

## MỞ ĐẦU

Trong thời đại ngày nay công nghệ thông tin hầu như đã thâm nhập vào toàn bộ các lĩnh vực đời sống xã hội. Xã hội càng phát triển thì nhu cầu về công nghệ thông tin ngày càng cao, do vậy dữ liệu số hầu như không còn xa lạ đối với mỗi người chúng ta. Trong mọi lĩnh vực các ứng dụng công nghệ thông tin đã trợ giúp con người rất nhiều.

Hiện nay, thông tin hình ảnh đóng vai trò rất quan trọng trong trao đổi thông tin, bởi phần lớn các thông tin mà con người thu nhận được đều thông qua thị giác. Trong các lĩnh vực công nghệ thông tin thì lĩnh vực giám sát tự động đã và đang thu hút được nhiều sự quan tâm của các nhóm nghiên cứu trong và ngoài nước. Cùng với sự phát triển của sức mạnh máy tính, các hệ thống giám sát tự động ngày càng tinh vi và hiện đại đã trợ giúp con người rất nhiều trong lĩnh vực an ninh, giám sát giao thông v.v.

Thời gian qua, hệ thống giám sát bằng camera đã trở thành lĩnh vực nghiên cứu rất phát triển. Trước các nhu cầu giám sát, đảm bảo an ninh mục tiêu, trụ sở, nhà riêng... của các tổ chức, cá nhân ngày càng cao đã thúc đẩy việc nghiên cứu, ứng dụng các sản phẩm giám sát thông qua camera theo dõi. Bài toán đặt ra là xây dựng và đưa vào sử dụng các hệ thống giám sát đủ mạnh, có độ chính xác cao, phù hợp với từng mục đích cũng như đối tượng sử dụng. Một số ứng dụng quan trọng của hệ thống giám sát camera trong các lĩnh vực như: Các thiết bị giám sát an ninh chuyên dụng sử dụng trong lực lượng vũ trang; các thiết bị giám sát an ninh dân dụng; các thiết bị giám sát, phân luồng giao thông; công nghệ nghiên cứu, chế tạo Robot thông qua việc phân tích các cử động của con người...

Ở nước ta hiện nay, lĩnh vực giám sát Video cũng có những bước phát triển đáng kể. Tuy nhiên nó chỉ mới dựa trên nền tảng phần cứng và cũng chưa áp dụng nhiều trong thực tế. Việc giải quyết bài toán này theo hướng tiếp cận sử dụng phần mềm chưa được quan tâm phát triển.

Một hệ thống giám sát camera bao gồm nhiều chức năng: Phát hiện chuyển động, phân loại, theo dấu, cảnh báo, phân tích hành động. Phát hiện chuyển động được xác định là một trong những chức năng đầu tiên và quan trọng nhất của một hệ thống giám sát camera. Nó không chỉ có chức năng trích xuất đối tượng chuyển động mà còn có ý

nghĩa quan trọng đối với ứng dụng thị giác máy tính như mã hóa video dựa trên đối tượng, phân tích chuyển động của con người và tương tác người – máy.

Xuất phát từ các nhu cầu thực tế, học viên đã báo cáo xin ý kiến thầy giáo hướng dẫn nghiên cứu luận văn của mình là: ***“Phát hiện chuyển động bằng phương pháp trừ nền”***. Kết quả của nghiên cứu là xây dựng giải thuật phát hiện chuyển động và phần mềm phát hiện và cảnh báo khi có đối tượng đột nhập vào khu vực giám sát một cách nhanh chóng và hiệu quả. Trên cơ sở đó nhằm xây dựng hệ thống giám sát bằng camera trụ sở của đơn vị sử dụng các nghiên cứu về thuật toán phát hiện chuyển động.

Nội dung chính của luận văn gồm 3 chương:

**Chương 1:** Khái quát về Video và phát hiện chuyển động

**Chương 2:** Nội dung giải thuật phát hiện chuyển động

**Chương 3:** Chương trình thử nghiệm phát hiện chuyển động

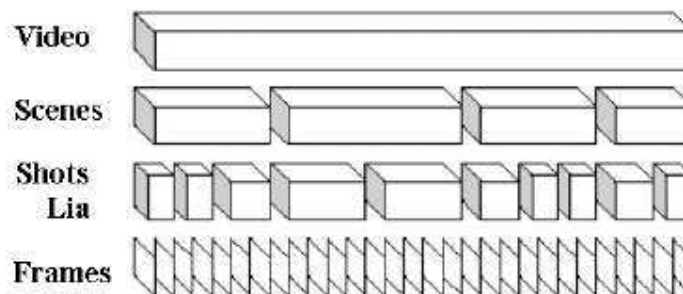
## Chương 1

### KHÁI QUÁT VỀ VIDEO VÀ PHÁT HIỆN CHUYỂN ĐỘNG

#### 1.1. Các khái niệm cơ bản về video [3]

Video là tập hợp các khung hình, mỗi khung hình là một ảnh. Shot (lia) là đơn vị cơ sở của video. Một lia là một đơn vị vật lý của dòng video, gồm các chuỗi các khung hình liên tiếp, không thể chia nhỏ hơn, ứng với một thao tác camera đơn.

Scene (cảnh) là các đơn vị logic của dòng video, một cảnh gồm các lia liên quan về không gian và liên kết về thời gian, cùng mô tả một nội dung ngữ nghĩa hoặc một tình tiết.



Hình 1.1: Cấu trúc phân đoạn video

Khi phim được chiếu, các khung hình lần lượt được hiển thị ở tốc độ nhất định. Tốc độ thường thấy ở các định dạng video khác nhau là 30 và 25 hình/s. Như vậy một giờ video sẽ có số khung hình tương ứng là 108000 hoặc là 90000. Dù là video ở định dạng nào thì nó cũng có dung lượng rất lớn và nếu xử lý với tất cả các khung hình thì thật không hiệu quả.

Phân đoạn là quá trình phân tích và chia nội dung hình ảnh video thành các đơn vị cơ sở gọi là các lia (shot). Việc lấy mẫu chính là chọn gần đúng một khung video đại diện cho mỗi lia (hoặc nhiều hơn tùy theo mức độ phức tạp của nội dung hình ảnh của lia) và được gọi là các khung - khóa.

Khung – khóa là khung hình đại diện mô tả nội dung chính của shot.

Quá trình phân đoạn dữ liệu video tiến hành phân tích, phát hiện sự chuyển đổi từ lia này sang lia khác hay chính là sự phát hiện ranh giới giữa các lia (đó chính là

đo sự khác nhau giữa các khung hình liên kề). Trong hình dưới đây là ví dụ về sự chuyển đổi giữa các lia:

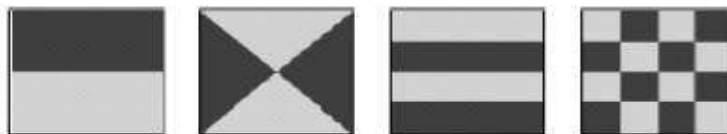


*Hình 1.2: Chuyển đổi Lia giữa khung hình thứ 3 và thứ 4*

Một số thuộc tính đặc trưng của video: Video có 4 đặc trưng chính đó là: Color (màu), Texture (kết cấu), Shape (hình dạng), Motion (chuyển động).

### **1.1.1. Màu (Color)**

Màu là một thuộc tính quan trọng của ảnh. Biểu đồ màu, biểu diễn sự phân bố màu, là một đặc trưng màu phổ biến nhất hiện nay. Biểu đồ màu không phụ thuộc vào sự quay, dịch chuyển ảnh cũng như chiều nhìn ảnh. Tính hiệu quả của nó lại phụ thuộc vào hệ màu và phương pháp định lượng được dùng. Có một vấn đề với biểu đồ màu là nó không biểu diễn thông tin về không gian phân bố các điểm ảnh, do đó hai ảnh có cùng biểu đồ màu có thể có nội dung rất khác nhau.



*Hình 1.3: Bốn khung hình khác nhau song có cùng một biểu đồ màu*

### **1.1.2. Kết cấu (Texture)**

Đây là một đặc trưng quan trọng của bề mặt, nơi xảy ra việc lặp lại mẫu cơ bản. Có hai dạng biểu diễn texture phổ biến: biểu diễn dạng ma trận đồng thời và biểu diễn Tamura. Ma trận đồng thời mô tả hướng và khoảng cách giữa các điểm ảnh, nhờ đó các thông kê có nghĩa có thể được trích chọn. Ngược lại, người ta thấy rằng entropy và mô men chênh lệch nghịch đảo lại có khả năng phân biệt tốt nhất. Biểu diễn Tamura được thúc đẩy nhờ các nghiên cứu về tâm lý trong việc thu nhận trực giác của

con người và nó bao gồm các đại lượng đo tính thô, độ tương phản, hướng, tính tròn, tính cân đối và độ ráp. Các đặc trưng Tamura rất hấp dẫn trong việc hiểu nội dung ảnh vì nó biểu đạt trực quan. Ngoài ra còn có một số các dạng biểu diễn khác như trường ngẫu nhiên Markov, biến đổi Gabor, biến đổi gợn sóng,...

### ***1.1.3. Hình dáng (Shape)***

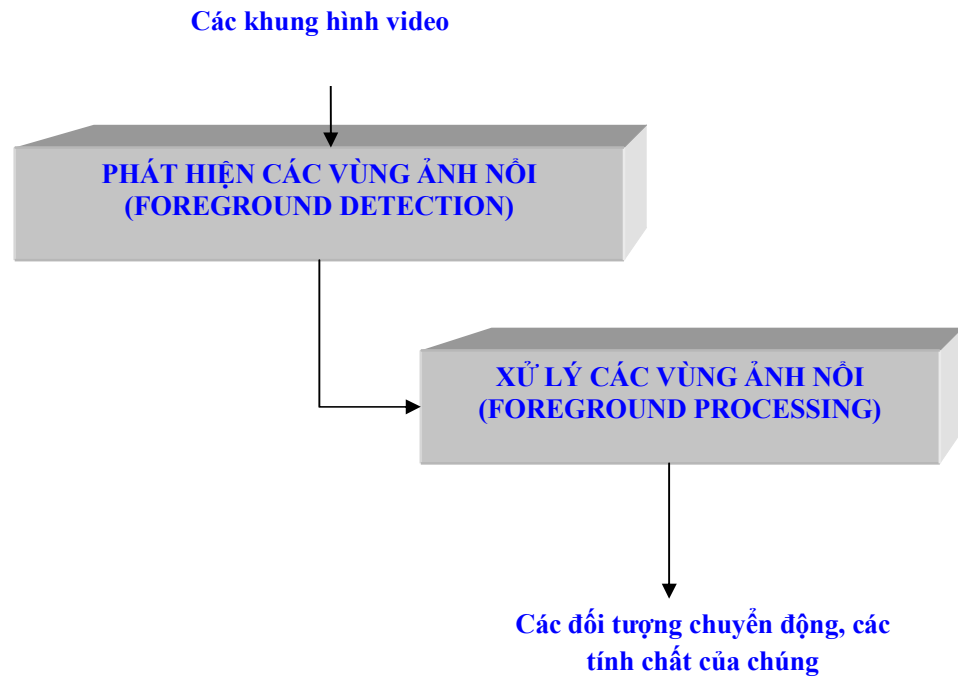
Các đặc trưng hình dáng có thể được biểu diễn sử dụng phân tích hình dáng truyền thống như bất biến mô men, mô tả Fourier, mô hình học tự động quay lui và các thuộc tính hình học. Các đặc trưng này có thể được phân chia thành đặc trưng toàn cục và đặc trưng cục bộ. Đặc trưng toàn cục là đặc trưng thuộc tính thu được từ toàn bộ hình dáng ảnh, chẳng hạn như chu vi, tính tròn, mô men trung tâm, hướng trục chính... Đặc trưng cục bộ là đặc trưng thu được từ việc thao tác với một phần của ảnh, không phụ thuộc vào toàn bộ ảnh.

