)$  là hàm trơn Dirac được cho như công thức trên và  $e_1(x), e_2(x)$  được tính theo công thức sau:

$$e_1(x) = \int_{\Omega} K_\sigma(y - x) |I(x) - f_1(y)|^2 dy \quad (3.34)$$

$$e_2(x) = \int_{\Omega} K_\sigma(y - x) |I(x) - f_2(y)|^2 dy \quad (3.35)$$

Ta sẽ có thuật toán như sau:

- Bước 1: Khởi tạo  $\phi^0 = \phi_0, n = 0$
- Bước 2: Tính toán các giá trị  $c_1(\phi^n), c_2(\phi^n)$
- Bước 3: Giải phương trình đạo hàm riêng (1.14) để có  $\phi^{n+1}$
- Bước 4: Kiểm tra điều kiện dừng, nếu không thỏa mãn thì gán  $n = n + 1$  và quay lại bước 2

Với việc này

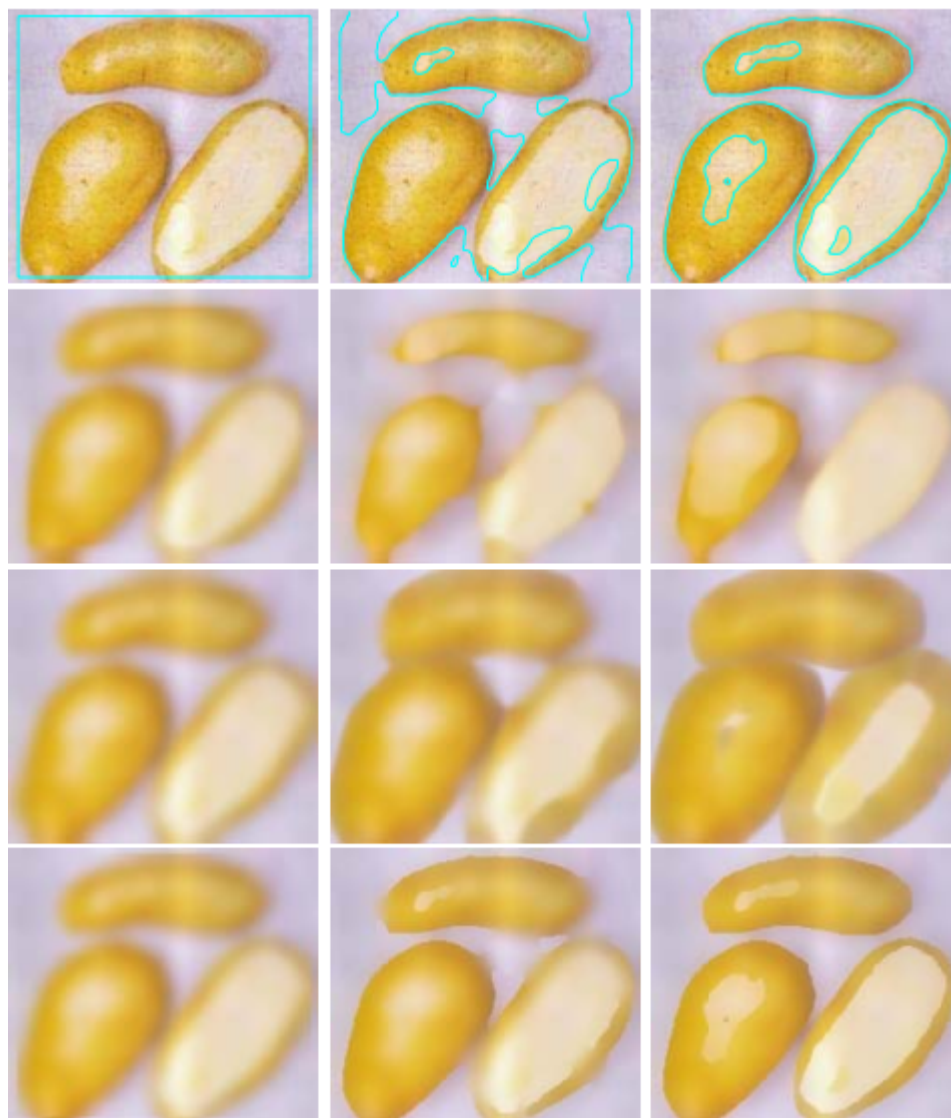
### 3.2.3 Ưu nhược điểm của mô hình

Ưu điểm của mô hình này là không cần thiết phải chuẩn hóa  $f_1, f_2$ . Thực tế,  $f_1, f_2$  làm cực tiểu hàm năng lượng có thể được cho bởi công thức (14)(15) và là các hàm trơn. Hơn nữa mô hình này cũng không cần phải mở rộng  $f_1, f_2$  vì nó được định nghĩa trên toàn miền ảnh  $\Omega$ . Một ưu điểm nữa của mô hình này là việc khởi tạo lại hàm  $\phi$  là không cần thiết do việc thêm hàm chuẩn hóa khoảng cách. Nhờ việc chuẩn hóa khoảng cách này mà việc khởi tạo hàm  $\phi$  lúc đầu trở nên linh hoạt hơn. Một trường hợp đặc biệt là ta hoàn toàn có thể khởi tạo hàm  $\phi$  là hàm nhị phân, nhận giá trị  $c_0$  ngoài miền  $R_0$ , nhận giá trị  $-c_0$  ở trong miền  $R_0$ .

Ngoài việc phục vụ cho việc phân vùng, mô hình LBF còn có thể sử dụng cho việc khử nhiễu. Hình dưới là kết quả của mô hình LBF với ảnh màu, giải thích cho sự kết hợp giữa phân vùng và khử nhiễu được sử dụng trong mô hình LBF. Dòng đầu tiên là kết quả của LBF từ đường cong khởi tạo cho đến kết quả cuối cùng sau khi phân vùng. Dòng 2 và dòng 3 là kết quả tương ứng với ảnh  $f_1, f_2$  được tính theo công thức (3.31)(3.32). Chúng ta định nghĩa một ảnh khớp sau dựa trên  $f_1, f_2, \phi$ :

$$f = H_\epsilon(\phi)f_1 + (1 - H_\epsilon(\phi))f_2 \quad (3.36)$$

$f$  có thể dùng sử dụng để xấp xỉ ảnh gốc trong quá trình khử nhiễu. Dòng thứ 4 trong hình (3.5) là kết quả của quá trình tiến hóa  $f$ . Khi hàm  $\phi$  hội tụ ta có  $f$  khớp với ảnh gốc và đồng thời khử được nhiễu của ảnh gốc ban đầu.



