**2.5 Độ lệch chuẩn.**

**2.5.1.Khái niệm và vai trò**

Độ lệch chuẩn (standard deviation) là một đại lượng thống kê dùng để đo mức độ phân tán (biến động) của một tập hợp giá trị xung quanh giá trị trung bình.

Trong xử lý ảnh, độ lệch chuẩn cho biết mức độ biến đổi cường độ sáng trong một vùng ảnh.

Biểu thức toán học:

Trong đó:

* : giá trị cường độ tại pixel thứ i
* : giá trị trung bình của các pixel trong toàn bộ vùng.
* :sigmaσ: độ lệch chuẩn
* N:số pixel trong toàn bộ vùng

Trong matlab độ lệch chuẩn đã được lập trình sẵn bằng lệnh *std(ảnh cần đo);*

**2.5.2 Cách hoạt động trong hệ thống.**

Sau giai đoạn tiền xử lí,xác định khoanh vùng nghi ngờ là có chuyển động ta cần loại bỏ thêm những yếu tố không chắc chắn như đổ bóng nhẹ do tần số quét của camera khi thu sáng,ánh sáng lập lòe.Những yếu tố đó sẽ có cường độ mức xám xấp xỉ nhau. sẽ vô cùng thấp.  
Còn nếu là vùng thực sự có chuyển động các mức xám sẽ tạo ra độ chênh lệch lớn bởi sự đổ bóng,ánh sáng dìa vật thể thay đổi mạnh so với phần phía trong của vật nên của vùng này sẽ trả về kết quả lớn(hơn mức qui định của chúng ta khi lập trình).

Ví dụ:

Chương trình đã xác định được 2 vùng là có chuyển động.

**A**: Vùng có chuyển động giả do nhiễu.

**B**:Vùng thực sự có chuyển động.

|  |  |  |
| --- | --- | --- |
| 0.2 | 0.5 | 0.4 |
| 0.25 | 0.4 | 0.5 |
| 0.1 | 0.2 | 0.7 |

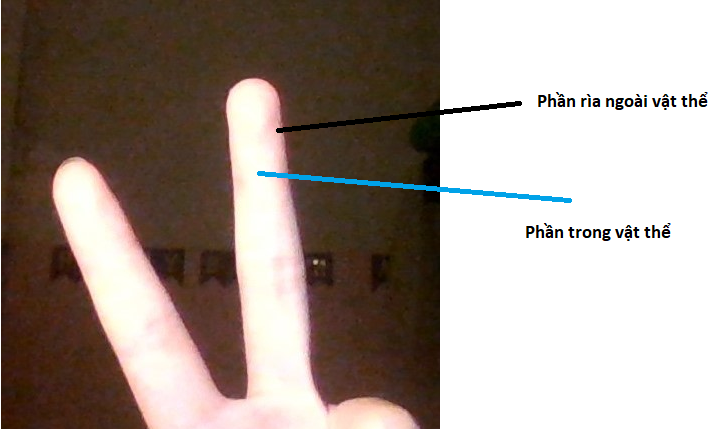
Vùng B

|  |  |  |
| --- | --- | --- |
| 0.4 | 0.5 | 0.4 |
| 0.5 | 0.4 | 0.5 |
| 0.4 | 0.4 | 0.4 |

Vùng A

Có thể thấy các pixel phía ngoài trong vùng B có cường độ sáng chênh lệch khá lớn-hoặc rất cao so với phần trung tâm.nguyên nhân chính như sau:

Phần này nằm mép ngoài có diện tích khá nhỏ nên phản xạ ánh sáng kém và trong nền tối sẽ xu hướng có màu tối tức mức xám thấp,ngược lại trong tường hợp gần nguồn sáng và chất liệu có độ phản xạ cao thì nó lại dẫn tới tình trạng lóa viền tức mức xám cao.Còn phần giữa có diện tích lớn hơn có khả năng hấp thụ và phân tán ánh sáng tốt nên mức xám của các pixels thường có giá trị xấp xỉ nhau.  
Ví dụ bằng ảnh thực tế:



Hình hai ngón tay có mức xám(độ sáng khá đều nhau ở phía bên trong) nhưng phần viền của ngón tay lại khá mờ và tối do nền xung quanh thiếu sáng.

Đến đây lại có 1 câu hỏi :

Độ lệch chuẩn dựa vào sự chêch lệch mức xám trong 1 vùng nghi ngờ để kết luận.