### ***1.1.4. Chuyển động (Motion)***

Motion là thuộc tính quan trọng của video. Thông tin về chuyển động có thể được sinh ra bằng các kỹ thuật ghép khối hoặc luồng ánh sáng. Các đặc trưng chuyển động như mô men của trường chuyển động, biểu đồ chuyển động hoặc là các tham số chuyển động toàn cục có thể được trích chọn từ vectơ chuyển động. Các đặc trưng mức cao phản ánh di chuyển camera như quét camera (pan), nghiêng (tilt), phóng to (zoom out), thu nhỏ (zoom in) cũng có thể được trích chọn.

## **1.2. Phát hiện đối tượng chuyển động là gì?**

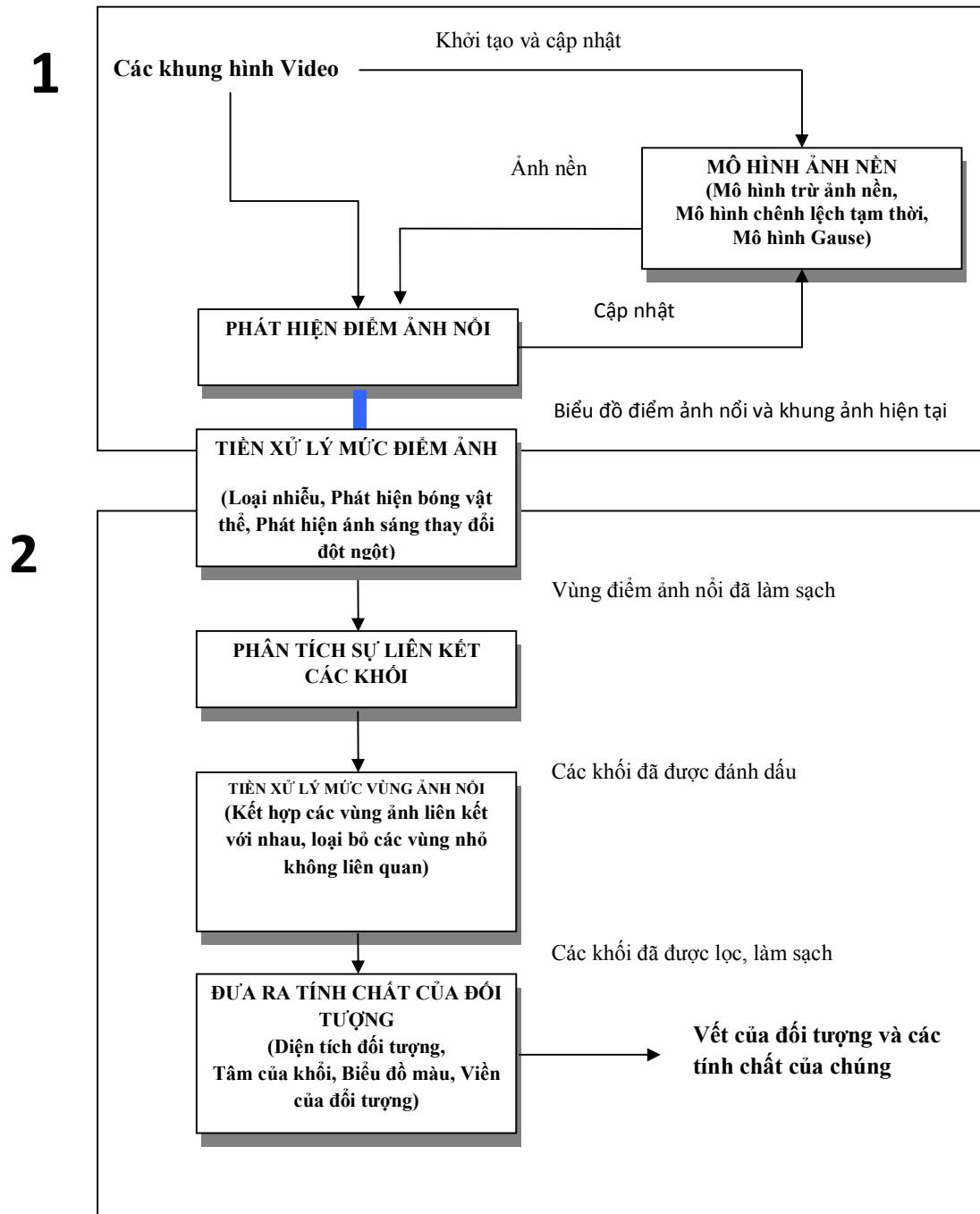
Đó là quá trình đưa ra vết các đối tượng chuyển động từ các khung hình video. Quá trình này thực chất là quá trình xử lý chuỗi ảnh liên tiếp trong một đoạn video để phát hiện ra các đối tượng chuyển động trong một đoạn hình ảnh theo mô hình dưới đây:



Hình 1.4: Tổng quan các khối xử lý trong bài toán phát hiện đối tượng

### 1.2.1. Thiết kế các khối xử lý

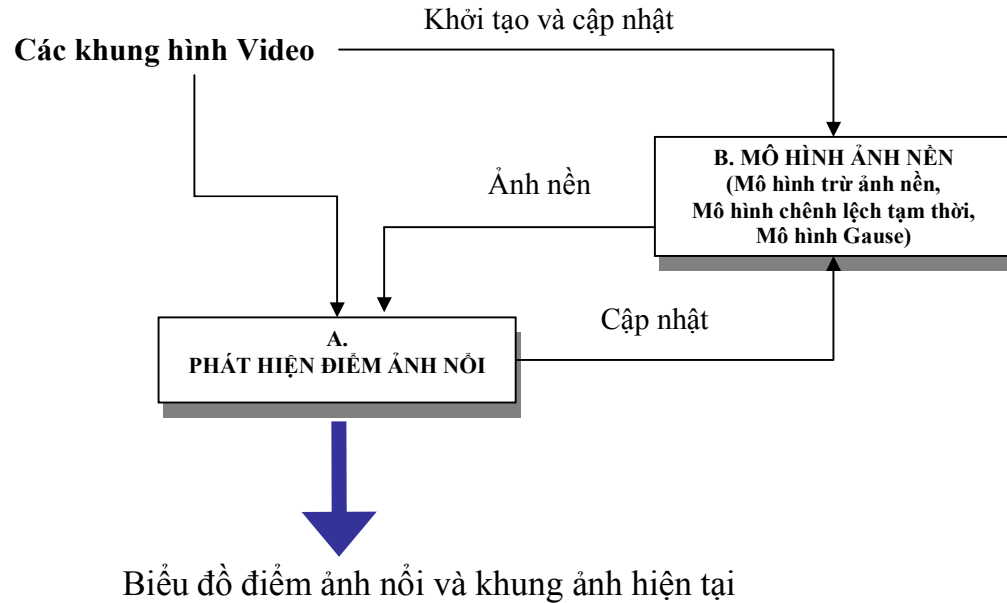
Mô hình tổng quan khối xử lý cho cả bài toán phát hiện đối tượng chuyển động



Hình 1.5: Mô hình chi tiết các khối xử lý

### 1.2.1.1. Khối phát hiện các vùng ảnh nổi

Trong phần thiết kế xử lý cho khối phát hiện vùng ảnh nổi ta sử dụng một mô hình ảnh nền (Background Model) nhằm tích hợp các phương pháp trừ ảnh nền, mô hình chênh lệch tạm thời và mô hình thống kê Gausse nhằm giải quyết vấn đề phát hiện các vùng ảnh nổi.



Hình 1.6: Khối phát hiện vùng ảnh nổi

Sơ đồ trên chỉ ra các khối xử lý nhằm phát hiện các vùng ảnh nổi. Trong đó mô hình trừ ảnh nền với các phương thức khởi tạo và cập nhật các khung hình ảnh nền nhằm đưa đầu vào: các ảnh nền cho quá trình xử lý phát hiện điểm ảnh nổi.

Sau khi đã có các ảnh nền, thao tác xử lý bên trong khối phát hiện điểm ảnh nổi đơn giản chỉ là thực hiện việc trừ ảnh, lấy khung hình hiện tại so sánh với ảnh nền để đưa ra một biểu đồ các điểm ảnh nổi lên từ ảnh hiện tại. Biểu đồ điểm ảnh này với khung ảnh hiện tại sẽ là đầu vào của quá trình xử lý tiếp theo

Trong sơ đồ xử lý này, khối xử lý mô hình ảnh nền được coi là khối xử lý quan trọng nhất. Khối này sẽ thực hiện tích hợp các phương pháp xử lý khác nhau như đã trình bày ở trên nhằm đưa ra một mô hình ảnh nền phục vụ cho bước xử lý tiếp theo. Cụ thể các mô hình được sử dụng trong khối xử lý này là:



- Tương hợp mô hình trừ ảnh nền
- Tương hợp mô hình pha trộn Gausse
- Mô hình chênh lệch tạm thời

Các mô hình này sẽ được thực hiện dưới dạng các thư viện xử lý, tích hợp vào trong hệ thống. Khi gặp các điều kiện quan sát khác nhau, hệ thống có thể linh động sử dụng các thư viện xử lý này để đưa ra hiệu quả cao nhất. Dưới đây là một số cơ sở lý thuyết về các mô hình được đề cập ở trên.