**Hình 3.5:** Khử nhiễu với mô hình LBF

Một nhược điểm của mô hình là quá trình tính toán  $\lambda_1 e_1 - \lambda_2 e_2$  còn tốn nhiều tài nguyên. Điều này có thể khắc phục nhờ việc dùng mô hình LIF sẽ được trình bày tiếp theo



### 3.3 Phương pháp Local image fitting(LIF)

Ở phần trước chúng ta đã thấy được mô hình LBF phân vùng hiệu quả với ảnh không đồng nhất cường độ, đồng thời hiệu suất của mô hình LBF cũng tốt hơn trong cả độ chính xác của kết quả cũng như tốc độ tính toán. Phần dưới đây sẽ trình bày về mô hình LIF với khả năng xử lý với ảnh không đồng nhất cường độ xám, sử dụng thông tin địa phương của ảnh để xây dựng hàm năng lượng dựa trên sự khác nhau của ảnh đã khớp so với ảnh ban đầu. Mô hình LIF thực hiện chuẩn hóa hàm tập mức bằng việc sử dụng hàm lọc Gauss sau mỗi vòng lặp. Điều này cũng giúp cho phương pháp này tránh khỏi quá trình khởi tạo lại hàm tập mức bằng dấu khoảng cách. Kết quả thực nghiệm đã cho thấy tốc độ khi thực hiện phân vùng với mô hình này tốt hơn so với mô hình LBF trong khi đưa ra một kết quả phân vùng tương đương.

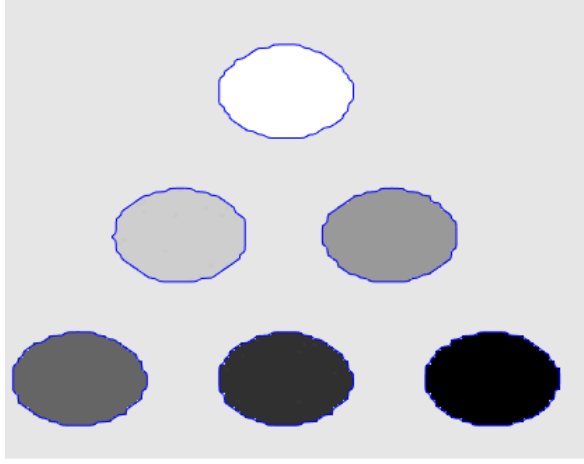
#### 3.3.1 Mô hình và phương trình với tập mức

Một hàm khớp địa phương (Local fitted image fomulation) được định nghĩa như sau:

$$I^{LFI} = m_1 H_\epsilon(\phi) + m_2 (1 - H_\epsilon(\phi)) \quad (3.37)$$

Trong đó  $m_1, m_2$  được định nghĩa như sau

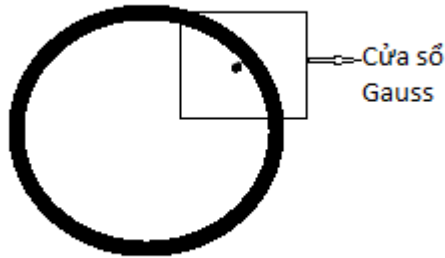
$$\begin{cases} m_1 = \text{mean}(I \in (\{x \in \Omega | \phi(x) < 0\} \cap W_k(x))) \\ m_2 = \text{mean}(I \in (\{x \in \Omega | \phi(x) > 0\} \cap W_k(x))) \end{cases} \quad (3.38)$$



**Hình 3.6:** Phân vùng với ảnh nhiều đối tượng có mức xám khác nhau

trong đó  $W_k(x)$  là một hàm cửa sổ hình chữ nhật. Ở đây ta chọn  $W_k(x)$  là hàm cửa sổ Gaussian  $K_\sigma(x)$  với độ lệch chuẩn  $\sigma$  và với kích thước  $4k + 1$  và  $4k + 1$  với  $k$  là số nguyên lớn nhất không vượt quá  $\sigma$ . Việc sử dụng cửa sổ này thể hiện tính chất địa phương của mô hình. Việc tính  $m_1, m_2$  chỉ thực hiện với vùng nằm trong cửa sổ mà không phải thực hiện trên toàn vùng như mô hình CV. Nhờ tính chất này mà mô hình LIF có thể xử lý được với ảnh không đồng nhất cường độ, ngoài ra còn có thể phân vùng với ảnh nhiều đối tượng với các cường độ xám khác nhau. Mô hình này sử dụng hàm năng lượng khớp địa phương thể hiện độ lệch giữa ảnh sau khi khớp và ảnh gốc:

$$E^{LIF}(\phi) = \frac{1}{2} \int_{\Omega} |I(x) - I^{LFI}(x)|^2 dx \quad (3.39)$$



**Hình 3.7:** Cửa sổ Gauss

Thêm thành phần biến thiên  $\eta$  vào hàm  $\phi$  ta được  $\bar{\phi} = \phi + \epsilon\eta$  Cố định  $c_1, c_2$ . Đạo hàm theo  $\phi$  và cho  $\epsilon$  tiến dần tới 0 ta có

$$\begin{aligned} \frac{\delta E^{LIF}}{\delta \phi} &= \lim_{\epsilon \rightarrow 0} \frac{d}{d\epsilon} \left( \frac{1}{2} \int_{\Omega} |I(x) - m_1 H_{\epsilon}(\bar{\phi}) - m_2 (1 - H_{\epsilon}(\bar{\phi}))|^2 dx \right) \\ &= \lim_{\epsilon \rightarrow 0} \frac{d}{d\epsilon} \left( - \int_{\Omega} -\delta_{\epsilon}(\phi) (I(x) - m_1 H_{\epsilon}(\bar{\phi}) - m_2 (1 - H_{\epsilon}(\bar{\phi}))) (m_1 - m_2) \eta dx \right) \quad (3.40) \\ &= - \left( \int_{\Omega} -\delta_{\epsilon}(\phi) (I(x) - m_1 H_{\epsilon}(\phi) - m_2 (1 - H_{\epsilon}(\phi))) (m_1 - m_2) \eta dx \right) \end{aligned}$$