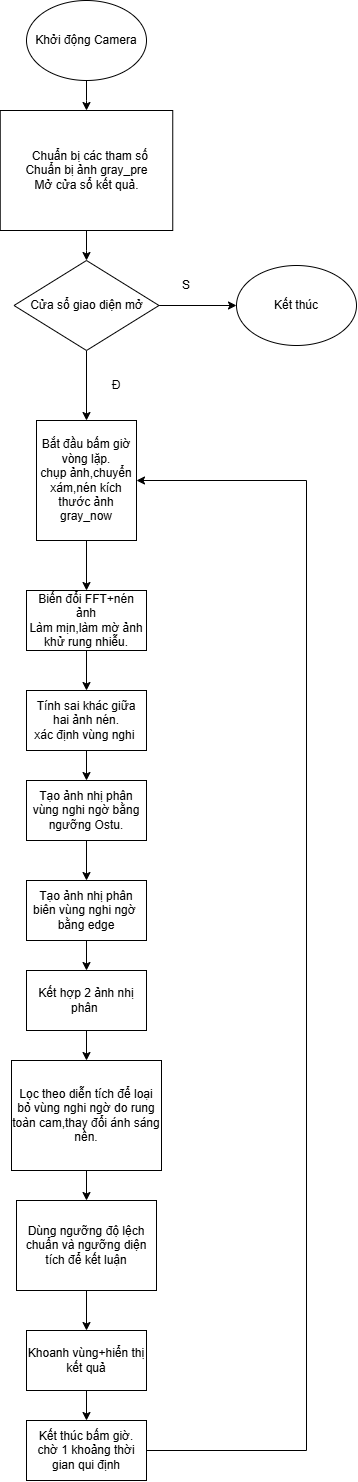
“Nếu vậy tôi mặc quần đen áo trắng,hoặc 1 vật thể chứa nhiều họa tiết có độ tương phản rõ ràng dẫn tới sự chêch lệch mức xám khác nhau trong cùng 1 vùng ,liệu thuật toán có hiểu nhầm là đang có chuyển động không?Mặc dù tôi/vật thể đứng yên.”

Đây là điểm yếu khi dùng độ lệch chuẩn.Tuy nhiên trong trường hợp đã qua nhiều giai đoạn tiền xử lí.Khung hình phải có sự thay đổi và được xác định/nghi ngờ thì chúng ta mới bắt đầu tính độ lệch chuẩn.

Còn nếu người/vật thể đứng yên thì vùng chứa bạn trong hình 1 và hình 2 không thay đổi sẽ bị thuật toán phía trên bỏ qua.

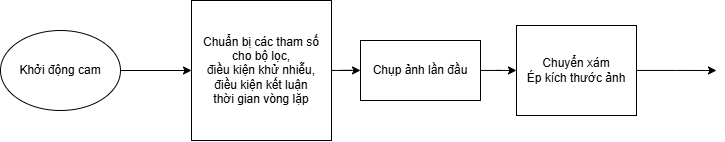
**3.Lưu đồ thuật toán.**

**3.1 Lưu đồ tổng quát**

****

**3.2 Lưu đồ từng giai đoạn.**

**3.2.1 Chuẩn bị**

****

**Chuẩn bị các tham số:**

* Tỷ lệ bán kính vùng tần số được giữ lại so với chiều dài của ảnh:Quyết định vùng tần số thấp được giữ lại khi áp dụng bộ lọc Gaussian.

Tên biến trong code: **keep\_ratio** ;

* Diện tích vùng tối thiểu được coi là chuyển động hợp lệ:Nhỏ hớn vùng này được coi là nhiễu.

Tên biến trong code: **area\_box\_thresh ;**

* Tỷ lệ Diện tích vùng tối đa được coi là chuyển động hợp lệ so với ảnh: Lớn hơn vùng này,chẳng hạn như rung camera,tắt điện phòng,...đều được coi là nhiễu nền dẫn tơi sự thay đổi toàn bộ khung hình và được coi là nhiễu.Ví dụ >40% kích thước ảnh .

Tên biến trong code: **max\_area\_ratio ;**

* Cửa số bộ lọc trung bình:Làm mượt hoặc xóa hoàn toàn những chi tiết nhỏ như ánh sáng lập lòe,người hơi rung nhẹ.Thậm chí cả những chuyển động nhỏ như chớp mắt,người khẽ nhúc nhích,ánh sáng lập lòe.

Kí hiệu trong code: **avg\_kernel;**

* Thời gian thực hiện mỗi vòng lặp:Qui định thời gian chờ chương trình chạy camera chụp ảnh mới,ảnh hưởng bởi tốc độ xử lí của CPU và độ mượt khi so sánh hai khung hình trước đó và hiện tại thường được gọi :

(Khung hình/s)

Ví dụ ta đặt thời gian thực hiện mỗi vùng lặp là: 0.05s thì kết quả sẽ trả về 20(khung hình/s) đảm bảo không quá tải cho CPU mà vẫn xử lí mượt mà.

Kí hiệu trong code: **update\_delay;**

Xử lí ảnh cho khung hình đầu tiên:vì bản chất của thuật toán là so sánh sự khác biệt giữa hai khung hình nên trước khi vào vòng lặp ta cần tạo giá trị khởi đầu bên ngoài.