#### **1.2.1.1.1. Mô hình trừ ảnh nền thích hợp**

Thuật toán trừ ảnh nền thích hợp được thực hiện như sau:

Gọi  $I_n(x)$  là cường độ của điểm ảnh tại vị trí  $x$  và tại thời điểm khung hình thứ  $n$  trong chuỗi các khung hình video  $I(n \in [0, 255])$ . Gọi  $B_n(x)$  là giá trị cường độ của ảnh nền tương ứng cho vị trí  $x$  được ước lượng qua chuỗi hình ảnh từ  $I_0$  đến  $I_{n-1}$ . Theo phương pháp trừ ảnh thì điểm ảnh tại vị trí  $x$  trong khung hình video hiện tại thuộc lớp ảnh nổi khi nó thỏa mãn :

$$|I_n(x) - B_n(x)| > T_n(x) \quad (1.1)$$

Trong công thức này  $T_n(x)$  biểu diễn một ngưỡng được ước lượng qua các khung hình từ  $I_0$  đến  $I_{n-1}$ . Công thức trên được sử dụng để tạo ra bản đồ các vùng ảnh nổi lên. Biểu diễn các điểm ảnh bởi một ma trận trong đó các điểm ảnh nổi lên sẽ có giá trị là một và các điểm ảnh nền sẽ có giá trị là 0 cho mỗi điểm ảnh. Điểm ảnh nền  $B_n(x)$  ban đầu được khởi tạo với khung hình đầu tiên là  $B_0 = I_0$ . Do sự thay đổi của khung cảnh thu được nên khung ảnh nền và ngưỡng liên tục được cập nhật theo công thức sau:

$$B_{n+1}(x) = \begin{cases} \alpha B_n(x) + (1 - \alpha) I_n(x), & x \in BG \\ \beta B_n(x) + (1 - \beta) I_n(x), & x \in FG \end{cases} \quad (1.2)$$

$$T_{n+1}(x) = \begin{cases} \alpha T_n(x) + (1 - \alpha) I_n(x), & x \in BG \\ T_n(x), & x \in FG \end{cases} \quad (1.3)$$

Trong đó ký hiệu BG là vùng ảnh nền và FG là vùng ảnh nổi lên trên.  $\alpha$  và  $\beta$  thuộc khoảng  $[0.0, 1.0]$ ,  $\gamma$  là số điểm ảnh khác nhau từ khung hình được so sánh với ảnh nền.

$$\begin{bmatrix} 10100001000110000100010111 \\ 01000010111100111100110101 \\ 11100010010110100100110011 \\ 01000100101001000100111110 \\ 01001011011110110000000000 \\ 10001010101000100100101101 \\ 10001000100100110001001001 \end{bmatrix}$$

*Ma trận minh họa cho thao tác trừ ảnh. một đại diện cho điểm ảnh nổi, 0 đại diện cho điểm ảnh nền*

Hình minh họa dưới đây mô tả cho kết quả thuật toán trừ ảnh này:



*Hình 1.7: Minh họa thuật toán trừ ảnh*

a là ảnh nền được lấy từ khung hình đầu tiên

b là ảnh hiện tại và c là kết quả phép trừ ảnh nền

#### **1.2.1.1.2. Mô hình của Stauffer và Grimson**

Stauffer và Grimson [6] đã đưa ra phương pháp trộn các mô hình nền lại để giải quyết vấn đề ánh sáng thay đổi, các hành động lặp lại, sự lộn xộn từ khung cảnh thực tế. Mục đích của họ là chứng minh: một mô hình ảnh nền hợp nhất thì không thể xử lý được khi các khung hình liên tục trong một thời gian. Do đó, họ sử dụng phương pháp pha trộn phân tán Gauss để biểu diễn mỗi điểm ảnh trên một mô hình. Theo luận điểm đó, chúng ta thực hiện và tích hợp phương pháp này vào trong hệ thống giám sát.

Trong mô hình này, giá trị của một điểm ảnh cá biệt (độ sáng của điểm ảnh hay màu của điểm ảnh) sẽ được coi là một “pixel process” và điểm ảnh trước đó,  $\{X_1, \dots, X_t\}$ , được mô hình hoá bởi phương pháp pha trộn phân tán Gauss. Xác xuất của điểm ảnh hiện tại đang quan sát được tính bởi công thức:

$$p(X_t) = \sum_{i=1}^K w_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (1.4)$$

Trong đó  $w_{i,t}$ : Ước lượng trọng số thức  $I(G_{i,t})$  tại thời gian  $t$

$\mu_{i,t}$  là giá trị của  $G_{i,t}$

$\Sigma_{i,t}$ : Là ma trận kết hợp của  $G_{i,t}$  và  $\eta$  là hàm mật độ xác suất được xác định bởi công thức :

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)} \quad (1.5)$$

K phụ thuộc vào bộ nhớ và khả năng xử lý của máy tính do đó ma trận kết hợp được đưa ra để chỉ ra ảnh hưởng của khả năng tính toán:

$$\frac{\sum k, t = ak}{\sum k, t = a_k^2 I} \quad (1.6)$$

Với giả thiết các màu red, green, blue là các thành phần độc lập cùng biến đổi.

Thủ tục phát hiện các điểm ảnh nổi (Foreground pixel)

- Trước tiên các phân phối K Gause cho một điểm ảnh được khởi tạo với các giá trị có nghĩa, có trọng số thấp như các công thức ở trên:

- Khi có một điểm ảnh mới trong chuỗi khung hình được quan sát, để xác định kiểu của nó, vector RGB được kiểm tra lại với K Gause cho đến khi một trường hợp đúng được tìm ra. Một trường hợp đúng được xác định như một giá trị của điểm ảnh với  $y (=2.5)$  là một chuẩn phân phối.

- Bước tiếp theo, xét các phân phối  $k$  tại thời điểm  $t$ :  $w_{i,t}$  được cập nhật theo công thức:

$$W_{k,t} = (1 - \alpha) W_{k,t-1} + \alpha (M_{k,t}) \quad (1.7)$$

Với  $\alpha$  là tỉ lệ học (learning rate)

$M_{k,t} = 1$  khi đó là phân phối Gause và  $M_{k,t} = 0$  khi nó là các phân phối khác.

- Sau khi bước trên kết thúc, các trọng số phân tán được chuẩn hóa và các tham số nhằm chính xác hoá Gause sẽ được cập nhật với các quan sát tiếp theo:

$$\begin{aligned}\mu_t &= (1-\rho)\mu_{t-1} + \rho(X_t) \\ \sigma_t^2 &= (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T (X_t - \mu_t)\end{aligned}\quad (1.8)$$

Trong đó:

$$\rho = \alpha \eta (X_t | \mu_k, \sigma_k) \quad (1.9)$$

- Nếu không trường hợp nào đúng thì phân phối Gauss với xác suất thấp nhất sẽ được thay thế bởi xác suất phân bố mới với giá trị là giá trị điểm ảnh mới được thêm vào. Trong bài toán phát hiện ra loại điểm ảnh ( Là điểm ảnh nổi – foreground pixel hay điểm ảnh nền – background pixel). Phân phối K Gauss được sắp xếp bởi giá trị  $\omega/\sigma$ . Theo danh sách này các điểm ảnh thuộc ảnh nền sẽ tương ứng với xác suất phân phối có trọng số lớn và ít biến đổi. Từ đó phân phối B sẽ được chọn như là một mô hình xác định ảnh nền theo công thức:

$$B = \arg \min_b \left( \sum_{k=1}^b w_k > T \right) \quad (1.10)$$

Trong đó: T là giá trị nhỏ nhất được tìm trong mô hình ảnh nền.

### 1.2.1.1.3 Mô hình chênh lệch tạm thời

Phương pháp chênh lệch sử dụng sự so sánh mức điểm ảnh giữa hai hoặc ba khung hình liên tiếp trong một chuỗi các khung hình video liên tiếp để đưa ra vùng chuyển động của đối tượng. Phương pháp này là cách tiếp cận phù hợp với những khung cảnh hay thay đổi. Tuy nhiên, phương pháp này gặp thất bại trong việc liên kết các điểm ảnh nổi lên khi đối tượng chuyển động chậm hoặc ít thay đổi. Khi vùng điểm ảnh nổi lên ngừng chuyển động thì phương pháp chênh lệch tạm thời sẽ hoàn toàn thất bại và lúc đó đối tượng đó sẽ biến mất.

Thực hiện phương pháp này với hai khung hình liên tiếp được chỉ ra dưới đây:

Gọi  $I_n(x)$  là giá trị cường độ sáng của điểm ảnh tại vị trí (x) tại khung hình thứ n trong chuỗi các khung hình video liên tiếp I, n thuộc trong khoảng [0, 255]. Dựa vào sự chênh lệch tạm thời giữa hai khung hình một điểm ảnh được coi là chuyển động nếu thỏa mãn công thức sau

$$|I_n(x) - I_{n-1}(x)| > I_n(x) \quad (1.11)$$

Do đó nếu một đối tượng có màu đồng nhất thì theo công thức trên phương pháp này sẽ bị thất bại khi xác định vùng đối tượng chuyển động. Ngưỡng  $T_n(x)$  được định nghĩa trước theo công thức dưới đây:

$$T_{n+1}(x) = \begin{cases} \alpha T_n(x) + (1-\alpha)(\gamma^* |I_n(x) - I_{n-1}(x)|), & x \in BG \\ T_n(x), & x \in FG \end{cases} \quad (1.12)$$

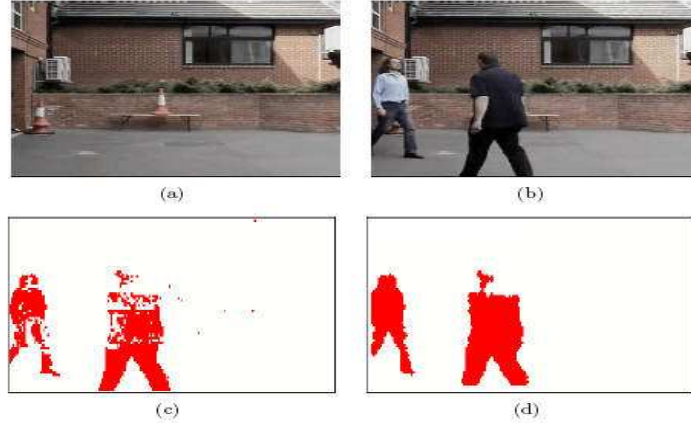
Trong đó BG là vùng ảnh nền, và FG là vùng ảnh nổi lên.  $\alpha$  thuộc khoảng  $[0,1]$  là tham số được đưa vào.

#### 1.2.1.2. Khối xử lý các vùng ảnh nổi

Đầu vào của khối xử lý này là đầu ra của khối xử lý thứ nhất: Phát hiện vùng ảnh nổi

Đầu ra của khối xử lý này cũng là đầu ra của bài toán phát hiện đối tượng, đó là đưa ra được hình dạng các đối tượng chuyển động và các tính chất của chúng như vị trí, diện tích, hình bao, màu, ...