Theo đó ta có phương trình Euler-Lagarange:

$$\delta_{\epsilon}(\phi) \{ (I - I^{LFI})(m_1 - m_2) \} = 0 \quad (3.41)$$

Sử dụng phương pháp hướng giảm ta có

$$\frac{\partial \phi}{\partial t} = (I - I^{LFI})(m_1 - m_2) \delta_{\epsilon}(\phi) \quad (3.42)$$

Trong quá trình làm việc theo phương pháp với tập mức, để đảm bảo tính hội tụ của  $\phi$ , ta cần thực hiện thao tác khởi tạo lại  $\phi$ . Tuy nhiên việc này đòi hỏi khá nhiều tài nguyên. Mô hình này thực hiện việc khởi tạo lại thành chuẩn hóa hàm  $\phi$  sử dụng hàm chuẩn hóa Gauss  $\phi = \phi * G_{\psi}$ , trong đó  $\psi$  là độ lệch chuẩn.

Các bước của thuật toán như sau:

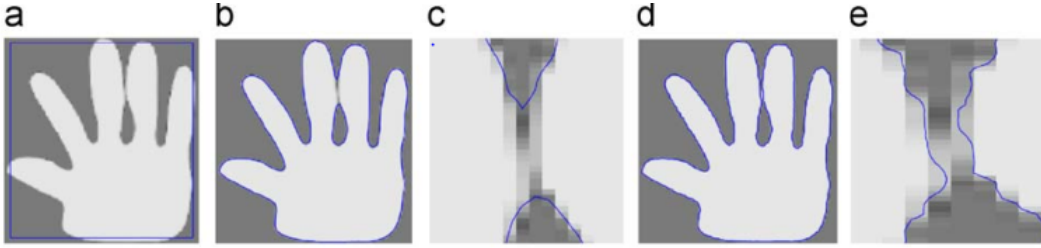
- Bước 1: Khởi tạo hàm  $\phi$  là hàm nhị phân như sau:

$$\phi(x, t = 0) = \begin{cases} -\rho, & x \in \Omega_0 - \partial\Omega_0 \\ 0, & x \in \partial\Omega_0 \\ \rho, & x \in \Omega - \Omega_0 \end{cases} \quad (3.43)$$

trong đó  $\rho > 0$  là hằng số,  $\Omega_0$  là tập con của miền ảnh  $\Omega$  và  $\partial\Omega_0$  là biên của  $\Omega_0$

- Bước 2: Tính toán  $\phi$  sử dụng phương trình (3.4)
- Bước 3: Chuẩn hóa hàm tập mức  $\phi$  bởi hàm nhân Gaussian. Ví dụ  $\phi = \mathcal{G}_{\psi} * \phi$  trong đó  $\psi$  là độ lệch chuẩn thỏa mãn lớn hơn  $\sqrt{t}$  để tăng cường khả năng làm mịn

**Hình 3.8:** Hình b và c thể hiện kết quả của phương pháp C-V, hình d và e thể hiện kết quả của phương pháp LIF



- Bước 4: Kiểm tra điều kiện dừng, nếu không thỏa mãn quay lại bước 2

Qua quá trình thử nghiệm, tham số  $\psi$  ở bước 3 thường được chọn trong khoảng 0.45 đến 1. Nếu nhiều lớn thì  $\psi$  nên chọn lớn hơn

### 3.3.2 Ưu nhược điểm của mô hình

So với mô hình LBF, tuy mô hình LBF không cần thao tác re-initialization không cần thiết, tuy nhiên độ phức tạp tính toán còn tương đối cao. Chi phí tính toán của mô hình LBF chủ yếu nằm ở thao tác tính toán  $\lambda_1 e_1 - \lambda_2 e_2$ . Mô hình LIF có tốc độ tính toán tốt hơn hẳn so với LBF. Đồng thời thực nghiệm cho thấy mô hình LIF có thể đạt tới việc phân chia nhỏ hơn pixel (Hình 3.6). Hình dưới cho thấy với phương pháp C-V kết quả cuối cùng thu được 2 ngón tay ở giữa bị dính vào nhau trong khi với phương pháp LIF thì chúng tách rời hoàn toàn.

## 3.4 Mô hình kết hợp Global và Local

### 3.4.1 Mô hình

Cả 3 phương pháp phân vùng sử dụng biên động như đã trình bày ở trên đều có ưu nhược điểm của riêng mình. Với mô hình CV, nó hoạt động tốt với các ảnh mà cường độ xám

trong và ngoài đối tượng trong ảnh đồng nhất, trong trường hợp ảnh có cường độ sáng không đồng nhất thì mô hình này thường cho ra kết quả sai. Ngược lại, mô hình LBF, LIF trình bày ở phần 2,3 lại hoạt động tốt với ảnh không đồng nhất mức xám và tỏ ra không hiệu quả với ảnh đồng nhất mức xám. Trong phần này tôi đề xuất một mô hình kết hợp mô hình Global và Local để có thể xử lý được với nhiều loại ảnh khác nhau, giảm thời gian tính toán, đồng thời tăng tính hội tụ do việc giảm bớt các thành phần dễ bị ảnh hưởng bởi đường biên ban đầu. Trong mô hình CV ta lấy thành phần fitting term và bỏ đi thành phần regularize term :

$$E^{GIF} = \int_{\Omega} |u_0 - c_1|^2 H(\phi(x, y)) + |u_0 - c_2|^2 (1 - H(\phi(x, y))) dx dy \quad (3.44)$$

Chúng ta gọi nó là GIF (Global image fitting). Ngoài ra trong mô hình này sử dụng thêm thành phần  $E^{LIF}$  trong mô hình LIF:

$$E^{LIF}(\phi) = \frac{1}{2} \int_{\Omega} |I(x) - m_1 H_{\epsilon}(\phi) - m_2 (1 - H_{\epsilon}(\phi))|^2 dx \quad (3.45)$$

trong đó

$$\phi(x, t = 0) = \begin{cases} -\rho, x \in \Omega_0 - \partial\Omega_0 \\ 0, x \in \partial\Omega_0 \\ \rho, x \in \Omega - \Omega_0 \end{cases} \quad (3.46)$$