Giá trị này cũng phải được xử lí y hệt như trong vòng lặp để đảm bảo không gặp lỗi khi so sánh:đều là ảnh xám(1 kênh sáng),có cùng kích thước.

Ki hiệu trong code: **gray\_prev;**

Với chương trình hiện tại ta nén ảnh về kích thước **256x256**

Có 3 nguyên chính để ta chọn kích thước này:

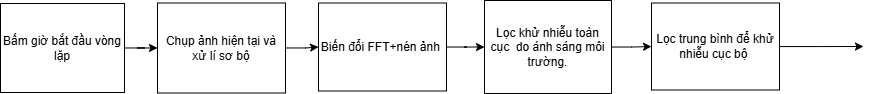
1. Giá trị của nó thuộc dạng lũy thừa 2 thuận tiện cho thuật toán FFT.
2. Kích thước trung bình giúp CPU xử lí mượt mà.
3. Không quá nhỏ để bị mất nhiều thông tin.

**3.2.2 Vòng lặp chính.**

Vì nội dung của FFT và nén ảnh đã đề cập chi tiết trong phần ứng dụng nén ản nên chúng ta sẽ bỏ qua khi giải thích lưu đồ.

**3.2.2.1.Tiền so sánh**

Ở giai đoạn,nhiệm vụ chính là dùng các bộ lọc trong miền tần số/không gian để khử những loại nhiễu ảnh hưởng đến quá trình xử lí ,phát hiện.



* **Bấm giờ:**Matlab đã lập trình và khởi tạo hàm đo thời gian được lưu bằng lệnh **tic;** tuy nhiên để thuận tiện khi bấm giờ kết thúc ta sẽ gán lệnh này cho 1 biến cụ thể.

Ký hiệu trong code: **loop\_start;**

* **Ảnh hiện tại:**Xử lí tương tự ảnh đầu tiên(**gray\_pre)** bên ngoài vòng lặp.

Ký hiệu trong code: **gray\_now;**

* **Biến đổi FFT và nén ảnh** đối với cả ảnh hiện tại và trước đó:thu được 2 ảnh nén

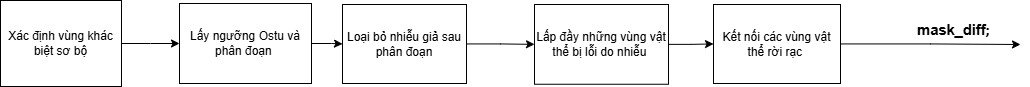
Ký hiệu trong code: **I\_prev\_nen; I\_prev\_nen**

* **Lọc khử nhiễu toàn cục**:Nhiễu ở đây đến từ ánh sáng môi trường,các vật thể/người trong khung hình sẽ phản xạ lại mạnh/yếu tùy theo cường độ sáng đó:ví dụ môi trường quá sáng,vật thể thường có xu hướng lóa hoặc nhòe viền,giai đoạn này sẽ giúp loại bỏ ánh sáng toàn cục đó bằng cách trừ đi độ sáng trung bình của toàn ảnh được matlab tính sẵn dưới lệnh **mean(tên\_ảnh);**

Ký hiệu của tên biến đầu ra vẫn không đổi so với bước trước.

* **Lọc nhiễu cục bộ:**Sau hai quá trình lọc sơ bộ,vẫn còn những nhiễu nhỏ đến từ bản thân người/vật thể ví vụ:vật có khả năng phản xạ tốt nhưng không đều trên toàn bộ bề mặt dẫn tới những điểm lóa nhẹ ví dụ: mặt nước.Hoặc con người dao động nhẹ rung người,chớp mắt,...  
  Kí hiệu trong code: **I1\_avg; I2\_avg;**

**3.2.2.2.So sánh và lấy phân đoạn vùng.**

****

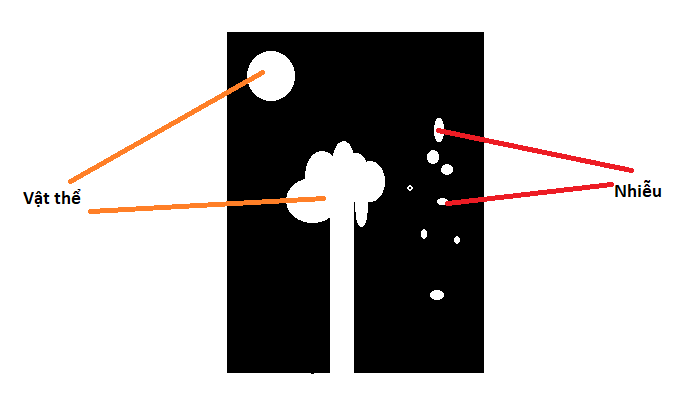
* **Vùng khác biệt**:Được xác định bằng hiệu không gian của hai khung hình.Để đảm bảo quá trình phân đoạn ảnh phía sau không bị lỗi kết quả của giá trị này phải là số thực và được chuẩn hóa nằm trong khoảng từ **0-1**