Hình sau đây minh họa cho đầu vào, đầu ra của quá trình xử lý các vùng ảnh nổi

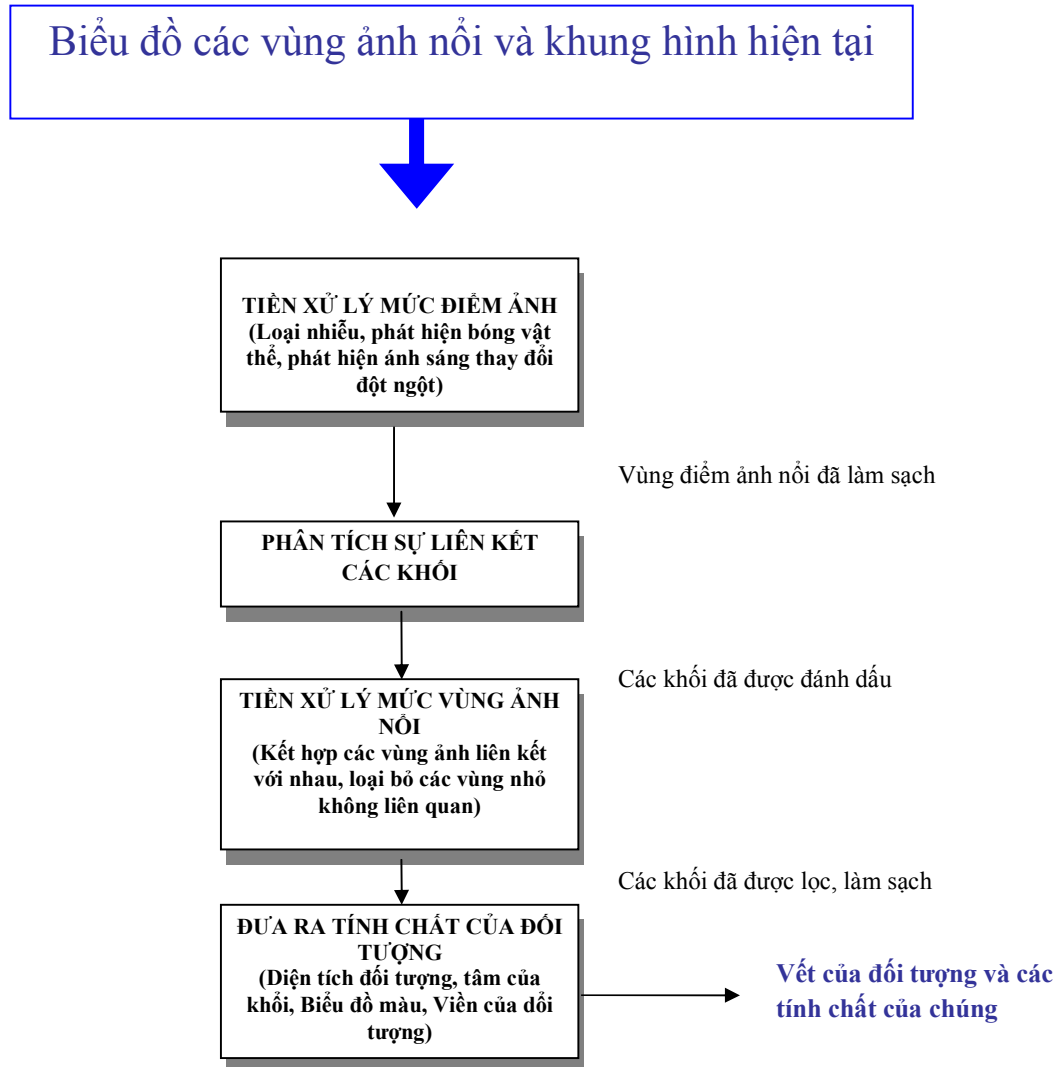


Hình 1.8: Minh họa quá trình xử lý các vùng ảnh nổi

- a. Là khung ảnh nền
- b. Là khung hình hiện tại
- c. Là đầu ra của quá trình phát hiện các vùng ảnh nổi
- d. Là đầu ra của quá trình xử lý các vùng ảnh nổi (làm sạch, loại bỏ nhiễu, đưa ra tính chất của đối tượng)

Như đã trình bày ở trên, các vùng ảnh nổi chứa các đối tượng chuyển động bao gồm rất nhiều nhiễu, các vùng vật thể không liên quan đến đối tượng chuyển động. Nhiệm vụ chính của khối xử lý này là làm sạch, loại bỏ các nhiễu, các vùng không liên quan đến đối tượng chuyển động.

Các bước xử lý được thể hiện ở sơ đồ dưới đây.



Hình 1.9: Chi tiết khối xử lý vùng ảnh nổi

#### 1.2.1.2.1. Tiền xử lý mức điểm ảnh

Như đã trình bày ở trên có nhiều yếu tố gây ra nhiễu. Áp dụng các phương pháp ăn mòn, giãn nở vào các biểu đồ các điểm ảnh nổi lên trên nhằm loại bỏ các nhiễu, chính là bước xử lý đầu tiên trong quá trình này. Mục đích của chúng ta đó là loại bỏ đi các điểm ảnh nổi được coi là nhiễu không phù hợp với các vùng ảnh nổi thực sự (NFN – non-foreground noise) và loại bỏ đi các điểm nhiễu nền (non-

background noise, NBN) gần với vùng vật thể vùng thực sự là các điểm ảnh nổi. Thực hiện phương pháp ăn mòn bằng cách bảo đi các điểm ảnh trên một đường biên bao quanh vùng điểm ảnh nổi còn ngược lại thực hiện giãn nở là thực hiện mở rộng đường biên của các vùng ảnh nổi bằng các điểm ảnh nổi mới.

Các vấn đề liên quan đến bóng và loại bỏ nhiễu khi ánh sáng thay đổi

Hầu hết các thuật toán phát hiện vùng ảnh nổi đều bị ảnh hưởng bởi cả hai yếu tố bóng của đối tượng và sự thay đổi ánh sáng. Đó là những nguyên nhân gây ra nhập nhằng trong việc phân đoạn các đối tượng là các vùng ảnh nổi.

Bước xử lý này có ảnh hưởng lớn đến các bài toán tiếp theo như phân loại đối tượng và theo vết đối tượng nên bước xử lý bóng của vật thể. Nhiều do thay đổi ánh sáng được đánh giá là rất quan trọng trong hệ thống xử lý và theo dõi đối tượng thông minh.

Trong phần này ta sử dụng một phương pháp xử lý phát hiện bóng dựa trên một nguyên lý là các điểm ảnh trong vùng là bóng của vật thể sẽ có vector màu RGB cùng hướng với vector màu của các điểm ảnh nền tương ứng với độ chênh lệch rất nhỏ, độ sáng của điểm ảnh là bóng luôn nhỏ hơn độ sáng của điểm ảnh nền tương ứng.

Giả sử  $I_x$  biểu diễn màu của điểm ảnh ở khung hình tại vị trí  $x$ , và  $B_x$  biểu diễn màu RGB của điểm ảnh nền tương ứng.  $\vec{I}_x$  là vector có gốc là  $0(0,0,0)$  trong hệ tọa độ màu RGB và  $\vec{B}_x$  là vector tương ứng cho điểm ảnh  $B_x$

Gọi  $d_x$  được tính theo công thức :

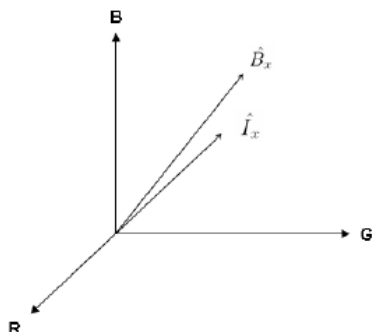
$$\left( d_x = \frac{\vec{I}_x \cdot \vec{B}_x}{\|\vec{I}_x\| \cdot \|\vec{B}_x\|} \right) \quad (1.13)$$

Điểm ảnh được coi là bóng của đối tượng khi nó thỏa mãn công thức sau:

$$\left( d_x = \frac{\vec{I}_x \cdot \vec{B}_x}{\|\vec{I}_x\| \cdot \|\vec{B}_x\|} \right) < \tau \quad (1.14)$$

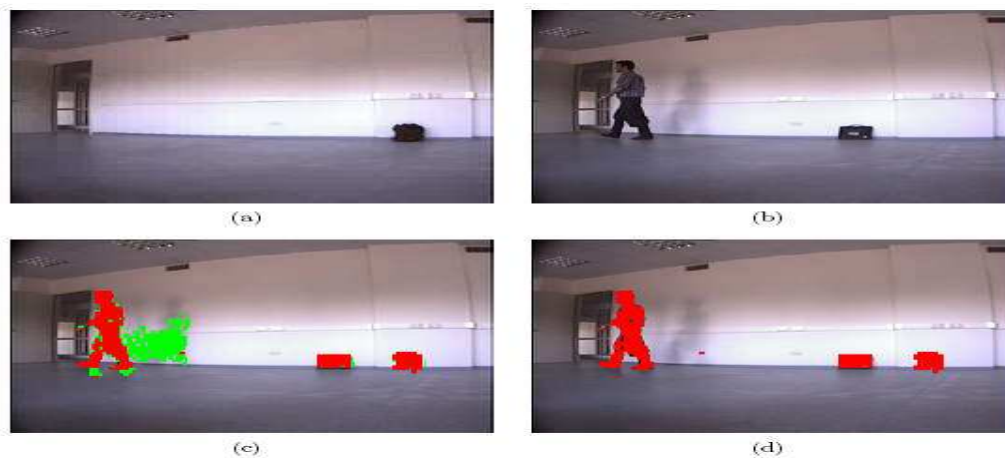
$$\text{và} \quad \|\vec{I}_x\| < \|\vec{B}_x\| \quad (1.15)$$

Ở đây  $\tau$  là một ngưỡng được xác định trước.  $dx$  được sử dụng để kiểm tra liệu  $\vec{F}_x$  và  $\vec{B}_x$  có cùng hướng hay không. Nếu  $dx$  thỏa mãn công thức trên thì khi đó ta sẽ suy ra chúng cùng một hướng với sự chênh lệch rất nhỏ.



Hình 1.10: Vector RGB cho điểm ảnh ở vị trí  $x$  của khung hình hiện tại và vector cho điểm ảnh tương ứng ở khung ảnh nền.