Hàm năng lượng được đề xuất chứa thành phần global và thành phần local :

$$E^{GLIF} = \alpha E^{LIF} + (1 - \alpha) E^{GIF} \quad (3.47)$$

trong đó  $\alpha$  là số không âm nằm trong đoạn  $[0,1]$ . Khi xử lý với ảnh có cường độ ảnh đồng nhất ta sử dụng hệ số alpha cao. Trong khi xử lý với ảnh có cường độ không đồng nhất ta sử dụng hệ số alpha thấp. Thành phần LIF chứa lực địa phương (local force) để trích biên và dừng nó ở biên của vật. Thành phần này cho phép mô hình có thể xử lý với vấn đề cường độ không đồng nhất. Thành phần GIF chứa lực toàn cục (global force) cho phép dịch chuyển biên khi biên tạm ở xa biên của vật. Điều này cho phép mô hình xử lý linh động hơn với các đường cong khởi tạo ban đầu. Ảnh hưởng của các lực địa phương và

lực toàn cục bổ sung cho nhau. Lực địa phương có hiệu quả khi ở gần biên của đối tượng trong khi lực toàn cục có hiệu quả khi ở xa biên của đối tượng.

Tham số độ lệch chuẩn  $\sigma$ , và tham số chuẩn hóa  $\psi$  đóng vai trò quan trọng trong mô hình. Tham số  $\sigma$  cho phép điều khiển khả năng mở rộng khu phức (region-scalability) từ một lân cận nhỏ cho đến toàn bộ miền ảnh. Tùy thuộc vào chất lượng ảnh và nội dung của ảnh mà ta chọn tham số  $\sigma$  cho phù hợp. Thực tế cho thấy với các ảnh nhiều nhiễu hoặc độ tương phản thấp thì tham số  $\sigma$  cần được chọn lớn. Điều này sẽ gây ảnh hưởng đến tốc độ chạy thuật toán. Cùng với đó, nếu chọn  $\sigma$  quá nhỏ sẽ làm ảnh hưởng đến kết quả dẫn đến kết quả không mong muốn.

### 3.4.2 Giải số

Tiếp theo chúng ta sẽ xây dựng các bước giải số cho mô hình. Giá trị  $c_1, c_2$  làm cực tiểu hàm năng lượng trên được xác định như sau:

$$c_1(\phi) = \frac{\int_{\Omega} u_0(x, y) H(\phi(x, y)) dx dy}{\int_{\Omega} H(\phi(x, y)) dx dy} \quad (3.48)$$

$$c_2(\phi) = \frac{\int_{\Omega} u_0(x, y) (1 - H(\phi(x, y))) dx dy}{\int_{\Omega} H(\phi(x, y)) (1 - H(\phi(x, y))) dx dy} \quad (3.49)$$

Cố định  $c_1, c_2$ . Thêm thành phần biến thiên  $\eta$  ta được  $\phi = \bar{\phi} + \epsilon\eta$ . Đạo hàm theo  $\phi$  và cho  $\epsilon$  tiến dần tới 0 ta có

$$\begin{aligned} \frac{\delta E^{GLIF}}{\delta \bar{\phi}} &= \lim_{\epsilon \rightarrow 0} \frac{d}{d\epsilon} \left( \frac{\alpha}{2} \int_{\Omega} |I(x) - m_1 H_{\epsilon}(\bar{\phi}) - m_2 (1 - H_{\epsilon}(\bar{\phi}))|^2 dx \right) \\ &\quad + (1 - \alpha) \int_{\Omega} |u_0 - c_1|^2 H(\bar{\phi}) + |u_0 - c_2|^2 (1 - H(\bar{\phi})) dx \\ &= \alpha \lim_{\epsilon \rightarrow 0} \frac{d}{d\epsilon} \left( - \int_{\Omega} -\delta_{\epsilon}(\bar{\phi}) (I(x) - m_1 H_{\epsilon}(\bar{\phi}) - m_2 (1 - H_{\epsilon}(\bar{\phi}))) (m_1 - m_2) \eta dx \right) \\ &\quad + (1 - \alpha) \delta_{\epsilon}(\bar{\phi}) \int_{\Omega} (I - c_1)^2 + (I_0 - c_2)^2 \eta dx \\ &= -\alpha \left( \int_{\Omega} -\delta_{\epsilon}(\bar{\phi}) (I(x) - m_1 H_{\epsilon}(\bar{\phi}) - m_2 (1 - H_{\epsilon}(\bar{\phi}))) (m_1 - m_2) \eta dx \right) \\ &\quad + (1 - \alpha) \delta_{\epsilon}(\bar{\phi}) \int_{\Omega} ((I - c_1)^2 + (I_0 - c_2)^2) \eta dx \end{aligned} \quad (3.50)$$

Theo đó ta có phương trình Euler-Lagarange:

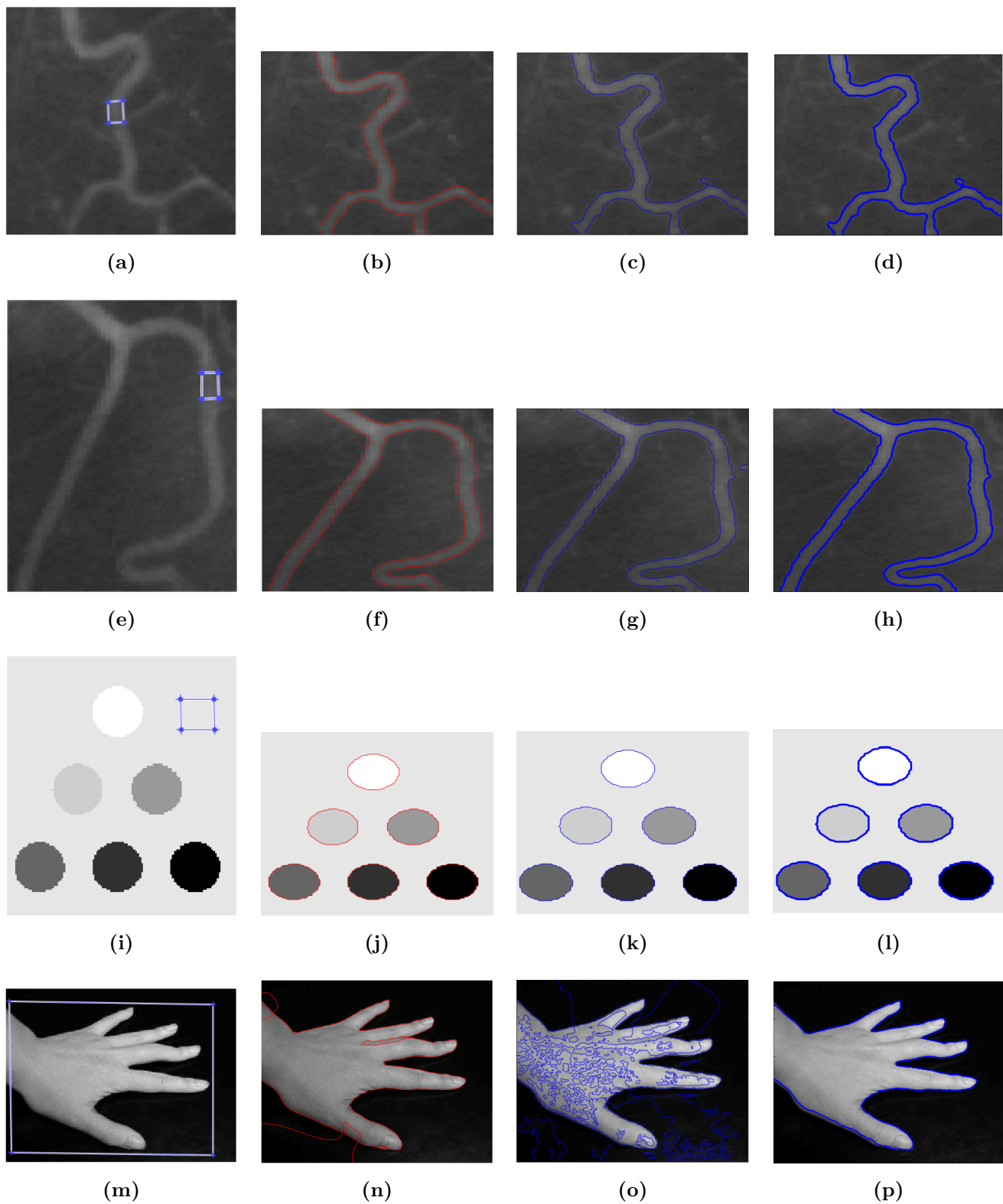
$$\delta_\epsilon(\phi)\{\alpha(I - I^{LFI})(m_1 - m_2) + (1 - \alpha)((I - c_1)^2 + (I_0 - c_2)^2)\} \quad (3.51)$$

Cụ thể các bước của thuật toán sẽ như sau:

- Bước 1: Khởi tạo đường cong  $\phi_0$  ban đầu
- Bước 2: Tính  $c_1, c_2$  theo công thức
- Bước 3: Tính giá trị  $\phi_{n+1}$  theo  $\phi_n$  theo công thức
- Bước 4: Chuẩn hóa hàm tập mức  $\phi$  sử dụng hàm nhân Gauss  $\phi = G_\psi \phi$ , trong đó  $\psi$  là độ lệch chuẩn
- Bước 5: Kiểm tra điều kiện dừng. Nếu thỏa mãn quay lại bước 2

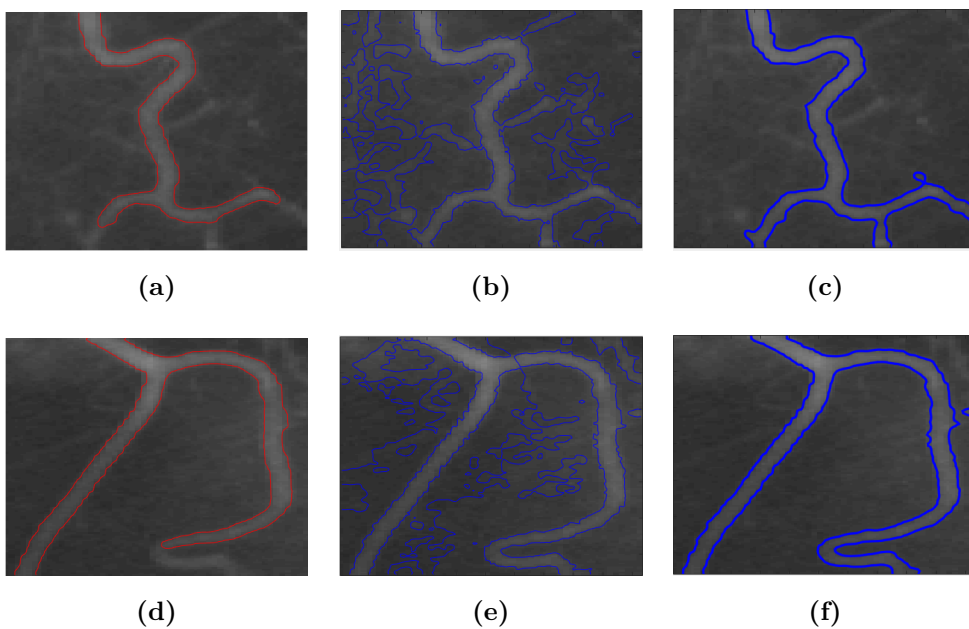
### 3.4.3 Kết quả

Kết quả hình (3.7) cho thấy mô hình GLIF xử lý tốt với cả ảnh đồng nhất cường độ và ảnh không đồng nhất cường độ. Đặc biệt kết quả thực nghiệm cho thấy mô hình GLIF hoạt động được trong trường hợp  $\Delta t$  lớn hơn so với các mô hình LBF và LIF trong khi vẫn đưa ra kết quả chấp nhận được. Điều này giúp tăng tốc độ chạy thuật toán. Hình (3.8) là kết quả minh chứng cho điều này:



**Hình 3.9:** Kết quả của mô hình GLIF so với các mô hình LBF và LIF. Ảnh a,e,i,m là đường cong khởi tạo ban đầu. Ảnh b,f,j,n là kết quả với mô hình LBF, ảnh c,g,k,o là kết quả với mô hình LIF. Ảnh d,h,l,p là kết quả với mô hình GLIF