Kí hiệu trong code: **diff\_norm;**

* **Ngưỡng Ostu:** đã được matlab hỗ trợ bằng hàm **graythresh(tên\_ảnh);**

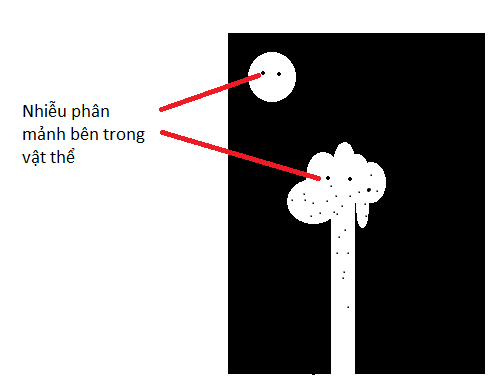
Kí hiệu trong code:**level;**

* **Loại bỏ nhiễu giả:** qua giai đoạn lọc nhiễu hầu hết các nhiễu đã được loại bỏ .Tuy nhiên vì môi trường vật thể khác nhau theo từng thời điểm nên những thông số bộ lọc không thể đảm bảo đúng cho mọi trường hợp hoặc đến từ bản thân quá trình phân đoạn do ngưỡng Ostu không hoàn toàn tương thích chính vì vậy vẫn còn những vùng nhiễu nhẹ được nghi ngờ là chuyển động,tuy nhiên những vùng này có kích thước khác nhỏ.Ta hoàn toàn có thể loại bỏ chúng bằng lọc ngưỡng sau phân đoạn:



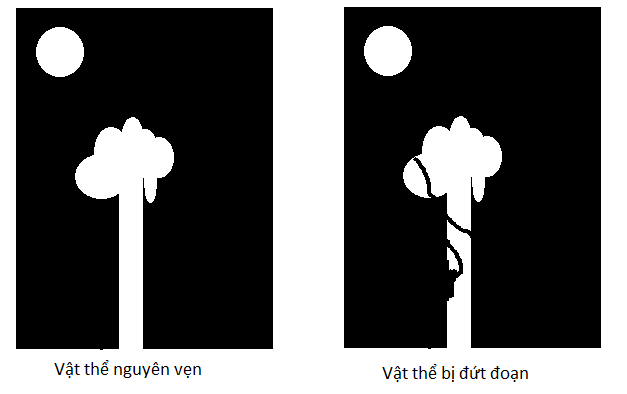
Hàm trong matlab: **bwareaopen(tên ảnh, 100);** Tức những vùng trắng có diện tích <100 pixel thì toàn bộ pixel trong đó đều được đưa về 0.

* **Lấp đầy vật thể**:Bản thân vật thể mang nhiều hình thái và chi tiết khác nhau hoặc đến từ việc bỏ sót nhiễu ánh sáng trong quá trình lọc sẽ dẫn đến việc có nhiều mảnh vỡ nhỏ trên thân vật thể sau phân đoạn.Bước này phải thực hiện sau khi đã loại bỏ những vùng chuyển động giả do nhiễu ở phía trên để trách khuếch đại những vùng chuyển động giả đó.



Câu lệnh của matlab: **mask\_diff = imclose(tên ảnh, strel('disk', 3));** tức là tại vị trí của mỗi pixel có giá trị 1 nó sẽ gán cho các pixel phạm vi xung quanh nó 3 ô(pixel) đều bằng 1 giống nó.

* **Kết nối các vùng vật thể rời rạc:**Tuy là rời rạc nhưng khoảng cách qui định ở đây phải có con số tối đa.  
  Tránh trường hợp nối hai vật thể khác nhau vào làm 1 .



Tên biến kí hiệu trong code của 3 quá trình: **mask\_diff;**

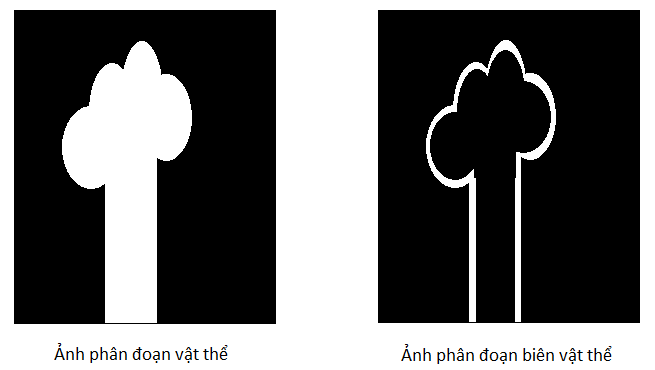
**3.2.2.2.So sánh và lấy phân đoạn biên.**

****

* **Phát hiện biên:** tập trung vào sự thay đổi đường nét và hình dạng của vật thể, thay vì bị ảnh hưởng bởi sự phân bố mức xám trên toàn bộ vật thể. Điều này giúp hệ thống phát hiện được chuyển động thật sự dựa trên hình học, ngay cả khi mức xám nội tại của vật thể thay đổi không đáng kể hoặc bị ảnh hưởng bởi ánh sáng môi trường.