Hình dưới đây cho ta một ví dụ về xử lý loại bỏ bóng của vật thể



Hình 1.11: Xử lý loại bỏ bóng

- Khung hình nền
- Khung hình hiện tại
- Kết quả sau khi phát hiện các vùng ảnh nổi (chứa cả bóng của đối tượng)
- Kết quả xử lý sau khi loại bỏ bóng



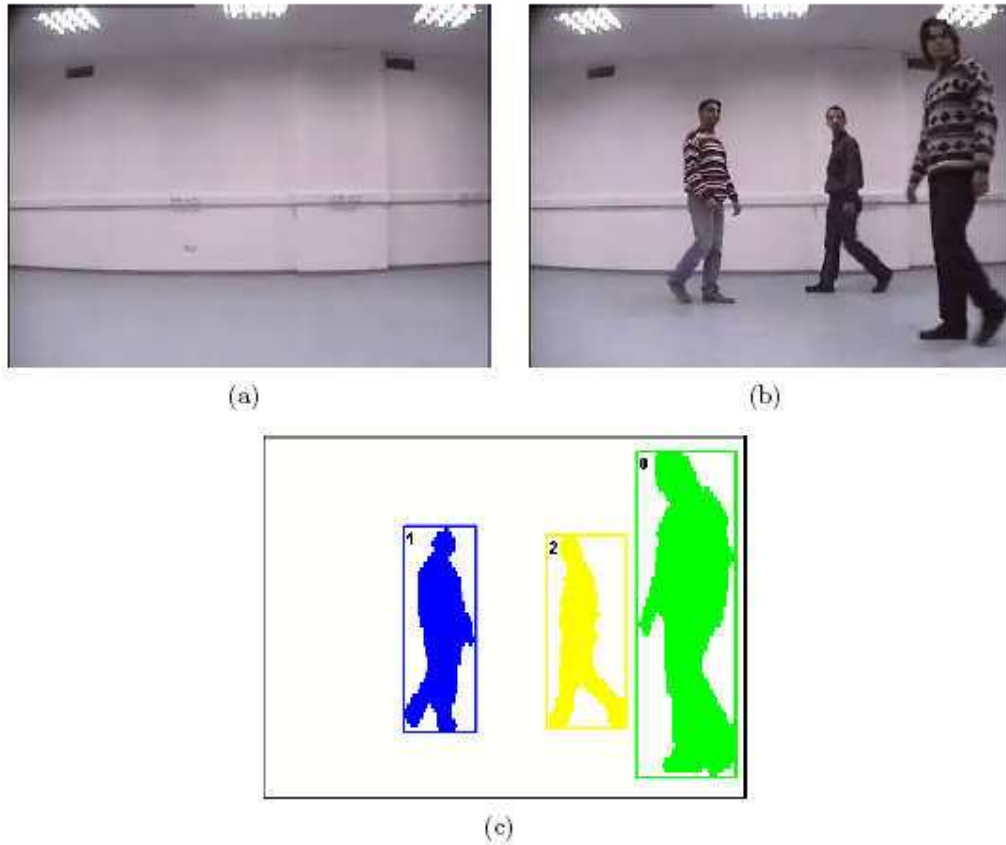
Bên cạnh các yếu tố nhiễu do bóng của đối tượng, yếu tố ánh sáng thay đổi do điều kiện quan sát cũng là một yếu tố đáng quan tâm và xử lý khi thực hiện quan sát ngoài trời. Ta biết rằng môi trường ngoài trời luôn luôn thay đổi, khi mây bay hay gió thổi đều ảnh hưởng rất lớn đến kết quả quan sát và xử lý.



*Hình 1.12: Hai khung cảnh liên tiếp khi ánh sáng thay đổi.*

#### **1.2.1.2.2. Phát hiện các sự liên kết giữa các khối**

Sau bước phát hiện các vùng ảnh nổi và qua xử lý ở mức điểm ảnh để loại bỏ nhiễu và các vùng bóng của vật thể, các vùng ảnh được nhóm lại theo thành các vùng liên kết (blobs) được dán nhãn bằng cách sử dụng thuật toán dán nhãn được trình bày trong tài liệu tham khảo. Sau khi tìm ra các đốm màu độc lập tương ứng với các đối tượng ta sẽ thực hiện tính hộp bao của vùng này. Hình dưới đây sẽ chỉ ra ví dụ về xử lý sau khi đã liên kết các vùng ảnh nổi dán nhãn và bao hộp.



Hình 1.13: Liên kết các khối ảnh nổi

- a. Khung hình nền
- b. Khung hình hiện tại
- c. Kết quả sau khi thực hiện xử lý liên kết các khối, dán nhãn và bao hộp.

#### 1.2.1.2.3. Tiền xử lý ở mức khối ảnh nổi

Sau bước loại bỏ nhiễu, đôi khi các vùng nhỏ vẫn còn do không chính xác trong bước phân đoạn đối tượng. Để xử lý vấn đề này chúng ta ước lượng kiểu của các vùng ảnh, kích cỡ trung bình ( $\gamma$ ) trong phạm vi của các điểm ảnh được tính cho mỗi khung hình và các vùng có kích thước nhỏ hơn một mẫu ( $\alpha$ ).

Nếu ( $\text{Size}(\text{region}) < \gamma * \alpha$ ) thì vùng đó sẽ được loại bỏ

Ngoài ra, do một số lỗi phân đoạn, một số phần của vật thể bị tách biệt với vùng chính. Với những trường hợp này dùng hộp bao các vùng này là cách giải quyết tốt nhằm gộp chúng lại lại và thực hiện dán nhãn.

#### 1.2.1.2.4. Đưa ra tính chất của các đối tượng được phát hiện

Khi đã có các vùng ảnh được phân đoạn chúng ta sẽ thực hiện xuất ra các tính chất tương ứng của đối tượng từ khung hình hiện tại. Các tính chất này bao gồm: Kích cỡ (s), Trọng tâm của khối ( $C_m$ ), Biểu đồ màu ( $H_c$ ) và hình chiếu của các vết đối tượng.

Để tính kích cỡ S chúng ta chỉ việc thực hiện đếm số lượng điểm ảnh nổi trong hình bao của đối tượng.

Tính trọng tâm khối  $C_m = (X_m, Y_m)$  của đối tượng O theo công thức sau :

$$X_{Cm} = \frac{\sum_i^n Xi}{n}, Y_{Cm} = \frac{\sum_i^n Yi}{n} \quad (1.16)$$

Trong đó n là số lượng điểm ảnh của đối tượng O

Biểu đồ màu  $H_c$  được tính qua độ đơn sắc của các điểm ảnh trong khung hình hiện tại. Để thực hiện giảm sự tính toán phức tạp ta sử dụng  $H_c$  giá trị màu đã được lượng tử hóa theo 255/N giá trị màu. Biểu đồ màu được tính toán việc lặp đi lặp lại các điểm ảnh của O và tăng giá trị lưu trữ của lô màu tương ứng trong biểu đồ  $H_c$ . Từ đó một biểu đồ màu của đối tượng O sẽ được cập nhật qua công thức:

$$H_c \left[ \frac{C_i}{N} \right] = H_c \left[ \frac{C_i}{N} \right] + 1, \forall C_i \in O \quad (1.17)$$

Ở đây  $C_i$  biểu diễn giá trị màu tại điểm ảnh thứ i. Trong bước tiếp theo, biểu đồ màu sẽ được xem xét để có thể đưa ra so sánh phù hợp với các biểu đồ màu khác trong bước tiếp theo. Theo công thức sau:

$$\bar{H}_c[i] = \frac{H_c[i]}{\sum_i^N H_c[i]} \quad (1.18)$$

Qua phương pháp tính này ta có thể có một số nhận xét đó là: Cách tính này sẽ gặp thất bại trong các trường hợp các đối tượng có cùng một biểu đồ màu ví dụ như hai đối tượng có cùng màu quần áo, hoặc người mặc áo sơ mi trắng, quần đen và người mặc áo sơ mi đen, quần trắng. Cả hai sẽ cho kết quả biểu đồ màu như nhau. Nếu xét về biểu đồ màu thì hai đối tượng này hoàn toàn sẽ bị hiểu là cùng một đối tượng.

### 1.3. Giới thiệu bài toán phát hiện chuyển động

Ra đời từ những năm 1960 qua quá trình hoàn thiện và phát triển, ngày nay một hệ thống giám sát chuyển động là một trong những hệ thống trợ giúp đắc lực nhất cho con người thực hiện theo dõi, giám sát. Từ các hình ảnh thu được từ những nơi được quan sát, ta có thể phát hiện được chuyển động của các đối tượng trong các khung hình, có thể xác định được đối tượng đó là người, phương tiện hay vật thể gì. Minh họa rõ hơn, với một bài toán giám sát giao thông một hệ thống giám sát thông minh có thể cho chúng ta biết được số lượng phương tiện lưu thông qua đoạn đường được theo dõi, đưa ra thông tin về tốc độ chuyển động, đường đi của đối tượng được theo dõi. Với một bài toán phát hiện, dự đoán, hệ thống giám sát thông minh có thể phát hiện một đám cháy, tự động cảnh báo cháy ở nơi được quan sát và theo dõi.

Hiện nay, trên thế giới các hệ thống giám sát thông minh bằng video đã được phát triển và đã chứng minh được hiệu quả nhất định trên một số lĩnh vực như giám sát hoạt động con người, giám sát giao thông, ...Song các hệ thống vẫn gặp phải một số tồn tại như hiệu quả của việc quan sát luôn phụ thuộc vào điều kiện môi trường quan sát, kiểu chuyển động của đối tượng hay các lý do khách quan khác. Chính vì thế mà việc nghiên cứu và phát triển các giải pháp, công nghệ mới vẫn đang được tiến hành nhằm đáp ứng được yêu cầu về một hệ thống giám sát thông minh hoàn thiện nhất.

Bên cạnh sự phát triển của các thiết bị phần cứng, hệ thống giám sát đòi hỏi con người thường trực 24/24 để giám sát các màn hình theo dõi. Một phút lơ là của người bảo vệ có thể để đối tượng đột nhập vào mục tiêu. Kỹ thuật xử lý ảnh số thu được từ các thiết bị quan sát sẽ tự động phân tích và đưa ra cảnh báo khi có một đối tượng đột nhập vào mục tiêu, đồng thời hệ thống sẽ lưu lại hình ảnh đối tượng và theo dấu đối tượng trong vùng quan sát. Căn cứ vào cảnh báo tự động đó, người bảo vệ sẽ biết để có biện pháp xử lý đảm bảo mục tiêu an toàn.