**Hình 3.10:** Kết quả của mô hình GLIF so với các mô hình LBF và LIF với tham số  $\Delta t = 0.1$ . Ảnh a, d là kết quả với mô hình LBF, ảnh b, e là kết quả với mô hình LIF. Ảnh c, f là kết quả với mô hình GLIF

## Chương 4

# Áp dụng mô hình GLIF trong theo dõi đối tượng trong video

### 4.1 Theo dõi đối tượng trong video là gì? Tại sao phải theo dõi đối tượng trong video

Theo vết đối tượng là một vấn đề quan trọng trong lĩnh vực thị giác máy tính. Sự xuất hiện của các máy tính cấu hình mạnh và các camera chất lượng cao và sự cần thiết của việc phân tích video là điều kiện để thúc đẩy sự phát triển của các thuật toán theo vết. Có 3 bước chính trong phân tích video là: Phát hiện các đối tượng chuyển động trong video, theo vết các đối đó theo từng frame, phân tích vết để phát hiện hành động của đối tượng. Do vậy, theo vết đối tượng phù hợp với:

- Nhận dạng dựa trên chuyển động
- Giám sát tự động, hay là theo dõi một cảnh để phát hiện các hoạt động đáng ngờ hoặc các sự kiện bất thường.
- Lập chỉ mục video, tạo chú thích tự động và tìm iến video trong cơ sở dữ liệu

- Tương tác giữa con người và máy tính, hay là nhận diện cử chỉ, theo dõi mắt làm đầu vào cho máy tính.
- Theo dõi giao thông, thu thập thống kê dòng phương tiện

Ở dạng đơn giản nhất, theo vết có thể được định nghĩa là ước lượng chuyển động của đối tượng trong miền ảnh khi nó chuyển động xung quanh miền ảnh. Một cách khác, người theo dõi thực hiện gán nhãn các đối tượng được theo dõi trong các khung hình của video. Ngoài ra, tùy thuộc vào miền ảnh, việc theo dõi có thể cung cấp các đặc tính của đối tượng như: hình dạng, vị trí, hướng. Việc theo vết đối tượng có thể gặp khó khăn do một số ẩn đề như:

- Thiếu thông tin do các phép chiếu từ khung cảnh 3D ngoài đời thực vào ảnh 3D
- Ảnh nhiễu
- Những chuyển động phức tạp của đối tượng
- Thay đổi khung cảnh
- Yêu cầu xử lý thời gian thực

Điều này có thể đơn giản hơn nhờ việc đưa vào các ràng buộc về chuyển động cũng như là bề ngoài của đối tượng. Ví dụ, hầu hết các thuật toán theo vết đối tượng đều đưa vào ràng buộc tốc độ và gia tốc là hằng số.

Có nhiều phương pháp theo vết đối tượng được đề xuất. Các hướng này khác nhau ở cách tiếp cận để trả lời các câu hỏi: Đối tượng nào thích hợp để theo dõi? Các đặc tính nào của ảnh có thể sử dụng? Làm sao để mô hình hóa hình dạng, chuyển động của đối tượng trong video? Việc trả lời các câu hỏi này còn tùy thuộc vào hoàn cảnh, điều kiện

## 4.2 Biểu diễn đối tượng

Đối tượng có thể được định nghĩa là bất kỳ thứ gì mà người ta quan tâm và muốn phân tích nó. Ví dụ như các phương tiện ở trên đường, người đi bộ trên đường, hay các bong

bóng ở trong nước .... Các đối tượng có thể được đại diện bởi hình dạng, đối tượng.... Dưới đây là một số cách dùng để biểu diễn hình dạng đối tượng thường được sử dụng trong theo vết.

- **Điểm:** Một đối tượng được biểu diễn bởi một tập hợp các điểm. Cách này thường được dùng trong trường hợp đối tượng được theo vết có kích thước nhỏ
- **Các hình dạng cơ bản:** Đối tượng được đại diện bởi các tam giác, elips,.... Chuyển động của đối tượng theo cách tiếp cận này được mô hình hóa bằng các dịch chuyển, tuyến tính, hay phép chiếu. Mặc dù cách này phù hợp với việc theo dấu các đối tượng rigid tuy nhiên nó có thể dùng cho các đối tượng không cứng
- **Đường viền :** Đường viền đại diện cho biên của vật. Phần bên trong đường viền được gọi là silhouette của đối tượng. Cách biểu diễn sử dụng đường biên này phù hợp với đối tượng non-rigid
- **Khớp nối:** Một đối tượng sẽ được chia thành các phần khác nhau và được kết nối bởi các khớp nối. Ví dụ với cơ thể người có thể được chia ra làm các phần: thân, chân, tay, đầu được kết nối bởi các khớp nối. Quan hệ giữa các thành phần được chi phối bởi các mô hình động học.

#### **4.2.1 Lựa chọn đặc tính để thực hiện theo vết**

Thông thường, để đạt được kết quả như mong muốn, người ta thường chọn thuộc tính đảm bảo được tính duy nhất để đảm bảo đối tượng có thể được phân biệt. Các đặc tính này thường được chọn để phù hợp với cách biểu diễn đối tượng. Ví dụ màu thường được dùng làm đặc tính để tìm vết trong trường hợp đối tượng được biểu diễn bởi biểu đồ histogram. Còn cạnh đối tượng thường được dùng làm đặc tính trong trường hợp biểu diễn đối tượng bằng đường viền. Và thông thường chúng thường trong thực tế chúng thường được kết hợp với nhau để được kết quả cao hơn. Các đặc tính thường được dùng là:

- Màu sắc rõ ràng của một vật bị ảnh hưởng bởi các yếu tố phân bố điện quang của ánh sáng và tính chất phản xạ bề mặt của đối tượng. Không gian màu RGB thường được dùng nhiều trong xử lý ảnh, nó là không gian màu không đồng nhất cảm quan (non perceptually uniform), hay sự khác biệt về màu trong hệ RGB không tương ứng với sự khác biệt về màu đối với cảm nhận của con người. Ngược lại hệ màu  $L^*u^*v$ ,  $L^*a^*b$  lại là không gian màu đồng nhất cảm quan. Tuy vậy nó lại nhạy cảm với nhiễu. Do mỗi không gian màu sẽ có một ưu nhược điểm riêng và không thể nói được rằng không gian màu nào được dùng sẽ hiệu quả hơn nên trong thực tế người ta dùng rất nhiều loại không gian màu khác nhau để thực hiện tìm vết
- Cạnh : Biên của đối tượng thường là nơi có sự thay đổi lớn về cường độ ảnh. Thuộc tính quan trọng của đặc tính này là nó ít nhạy cảm hơn với sự thay đổi về độ sáng hơn là đặc tính màu.
- Texture: Texture giống như một độ đo sự biến đổi cường độ trên bề mặt thỏa mãn các thuộc tính. như độ trơn, độ đều. So với màu, texture cần thêm thao tác xử lý để có được đặc tính.