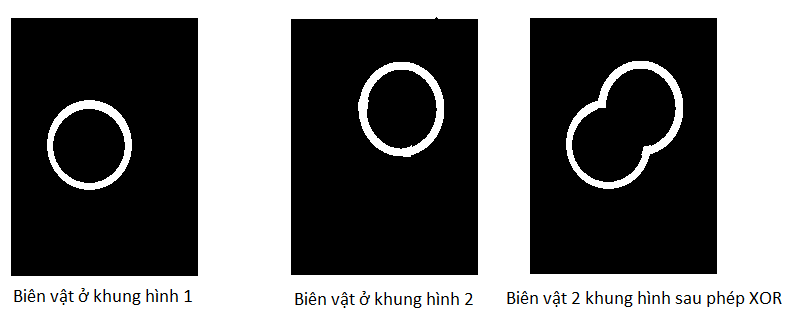
Trong giai đoạn tiền so sánh (3.2.2.1. ) ta đã có bước khử nhiễu toàn cục nhằm loại bỏ sự ảnh hưởng của ánh sáng môi trường nên vật thể .Tuy nhiên bước đó mang tính toàn cục,tức trên toàn ảnh còn nếu như những thay đổi vùng cục bộ thì nó không thế xử lí toàn bộ.

Dễ hình nhất đó chính là 1 chiếc đèn chiếu trực diện về phía nhấp nháy nhẹ hoặc sáng dần tối dần.Nếu chỉ phân đoạn vùng vật thể bằng ngưỡng Ostu thuật toán sẽ hiểu chiếc đèn biến mất và xuất hiện liên tục.  
Nhưng phát hiện biên thì sẽ chỉ nhìn thấy hình dạng đường nét của chiến đèn.Nếu nó vẫn giữ nguyên ở hướng đó thì thuật toán sẽ vẫn nhìn nó dưới dạng hình tròn.Khi thay đổi góc độ quay sang trái/phải có hình elip,hoặc hình dạng của thân đèn.



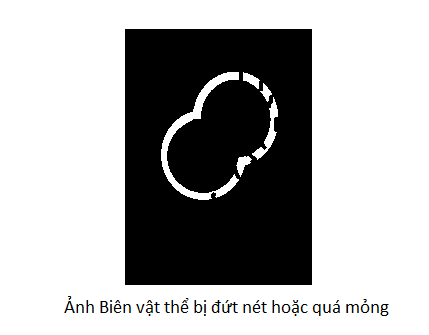
Ký hiệu trong code: **edge\_prev; edge\_now;**

* **Phát hiện điểm khác biệt giữa hai ảnh biên:**Thuật toán dùng phép XOR để giữ lại những phần khác nhau giữa 2 ảnh phân đoạn vùng vật thể.



* **Làm đậm đường biên vật thể:**Sau khi lấy những phần khác biệt,có thể bị đứt nét(nguyên nhân không phải đến từ bước xor mà từ quá trình phân đoạn biên trước đó).Nên để tránh trường hợp bỏ sót chuyển động ta cần mở rộng và nối liền những đường biên(nếu bị đứt nét).