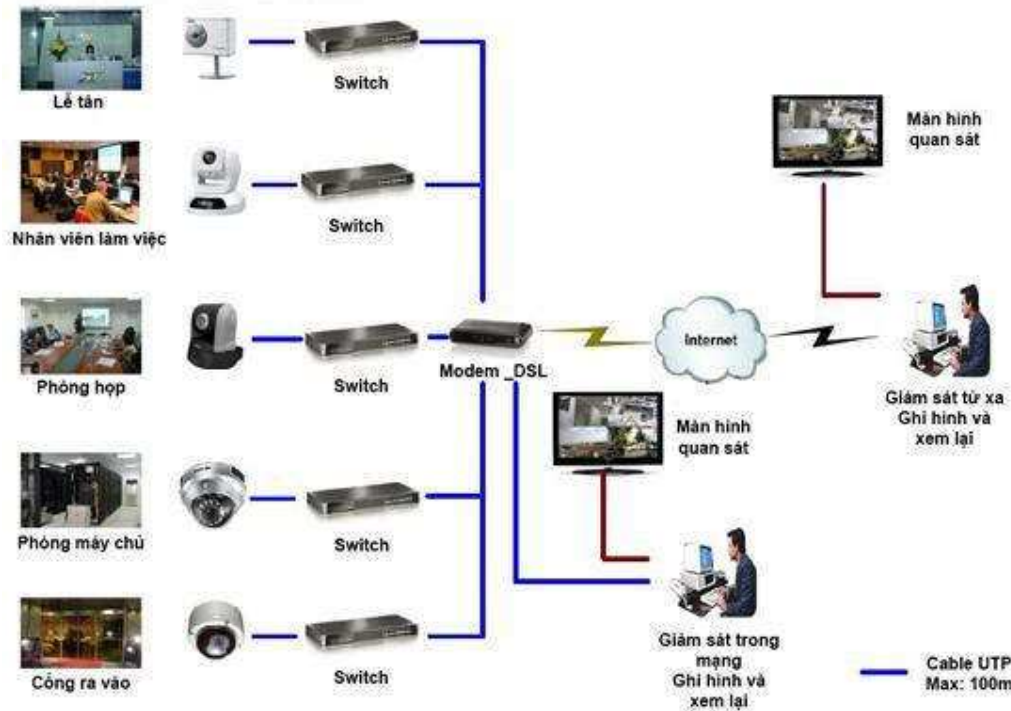


Hình 1.14. Minh họa việc sử dụng camera giám sát

**Một hệ thống giám sát bằng camera bao gồm các thành phần sau:**

- Các camera giám sát đặt tại các địa điểm ta cần quan sát: Cửa, hành lang, cầu thang, trong phòng...
- Máy tính để xử lý và lưu trữ dữ liệu hình ảnh do camera thu nhận được.
- Các thiết bị hiển thị màn hình, thiết bị truyền dữ liệu. Tín hiệu có thể được truyền trên mạng Internet phục vụ yêu cầu giám sát từ xa.
- Phần mềm chạy trên máy tính để xử lý dữ liệu thu được từ camera và đưa ra các quyết định.

Trong luận văn của mình, học viên tập trung nghiên cứu các phương pháp xử lý ảnh, video kỹ thuật số để tự động phát hiện, cảnh báo khi có đối tượng đột nhập thông qua hệ thống camera theo dõi. Để phát hiện được đối tượng chuyển động, trước hết bài toán đặt ra là phát hiện tất cả các chuyển động trong vùng camera theo dõi, sau đó phân tích và đưa ra những cảnh báo khác nhau khi đối tượng vào những khu vực giám sát. Bài toán phát hiện chuyển động (Motion Detection) là bài toán cơ sở, mà từ đó con người đã xây dựng rất nhiều ứng dụng như: Hệ thống giám sát bảo vệ mục tiêu, hệ thống giám sát và phân luồng giao thông, phân tích cử động của con người trong nghiên cứu chế tạo robot...



Hình 1.15. Sơ đồ hệ thống giám sát bằng camera theo dõi

Bài toán phát hiện chuyển động đã được thế giới nghiên cứu từ rất sớm. Cho tới thời điểm hiện nay, đã có nhiều giải thuật phát hiện chuyển động được công bố, một số giải thuật mới được công bố trong những năm qua đã được chứng minh có độ chính xác tương đối cao, thời gian tính toán thấp và xử lý được môi trường biến động, nhiều đối tượng chuyển động.

Có rất nhiều cách tiếp cận bài toán phát hiện chuyển động khác nhau. Nhưng quy về mức tổng quát, có ba phương pháp phát hiện chuyển động chính hiện nay là: Phương pháp trừ nền - Background Subtraction, Phương pháp dựa vào sự biến đổi giá trị theo thời gian - Temporal Differencing và Phương pháp luồng quang học - Optical Flow, cụ thể như sau:

- Phương pháp trừ nền: Đây là phương pháp thông dụng nhất, phương pháp này xác định sự sai khác do đối tượng chuyển động tạo ra bằng các trừ từng bit tương ứng của hai frame liên tiếp nhau trong chuỗi video thu được từ camera. Phương pháp này thích ứng với môi trường động có nhiều đối tượng chuyển động phức tạp, phát hiện chuyển động biên độ nhỏ và ít phụ thuộc vào vận tốc cũng như kích thước của đối tượng chuyển động.

- Phương pháp dựa vào sự biến đổi giá trị theo thời gian: Phương pháp này dựa vào sự biến đổi của một giá trị ảnh theo thời gian để xác định đối tượng chuyển động (Ví dụ: Temporal Gradient – dựa trên biến đổi của Gradient theo thời gian. Khả năng chuyển động của đối tượng theo phương pháp này được đo bằng sự thay đổi tức thời của cường độ hình ảnh). Phương pháp này thích ứng với sự thay đổi của môi trường, nhưng nó bị phụ thuộc vào tốc độ và kích thước và số lượng đối tượng chuyển động. Phương pháp này được áp dụng hiệu quả nhất trong trường hợp phát hiện và theo dõi một đối tượng chuyển động.

- Phương pháp luồng quang học: Phương pháp này chỉ ra sự chuyển động dự kiến của đối tượng trong ảnh. Phương pháp cho kết quả khá tốt với những ảnh đầu vào phức tạp. Tuy nhiên, nó đòi hỏi độ phức tạp tính toán cao. Chính điều này đã tạo ra khó khăn trong việc áp dụng, triển khai thực tế.

**Một hệ thống giám sát bằng camera có thể được mô tả bằng Sơ đồ tổng quát sau:**

*Input:*

Chuỗi video đầu vào thu được từ camera giám sát.

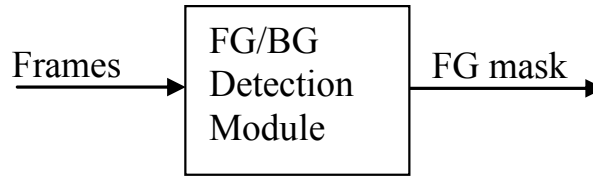
*Output:*

- Phát hiện và cảnh báo đối tượng chuyển động.
- Trích xuất đối tượng chuyển động.
- Theo dõi đối tượng chuyển động.

Chuỗi video đầu vào được hình thành từ những frame khác nhau tùy thuộc vào thông số kỹ thuật của camera (16fps, 24fps, 25fps...). Giải thuật sẽ tính toán và xử lý trên những frame này.

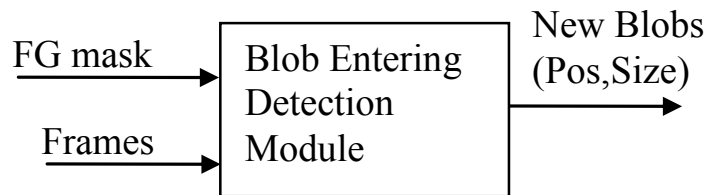
**Các chức năng chính của hệ thống giám sát bằng camera:**

(1)- Xác định các vùng có khả năng chứa đối tượng chuyển động - Mặt nạ vùng chuyển động:



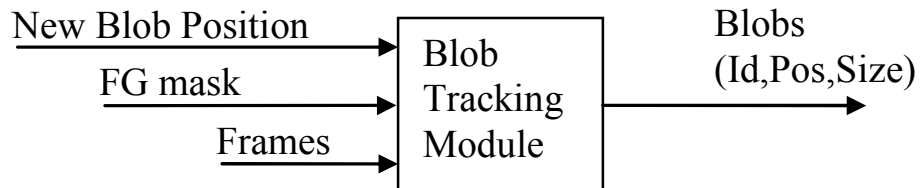
Hình 1.16. Sơ đồ tính mặt nạ vùng chuyển động (foreground mask detection)

(2)- Trên cơ sở mặt nạ vùng chuyển động, ta sẽ xác định các đối tượng chuyển động (vị trí, kích thước)



Hình 1.17. Sơ đồ phát hiện các đối tượng chuyển động

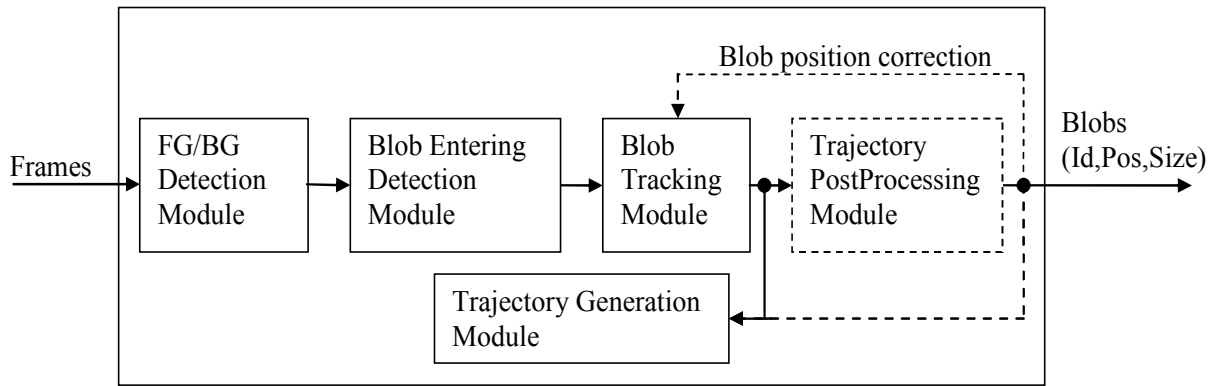
(3)- Khi phát hiện được đối tượng chuyển động, ta thực hiện chức năng theo dõi đối tượng chuyển động. Đối tượng chuyển động sẽ được theo dõi khi chuyển động trong vùng camera theo dõi:



Hình 1.18. Sơ đồ khối của chức năng tracking 1 hoặc 1 nhóm đối tượng

(4)- Tổng hợp hệ thống phát hiện và giám sát đối tượng chuyển động





Hình 1.19. Sơ đồ khối hệ thống phát hiện và giám sát đối tượng chuyển động

FG mask: foreground mask – Mặt nạ vùng chuyển động.

FG/BG: foreground/background

Foreground: Vùng chứa đối tượng chuyển động

Background: Nền (Vùng không chứa đối tượng chuyển động)

Blob: Đối tượng chuyển động

**Yêu cầu của bài toán phát hiện chuyển động là:**

- Phát hiện được tất cả các chuyển động trong vùng camera giám sát.
- Hệ thống phải xử lý được trong thời gian thực với độ trễ có thể chấp nhận được, có nghĩa là giải thuật tối ưu phải đạt được: tốc độ tính toán chấp nhận được, yêu cầu về bộ nhớ thấp...
- Hệ thống có tính linh hoạt, tương thích với các điều kiện, môi trường quan sát khác nhau.