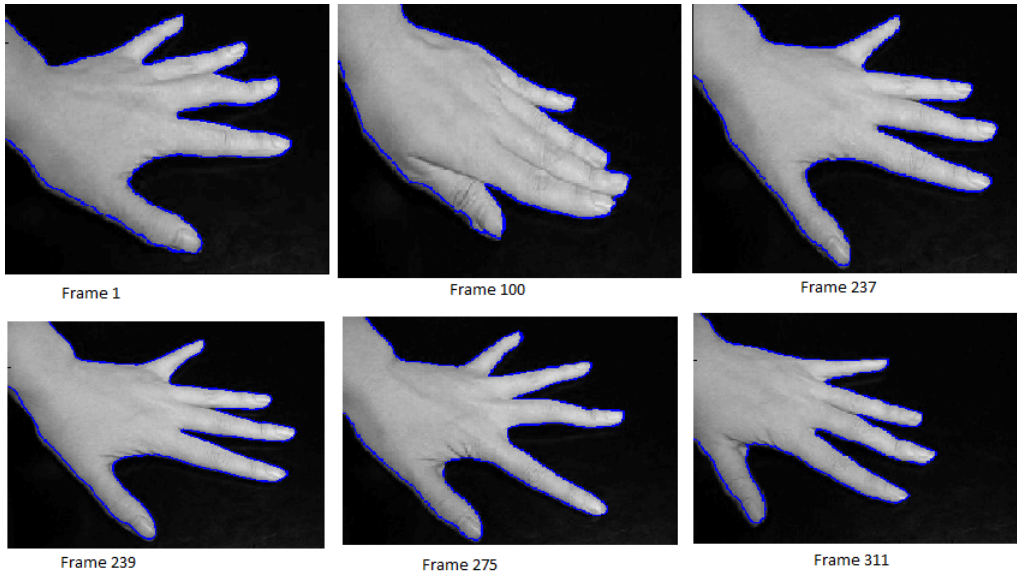
### 4.3 Theo vết đối tượng sử dụng mô hình GLIF

Thuật toán dừng lại cơ bản là áp dụng mô hình GLIF trong mỗi frame trong video. Tôi sử dụng kết quả cuối cùng của một khung hình làm đầu vào cho khung hình sau. Dưới đây là giả code cho mô hình này:

---

```
Co=initial Contour
for k=1 to n_frame
{
C_k=GLIF(C{k-1},I_{k});
draw contour on image
output frame
}
```

**Hình 4.1:** Kết quả trong các frame khác nhau



---

Thuật toán gặp khó khăn trong trường hợp đối tượng trong frame sau không giao với đường biên cuối cùng ở frame trước. Điều này có thể dẫn đến kết quả không mong muốn.

## 4.4 Kết quả

Hình 4.1 thể hiện kết quả của thuật toán trong các frame khác nhau của video. Kết quả cho thấy thuật toán hoạt động tốt ngay cả trong trường hợp có sự thay đổi lớn trong các frame. Tuy vậy thuật toán chỉ giải quyết được trường hợp frame sau có giao nhau với biên của đối tượng trong frame trước.

## Chương 5

# Áp dụng GLIF trong phân vùng ảnh 3 chiều

### 5.1 Ảnh 3 chiều trong thực tế

Ngày nay, với sự ra đời của các thiết bị y tế như máy chụp cắt lớp(CT), máy cộng hưởng từ(MRI) với khả năng cung cấp hình ảnh 3 chiều của các bộ phận trong cơ thể, các bác sĩ đã dễ dàng chuẩn đoán và thực hiện thao tác với các bộ phận trong thực tế hơn nhờ quan sát trên các hình ảnh 3 chiều. Các thiết bị hoạt động theo cơ chế dùng tia X quang chụp cắt lớp các bộ phận theo nhiều hướng khác nhau. Hình ảnh cắt lớp 2 chiều được chụp từ các thiết bị này sau đó sẽ được đưa vào máy tính xử lý để có thể tạo dựng lại hình ảnh 3 chiều Một ảnh 3 chiều được định nghĩa trên miền ảnh  $\Omega$  như sau:

$$\begin{aligned} f(x) &\in \mathbb{R} \\ x &\in \Omega \end{aligned} \tag{5.1}$$

Trong đó  $\Omega$  là không gian rời rạc 3 chiều. Một ảnh đã được phân vùng được định nghĩa trong cùng miền  $\Omega$  với ảnh  $f$  nhưng lấy giá trị trong không gian nhãn rời rạc.

$$g(x) \in \mathbb{N} \tag{5.2}$$

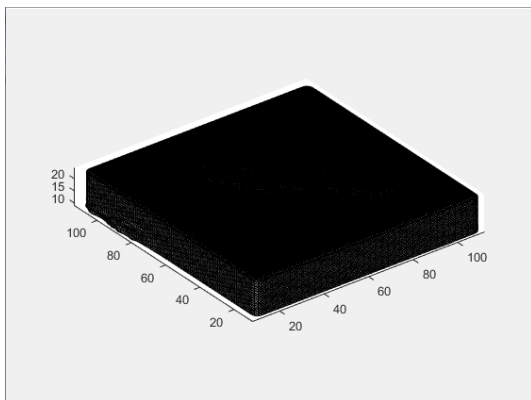
Ảnh nhị phân là một trường hợp đặc biệt của định nghĩa này, hàm  $g$  nhận giá trị 0 hoặc 1 tương ứng với nền và phần nội dung trong ảnh.