Ký hiệu trong code: **edge\_diff**

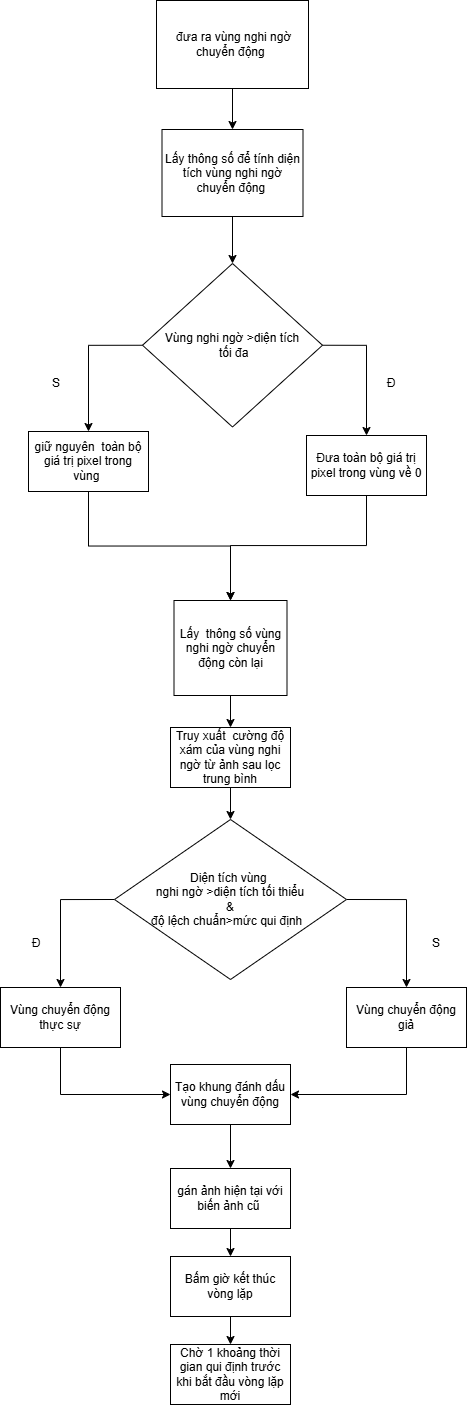


Lệnh **imdilate** vẫn được sử dụng và hoạt động tương tự như khi kết nối các vùng vật thể bị dứt đoạn như ở giai đoạn trước.

* **Loại bỏ biên giả:**Tương tự như phân đoạn vùng vật thể,sẽ còn những chấm nhiễu nhỏ sau khi phân đoạn.Lệnh **bwareaopen** vẫn là công cụ thực hiện nhiệm vụ này giúp chúng ta.Tuy nhiên ngưỡng diện tích vùng được coi là nhiễu sẽ nhỏ hơn so với khi dùng với vùng vật thể.Bởi bản thân vốn đã rất mỏng dù đã được mở rộng trước đó**.**Cũng vì lí do đó mà khác với ảnh phân đoạn vùng vật thể,đối với biên vật thể bước này phải được thực hiện sau khi mở rộng nối liền biên.

Ký hiệu trong code cả 2 bước này : **mask\_edge;**

**3.2.2.3. Kết hợp và đưa ra kết quả.**



* **Vùng nghi ngờ chuyển động :**Kết hợp từ kết quả của hai giai đoạn trước bằng phép AND.

Từ việc phân tích đánh giá những đặc tính của kết quả phân đoạn **biên** và **vùng** vật thể có hai nguyên chính để bắt buộc phải kết hợp:

1. **Vùng vật thể :**Đánh dấu vùng có sự khác biệt mức xám giữa hai thời điểm (giống kiểu "ảnh trừ"). Dễ bị nhiễu bởi ánh sáng, phản xạ, rung nhẹ.
2. **Biên Vật thể:** Đánh dấu vùng có thay đổi biên cạnh, tức là hình dạng vật thể bị thay đổi.Ổn định hơn nhưng có thể bỏ sót nếu vật thể thay đổi chậm.

Hai quá trình này bù trừ ưu nhược điểm cho nhau để cho ra kết quả tốt nhất

Ký hiệu trong code: **final\_mask;**

* **Lấy thông số các vùng nghi ngờ và tính diện tích:**Lệnh **regionprops** sẽ được cùng thuộc tính **BoundingBox** nhằm xác định hình chữ nhật có kích thước gần nhất bao quanh vùng nghi ngờ.Kết quả trả về gồm 4 thông số :

1. **X:**Toạn độ góc trên bên trái hình chữ nhật;
2. **Y:**Tọa độ góc trên bên phải hình chữ nhật;
3. **Width:**Chiều rộng hình chữ nhật
4. **Height:**Chiều cao/dài hình chữ nhật

4 thông số này được lưu vào 1 mảng cấu trúc: **stats\_all;**

Thực tế ở bước này chỉ có hai thông số và **height** và **width** được sử dụng để sử để tính diện tích và kết quả được lưu vào mảng có ký hiệu trong code: **total\_area;**

* **Loại bỏ vùng chuyển động giả(sơ bộ):**Những vùng nghi ngờ do thay đổi toàn cục dẫn tới nhòe hình,thay đổi toàn bộ Biên và mức xám sẽ có diện tích khá lớn thường từ 1 nửa đến toàn bộ khung hình.Ta sẽ dùng tham số **max\_area\_ratio** đã được thiết lập trong giai đoạn chuẩn bị để làm cơ sở loại bỏ.
* **Lấy thông tin vùng nghi ngờ(sau loại bỏ sơ bộ):**tiếp tục sử dụng lệnh **regionprops** và thuộc tính **BoundingBox** tuy nhiên ta cần thêm thuộc tính **PixelIdxList :** trả về danh sách tọa độ Pixel dưới dạng vector tuyến tính để thuận tiện cho bước truy xuất sau.

Tọa độ tuyến tính của các pixel trong ma trận ảnh.

Câu hỏi đặt ra:”Tại sao không lấy tọa độ trong gian (i;j) của ảnh mà phải gọi tọa độ dưới dạng vector?”