## Chương 2

### NỘI DUNG CHÍNH CỦA GIẢI THUẬT PHÁT HIỆN CHUYỂN ĐỘNG

#### 2.1. Một số giải thuật phát hiện chuyển động

Phương pháp trừ nền áp dụng trong bài toán phát hiện chuyển động nghiên cứu để tạo ra những mô hình nền tối ưu nhằm tăng khả năng phát hiện đối tượng chuyển động cũng như giảm bớt độ phức tạp tính toán của giải thuật. Một số giải thuật sử dụng phương pháp trừ nền đã được nghiên cứu và công bố như: Giải thuật trừ nền cơ bản - Simple background Subtraction; Giải thuật trừ nền trung bình - Running Average; Giải thuật  $\Sigma$ - $\Delta$  -  $\Sigma$ - $\Delta$  Estimation; Giải thuật  $\Sigma$ - $\Delta$  cải tiến - Multiple  $\Sigma$ - $\Delta$  Estimation; Giải thuật thống kê khác biệt cơ bản - Simple Statistical Difference; Giải thuật trừ nền trung bình với biến đổi cosine rời rạc - Running Average with Discrete Cosine Transform.

##### 2.1.1. Giải thuật trừ nền cơ bản (Simple Background Subtraction):

Phương pháp trừ nền cơ bản là phương pháp so sánh ảnh đơn giản nhất. Phương pháp này dựa trên sự sai khác giữa hai ảnh (trừ hai ảnh theo từng bit tương ứng) và so sánh sự sai khác này với một giá trị ngưỡng cho trước. Trường hợp sự sai khác này lớn hơn giá trị ngưỡng đã cho, có nghĩa đã có sự sai khác về các pixel ảnh. Ta đưa ra kết luận có đối tượng chuyển động.

##### Thuật toán tổng quát đối với pixel (x,y):

Chuỗi video đầu vào.

$\tau$  : Giá trị ngưỡng cho trước

$B(x,y)$ : Giá trị nền cố định

$I_t(x,y)$ : Giá trị frame video đến, t là chỉ số frame thuộc: 0..K

$B(x,y) = I_0(x,y)$  (Đặt frame đầu tiên làm nền).

$D(x,y)$ : Mặt nạ nhị phân phát hiện chuyển động được định nghĩa như sau:

$$D(x,y) = \begin{cases} 1, & \text{if } |I_t(x,y) - B(x,y)| > \tau \\ 0, & \text{if } |I_t(x,y) - B(x,y)| \leq \tau \end{cases}$$

$D(x,y) = 0$  (hiệu hai frame nhỏ hơn giá trị ngưỡng  $\tau$ ) đưa ra kết luận đây là những pixel của nền, ngược lại,

$D(x,y) = 1$  (hiệu hai frame lớn hơn ngưỡng  $\tau$  cho trước) đưa ra kết luận đây là những pixel của đối tượng chuyển động. Tập hợp các pixel này ta sẽ có hình ảnh của đối tượng chuyển động.

### **Đánh giá:**

*Ưu điểm:* Thuật toán rất đơn giản, thời gian tính toán nhanh. Thuật toán này rất hữu dụng trong trường hợp chỉ cần xác định những thay đổi bộ phận, mà không phải xác định sự thay đổi của toàn bộ khung hình.

*Nhược điểm:* Độ chính xác của thuật toán này thấp. Đặc biệt, thuật toán này đưa ra kết quả không chính xác đối với những trường hợp đối tượng trong đoạn video hầu như không di chuyển, camera thu ảnh có nhiễu nhiều, không thể phát hiện được sự thay đổi lớn trong một vùng ảnh nhỏ và thay đổi nhỏ trong một vùng ảnh lớn.

### **2.1.2. Giải thuật trừ nền trung bình (Running Average):**

Phương pháp trừ nền trung bình là một cải tiến của phương pháp trừ nền cơ bản. Thay vì giữ nguyên giá trị khung hình nền  $B(x,y)$  trong phép trừ nền, phương pháp trừ nền trung bình cải tiến bằng việc cập nhật liên tục giá trị khung hình nền nhằm tăng tính chính xác cho kết quả phát hiện đối tượng chuyển động.

#### **Thuật toán tổng quát đối với pixel $(x,y)$ :**

$\tau$  : Giá trị ngưỡng cho trước

$\beta$  : Tham số cập nhật nền cho trước  $\in [0,1]$

$B_t(x,y)$ : Giá trị khung nền

$I_t(x,y)$ : Giá trị khung video đến

Giá trị khởi tạo mô hình nền và frame video đến:  $B_0(x,y) = I_0(x,y)$

Bước 1: Cập nhật giá trị nền  $B_t(x,y)$  theo chỉ số frame  $t$ :

$$B_t(x, y) = (1 - \beta)B_{t-1}(x, y) + \beta I_t(x, y)$$

Bước 2: Tính mặt nạ nhìn phân phát hiện chuyển động  $D(x,y)$ :

$$D(x, y) = \begin{cases} 1, & \text{if } |I_t(x, y) - B_t(x, y)| > \tau \\ 0, & \text{if } |I_t(x, y) - B_t(x, y)| \leq \tau \end{cases}$$

$D(x, y) = 0$  (hiệu hai khung hình nhỏ hơn giá trị ngưỡng  $\tau$ ) đưa ra kết luận đây là những pixel của nền, ngược lại,

$D(x, y) = 1$  (hiệu hai khung hình lớn hơn ngưỡng  $\tau$  cho trước) đưa ra kết luận đây là những pixel của đối tượng chuyển động. Tập hợp các pixel này ta sẽ có hình ảnh của đối tượng chuyển động

### **Đánh giá:**

*Ưu điểm:* Việc cập nhật liên tục giá trị nền thông qua việc sử dụng tham số  $\beta$  đã làm tăng độ chính xác cho việc phát hiện đối tượng chuyển động so với thuật toán trừ nền cơ bản. Cài đặt giải thuật đơn giản, độ phức tạp tính toán đơn giản, tốc độ xử lý nhanh.

*Nhược điểm:* Phương pháp này chưa giải quyết được những hạn chế của phương pháp trừ nền cơ bản đó là: kết quả phát hiện không chính xác đối với những trường hợp đối tượng trong đoạn video hầu như không di chuyển, camera thu ảnh có nhiễu nhiều, không thể phát hiện được sự thay đổi lớn trong một vùng ảnh nhỏ và thay đổi nhỏ trong một vùng ảnh lớn.

### **2.1.3 Giải thuật $\Sigma$ - $\Delta$ ( $\Sigma$ - $\Delta$ Estimation):**

Giải thuật  $\Sigma$ - $\Delta$  dựa trên phương pháp đệ quy không tuyến tính đơn giản (còn được gọi là bộ lọc  $\Sigma$ - $\Delta$ ). Giải thuật  $\Sigma$ - $\Delta$  sử dụng hàm  $\text{sgn}(a)$  để ước lượng giá trị của nền,  $\text{sgn}(a)$  được định nghĩa như sau:

$$\text{sgn}(a) = \begin{cases} 1, & \text{if } a > 0 \\ 0, & \text{if } a = 0 \\ -1 & \text{if } a < 0 \end{cases}$$

$a$ : là giá trị thực cho trước

Khi đó mô hình nền sẽ được cập nhật theo hàm  $\text{sgn}(a)$  như sau:

$$B_t(x, y) = B_{t-1}(x, y) + \text{sgn}(I_t(x, y) - B_{t-1}(x, y))$$

$B_t(x, y), B_{t-1}(x, y)$ : Tương ứng là giá trị nền hiện tại thứ  $t$  và trước đó ( $t-1$ ).

$I_t(x, y)$ : Tương ứng là giá trị khung video đến hiện tại thứ  $t$ .

$B_0(x, y) = I_0(x, y)$ : Giá trị khởi tạo với frame đầu tiên

Như vậy, đối với mỗi frame, giá trị nền được tăng hoặc giảm hoặc giữ nguyên tùy thuộc vào giá trị của hàm  $\text{sgn}(a)$ .

Tính giá trị hiệu tuyệt đối giữa  $B_t(x, y)$  và  $I_t(x, y)$  như sau:

$$\Delta_t(x, y) = |I_t(x, y) - B_t(x, y)|$$

Tính ngưỡng cho việc phát hiện đối tượng chuyển động: Sử dụng biến  $V_t(x, y)$  - Biến theo thời gian (time-variance) được tính bằng việc ứng dụng hàm  $\text{sgn}(a)$ . Biến  $V_t(x, y)$  nhằm xác định xem mỗi pixel của một frame đang xét sẽ là pixel “nền” hay pixel của “đối tượng chuyển động”.

Khởi tạo, ta có  $V_0(x, y) = \Delta_0(x, y) = 0$ : Giá trị khởi tạo với frame đầu tiên, công thức tính  $V_t(x, y)$  như sau :

$$V_t(x, y) = V_{t-1}(x, y) + \text{sgn}(N \times \Delta_t(x, y) - V_{t-1}(x, y))$$

$V_t(x, y), V_{t-1}(x, y)$  tương ứng là giá trị biến theo thời gian hiện tại thứ  $t$  và trước đó thứ ( $t-1$ ).

$N$  là tham số cho trước trong khoảng từ 1 – 4

Mật độ nhị phân phát hiện đối tượng chuyển động  $D(x, y)$  được tính như sau:

$$D(x, y) = \begin{cases} 1, & \text{if } \Delta_t(x, y) > V_t(x, y) \\ 0, & \text{if } \Delta_t(x, y) \leq V_t(x, y) \end{cases}$$

Nếu  $D(x, y) = 0 \Rightarrow$  đây là những pixel của nền do đó sẽ không có đối tượng chuyển động, ngược lại.

Nếu  $D(x, y) = 1 \Rightarrow$  đây là những pixel của đối tượng chuyển động. Tập hợp những pixel này ta có được hình ảnh của đối tượng chuyển động.

**Đánh giá:**

*Ưu điểm:* Bằng việc sử dụng thuật toán được chuẩn hóa theo thời gian, Giải thuật  $\Sigma$ - $\Delta$  cho phép xử lý nhanh, tăng hiệu quả tính toán và độ chính xác hơn trong phát hiện đối tượng chuyển động so với hai giải thuật đã trình bày ở phần trên.

*Nhược điểm:* Hạn chế của phương pháp này là khả năng phát hiện kém đối với những đối tượng chuyển động trong cảnh phức tạp, có chứa nhiều đối tượng chuyển động, đồng thời mức độ và thời gian chuyển động của các đối tượng là khác nhau.

#### 2.1.4. Giải thuật $\Sigma$ - $\Delta$ cải tiến (Multiple $\Sigma$ - $\Delta$ Estimation):

Giải thuật  $\Sigma$ - $\Delta$  cập nhật mô hình nền theo hằng số thời gian  $\text{sgn}(\alpha)$ . Điều này tạo ra hạn chế đối với những ảnh chứa nhiều đối tượng chuyển động hoặc đối tượng có nhiều chuyển động. Giải thuật  $\Sigma$ - $\Delta$  cải tiến được đề xuất để giải quyết bài toán nhiều đối tượng và nhiều chuyển động. Phương pháp này sử dụng mô hình nền thích ứng (adaptive background model) để tăng khả năng phát hiện các chuyển động trong một ảnh phức tạp.