## 5.2 Kết quả mô hình GLIF cho ảnh ba chiều

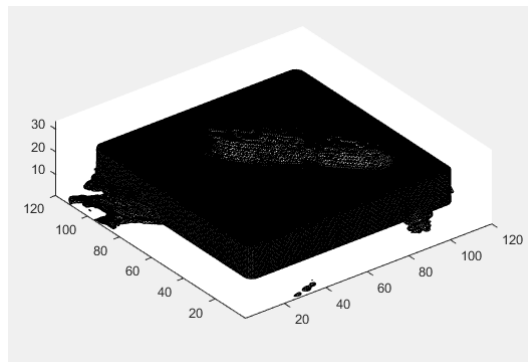
Có hai hướng tiếp cận chính trong phân vùng ảnh 3 chiều là thực hiện xử lý trên từng lát ảnh hoặc xử lý trực tiếp trên ba chiều. Trong phần này tôi thực hiện tiếp cận theo cách thứ 2 tức là thuật toán sẽ thực hiện xử lý trực tiếp trên ảnh 3 chiều thay vì xử lý trên từng lát ảnh. Để có thể áp dụng mô hình GLIF cho ảnh ba chiều thay vì xuất phát từ một đường cong khởi tạo ban đầu ta thực hiện xuất phát từ một mặt khởi tạo ban đầu. Các hàm năng lượng được viết lại trong ba chiều tương tự trong hai chiều. Dưới đây là kết quả phân vùng trong ảnh 3 chiều chạy chiều. Hình (5.1) là kết quả quá trình tiến hóa mặt với các tham số trong thuật toán là  $\Delta t = 4, \psi = 0.3, \mu = 0.1$ . Hình (5.2) là kết quả trên từng lát ảnh được trích ra từ kết quả đầu ra.

Hình (5.3) là kết quả quá trình tiến hóa mặt được thực hiện trên ảnh chụp cắt lớp não với các tham số trong thuật toán là  $\Delta t = 0.1, \psi = 0.3, \mu = 0.1$ . Hình (5.4) là kết quả trên từng lát ảnh ứng với kết quả đầu ra tương ứng với hình (5.3).

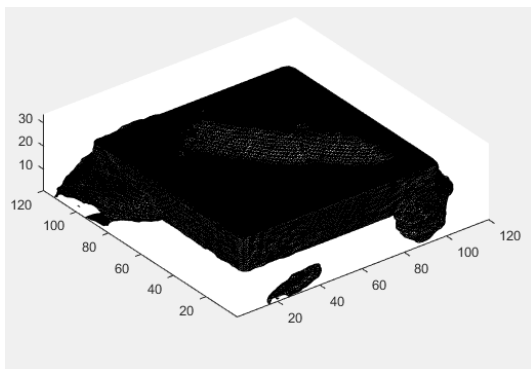




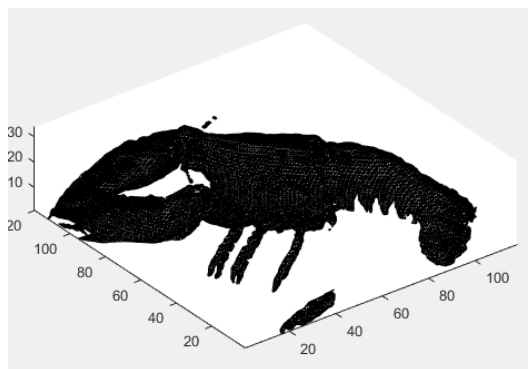
(a) Mặt khởi tạo ban đầu



(b) Bước lặp 11



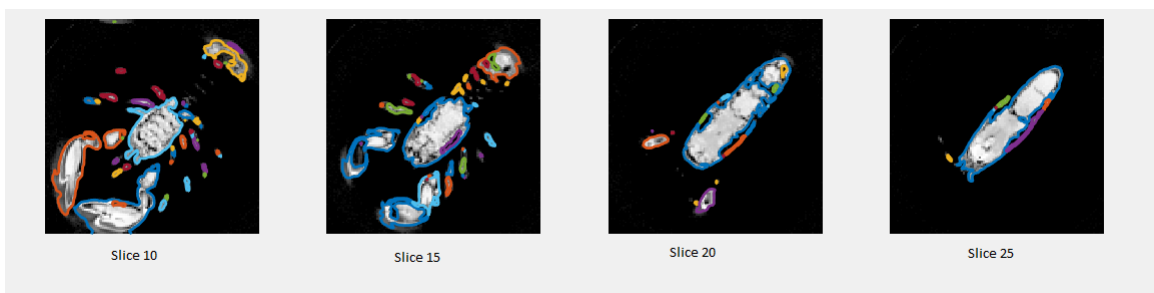
(c) Bước lặp 19

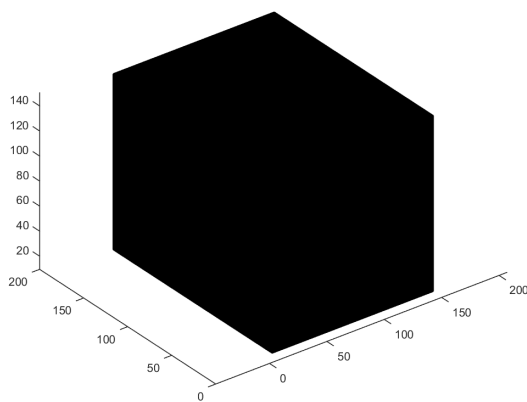


(d) Bước lặp 40

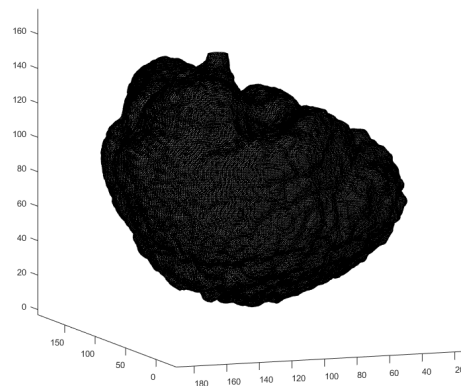
**Hình 5.1:** Kết quả của mô hình GLIF với ảnh chụp cắt lớp một con tôm

**Hình 5.2:** Kết quả phân vùng trong các lát ảnh khác nhau





(a) Mặt khởi tạo ban đầu



(b) Bước lặp 11

**Hình 5.3:** Kết quả của mô hình GLIF với ảnh chụp cắt lớp não người

**Hình 5.4:** Kết quả phân vùng hình chụp cắt lớp não trên các lớp ảnh khác nhau