Điều này xuất phát từ việc Matlab thực hiện chương trình:đối với tọa độ (i;j) với mỗi tọa độ lấy được ta cần phải trải qua hai vòng lặp **for i....for j**.So với vector thì matlab thực hiện vòng lặp mất rất nhiều thời gian điều này sẽ làm chậm thuật toán.

Ký hiệu kết quả bước này trong code: **stats;**

* **Truy xuất giá trị của các pixel từ tọa độ tuyến tính:** Vùng nghi ngờ là sự kết hợp của hai kết quả phân đoạn vùng và phân đoạn biên.Giá trị của vùng này chỉ là 1,vô giá trị khi tính độ lệch chuẩn ở bước kế tiếp.Nên giá trị để tính độ lệch chuẩn cần phải có giá trị mức xám cụ thể.Vì kết quả của bước này được sử dụng trong việc kết luận nên giá trị mức xám phải được lấy từ ảnh đã qua các giai đoạn lọc nhiễu tức kết quả của bước **Lọc nhiễu cục bộ :I2\_avg;**

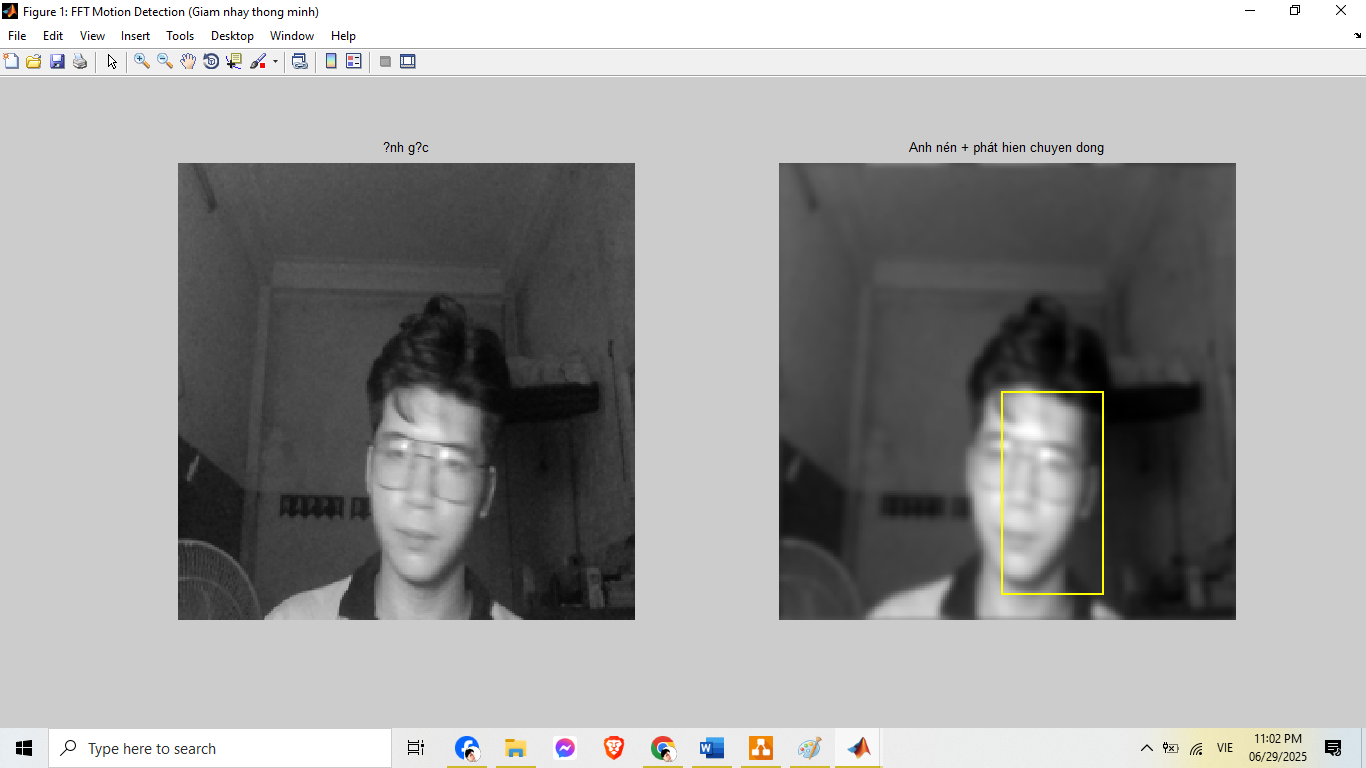
Ký hiệu trong code: **pixels;**

* **Loại bỏ vùng chuyển động giả (lần cuối):**Dựa vào hai thông số **area\_box\_thresh** và **ngưỡng độ lệch chuẩn.**
* **Vùng chuyển động đã được kết luận :**Được đóng khung viền bằng lệnh **rectangle.**
* **Khởi tạo giá trị khung hình cũ cho vòng lặp tiếp theo:**Gán ảnh hiện tại (**gray\_now)** cho biến **gray\_pre** để đảm bảo trong mỗi vòng lặp khung hình mới chụp được so sánh với khung hình trước đó**.**
* **Kết thúc bấm giờ:**Hàm **toc** trong matlab sẽ khóa biến **loop\_start** để kết thúc bộ đếm giờ đã khởi tạo ở đầu vòng lặp.

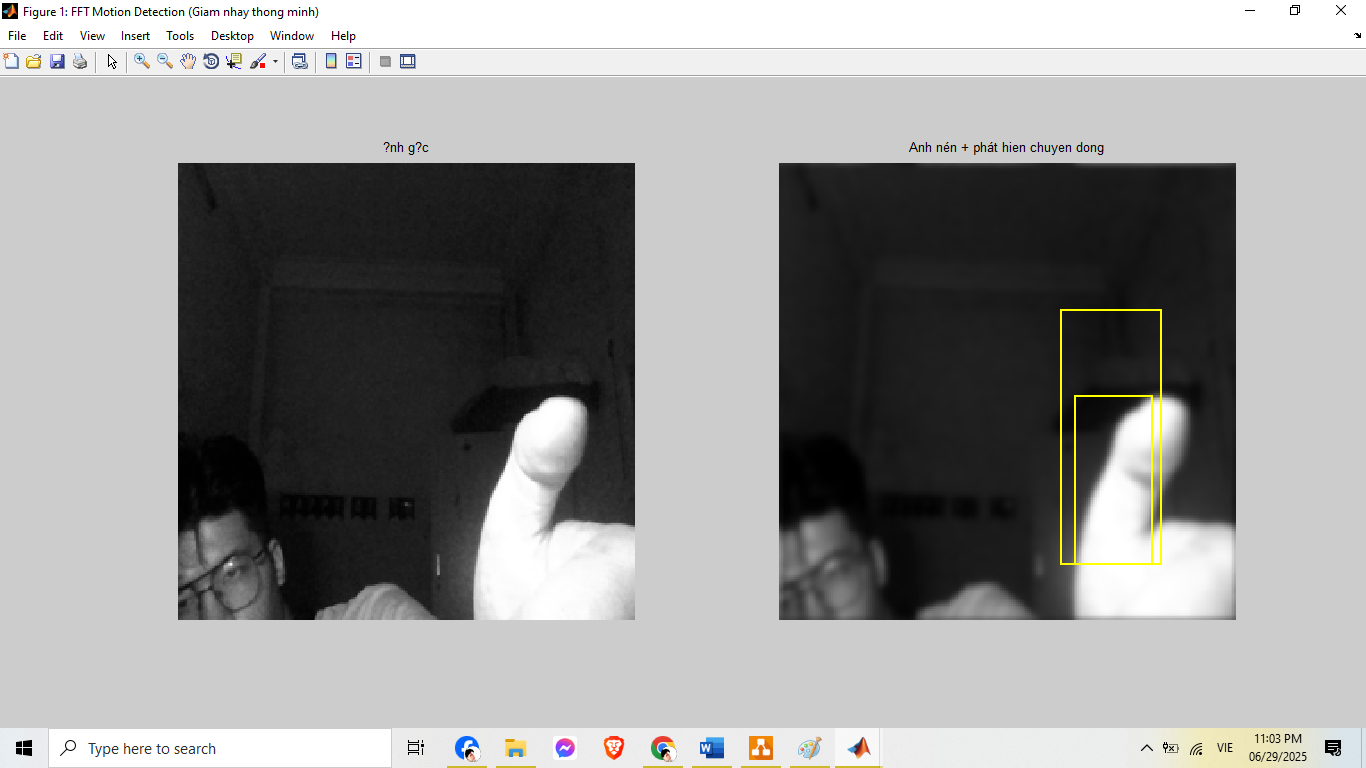
Ký hiệu trong code: **t\_elapsed** ;

* **Chờ đến vòng lặp kế tiếp:**Thời gian chờ đã được thiết lập trong quá trình chuẩn bị **update\_delay.**Thựchiện quá trình chờ bằng lệnh **pause(thời gian qui định-thời gian chương trình chạy)** tức là dù chương trình kết thúc sớm hơn thời gian qui định thì nó vẫn phải đợi trước khi vào vòng lặp tiếp theo.

**4.Mô phỏng**



Chuyển động của đầu.



Chuyển động của ngón tay.

**1.2 Mô tả hệ thống và nguyên lí hoạt động.**

****

Sơ đồ hệ thống phát hiện chuyển động

Nguyên lí hoạt động:Camera sẽ liên tục chụp ảnh của 2 khung hình liên tiếp ,trải qua những giai đoạn nén ,lọc nhiễu và so sánh sự khác biệt giữa 2 khung hình.Sau cùng là đối chiếu kết quả so sánh với những tham số điều kiện/ngưỡng để đưa ra kết luận và hiển thị.