Việc xử lý theo thời gian có thể cho ra kết quả phát hiện chuyển động rất hiệu quả trong trường hợp đối tượng chuyển động chậm dần, dừng lại hoặc quay vòng. Tuy nhiên, do giải thuật  $\Sigma$ - $\Delta$  đặc trưng bởi khoảng thời gian cố định: cập nhật theo giai đoạn và độ lớn số lượng mức xám trên một giây. Đây là lý do gây ra hạn chế của giải thuật  $\Sigma$ - $\Delta$  trong việc thích ứng với khung hình phức tạp nhất định.

Giải thuật  $\Sigma$ - $\Delta$  cải tiến thay vì tính một nền riêng lẻ, chúng ta sẽ tính một tập các nền:  $\{b_i^i\}_{1 \leq i \leq K}$ . Công thức tính như sau:

$$b_t^i(x, y) = b_{t-1}^i(x, y) + \text{sgn}(b_t^{i-1}(x, y) - b_{t-1}^i(x, y))$$

Trong đó,  $b_t^i(x, y)$  là giá trị nền tham chiếu thứ  $i$  tại thời điểm  $t$ ,  $b_{t-1}^i(x, y)$  là giá trị nền tham chiếu thứ  $i$  tại thời điểm  $(t-1)$ ,  $b_t^{i-1}(x, y)$  là giá trị nền tham chiếu thứ  $(i-1)$  tại thời điểm  $t$ , giá trị khởi tạo với  $i=0$ :  $b_t^0(x, y) = I_t(x, y)$

Mỗi nền  $b_t^i$  được đặc trưng bởi thời gian cập nhật  $\alpha_i$ .

Đối với mỗi frame, ta sẽ tính giá trị hiệu tuyệt đối  $\Delta_t^i(x, y)$  và giá trị biến theo thời gian  $v_t^i(x, y)$  như sau:

$$\Delta_t^i(x, y) = |I_t(x, y) - b_t^i(x, y)|$$

$$v_t^i(x, y) = v_{t-1}^i(x, y) + \text{sgn}(N \times \Delta_t^i(x, y) - v_{t-1}^i(x, y))$$

Giá trị mô hình nền thích ứng tổng hợp  $B_t(x, y)$  được tính như sau:

$$B_t(x, y) = \frac{\sum_{i \in [1, R]} \frac{\alpha_i (b_t^i(x, y))}{v_t^i(x, y)}}{\sum_{i \in [1, R]} \frac{\alpha_i}{v_t^i(x, y)}}$$

Với  $\alpha_i$  là giá trị được định nghĩa trước,  $i$  là chỉ số tham chiếu,  $R$  là tổng số chỉ số  $i$ . Giá trị thực nghiệm đặt  $R=3$ ,  $\alpha_1, \alpha_2, \alpha_3$  được đặt 1, 8, 16.

Trên mô hình nền thích ứng  $B_t(x, y)$  đã được tạo ra, ta áp dụng giải thuật  $\Sigma$ - $\Delta$  với mô hình nền  $B_t(x, y)$  này để xử lý xác định đối tượng chuyển động.

#### **Đánh giá:**

*Ưu điểm:* Bằng việc sử dụng mô hình nền thích ứng, giải thuật  $\Sigma$ - $\Delta$  cải tiến cho phép ta xác định được chính xác đối tượng trong trường hợp video thu được có nhiều đối tượng chuyển động.

*Nhược điểm:* Giải thuật này đòi hỏi độ phức tạp tính toán lớn.

#### **2.1.5 Giải thuật thống kê khác biệt cơ bản (Simple Statistical Difference):**

Giải thuật thống kê khác biệt cơ bản tính giá trị trung bình cho từng pixel riêng lẻ của khung video trước đó dựa trên việc sử dụng giá trị trung bình, độ lệch tiêu chuẩn cũng như sắp xếp mô hình nền. Mô hình nền thích ứng được tạo ra thông qua việc xác định giá trị từng pixel  $\mu_{xy}$  của mô hình nền. Giá trị  $\mu_{xy}$  được tính là giá trị trung bình của các pixel tương ứng từ một tập khung hình trước đó trong một khoảng

thời gian nhất định từ thời điểm khung video đầu tiên đến thời điểm khung video thứ K-1 (có thể coi khoảng thời điểm từ  $t_0 - t_{K-1}$ ).

$$\mu_{xy} = \frac{1}{K} \sum_{k=0}^{K-1} I_k(x, y)$$

K: là số lượng khung video đang xét

t: là chỉ số của khung video,  $t = 1..K$

$I_t(x, y)$ : là giá trị khung video đến hiện thời thứ t.

Với mỗi pixel, một giá trị ngưỡng biểu diễn bằng độ lệch chuẩn  $\sigma_{xy}$  trong cùng một khoảng thời gian ( $t_0 - t_{K-1}$ ), được tính bằng trung bình độ lệch giữa giá trị của pixel tương ứng trong các khung video trước và  $\mu_{xy}$ , công thức tính như sau:

$$\sigma_{xy} = \left( \frac{1}{K} \sum_{k=0}^{K-1} (I_k(x, y) - \mu_{xy})^2 \right)^{1/2}$$

Để phát hiện được chuyển động, giá trị tuyệt đối của hiệu giữa khung video đến và mô hình nền được tính toán. Từ đó ta sẽ xác định mặt nạ nhị phân phát hiện chuyển động  $D_t(x, y)$  được tính bởi công thức sau:

$$D_t(x, y) = \begin{cases} 1, & \text{if } |I_t(x, y) - \mu_{xy}| > \lambda \sigma_{xy} \\ 0, & \text{if } |I_t(x, y) - \mu_{xy}| \leq \lambda \sigma_{xy} \end{cases}$$

Ta chọn  $\lambda$  là tham số thực nghiệm.

Nếu  $D_t(x, y) = 0$  thì pixel này được xác định là pixel của nền, ngược lại

Nếu  $D_t(x, y) = 1$  thì pixel này là pixel của đối tượng chuyển động. Tập hợp những pixel này, ta sẽ xác định được đối tượng chuyển động.



### **Đánh giá:**

*Ưu điểm:* Giải thuật thống kê khác biệt cơ bản có ưu điểm là thuật toán đơn giản, tốc độ tính toán nhanh, phát hiện chính xác đối tượng chuyển động trong trường hợp video đầu vào đơn giản (có một đối tượng chuyển động).

*Nhược điểm:* Giải thuật này không phát hiện được hết các đối tượng trong trường hợp video đầu vào phức tạp: có nhiều đối tượng, có đối tượng không chuyển động trong một khoảng thời gian nhất định, hoặc các đối tượng chuyển động với tốc độ khác nhau.

#### **2.1.6 Giải thuật trừ nền trung bình với biến đổi cosine rời rạc (Running Average with Discrete Cosine Transform):**

Giải thuật trừ nền trung bình với biến đổi cosin rời rạc là sự cải tiến của giải thuật trừ nền trung bình (Running Average). Phương pháp này cho phép mô hình hóa nền thích ứng trong không gian miền biến đổi cosin rời rạc. Mô hình nền thích ứng được mô tả như sau:

$$d_t^{B,k} = (1 - \beta)d_{t-1}^{B,k} + \beta d_t^k$$

$\beta$  : Tham số thực nghiệm, tương tự như giải thuật trừ nền trung bình.

$L$  : Số lượng khối trong một frame.

$k$  : Chỉ số khối,  $k = 1, 2, \dots, L$

$d_t^k$  : Biểu thị vector hệ số biến đổi cosin rời rạc của khối pixel thứ  $k$ , tại thời điểm  $t$ , của khung video đến hiện tại.

$d_t^{B,k}$  : Biểu thị vector hệ số biến đổi cosin rời rạc nền của khối pixel thứ  $k$ , tại thời điểm  $t$ , trong miền biến đổi cosin rời rạc.

$d_{t-1}^{B,k}$  : Biểu thị vector hệ số biến đổi cosin rời rạc nền trước đó của khối pixel thứ  $k$ , tại thời điểm  $t-1$ , trong miền biến đổi cosin rời rạc

Để phát hiện đối tượng chuyển động, giải thuật này sử dụng kỹ thuật trừ nền. Giải thuật xác định sự khác biệt giữa frame đến và nền tương ứng trong một khối,

trong cùng miền biến đổi cosin rời rạc, ký hiệu  $\Delta_t^k$ ,  $\Delta_t^k$  được định nghĩa là khoảng cách Euclide giữa  $d_t^k$  và  $d_t^{B,k}$ :

$$\Delta_t^k = \sqrt{\|d_t^k - d_t^{B,k}\|}, k = 1, 2, \dots, L$$

Với  $\tau$  là một ngưỡng cho trước, thì nếu:

$\Delta_t^k \leq \tau$  thì khối thứ k này được xác định là nền

$\Delta_t^k > \tau$  thì khối thứ k được xác định là khối chuyển động (khối chứa những pixel của đối tượng chuyển động)

Như vậy, giải thuật trung bình liên tục với biến đổi cosin rời rạc đã xác định được ở mức khối của đối tượng chuyển động. Từ đây, có thể biến đổi để xác định đối tượng chuyển động ở mức pixel.

#### **Đánh giá:**

Giải thuật trung bình liên tục với biến đổi cosin rời rạc sử dụng hệ số biến đổi cosin rời rạc ở mức khối pixel để tạo mô hình nền thích ứng. Kết quả của giải thuật đưa ra các vùng chứa đối tượng chuyển động với độ chính xác nhất định thông qua quá trình xử lý hai giai đoạn. Giai đoạn đầu tiên, sử dụng kỹ thuật trừ nền mới trong không gian miền biến đổi cosin rời rạc để xác định toàn bộ hoặc một phần vùng chứa đối tượng chuyển. Giai đoạn thứ hai là xác định những pixel của đối tượng chuyển động trong những khối chứa chuyển động. Đặc biệt, giải thuật này sẽ giảm thời gian tính toán khi video đầu vào đã được nén sử dụng phương pháp biến đổi cosin rời rạc.

## 2.2. Giải thuật phát hiện chuyển động

Ở bài toán phát hiện chuyển động sử dụng phương pháp trừ nền và sử dụng giá trị Entropy đối với những khối pixel (một frame video được chia thành nhiều khối pixel). Xử lý ở mức khối pixel giá trị Entropy kết hợp với các phép toán hình thái học “giãn ảnh” và “co ảnh” để xác định tất cả các khối có chứa pixel của đối tượng chuyển động (gọi tắt là khối chuyển động). Xử lý ở mức pixel đối với các khối chuyển động để xác định những pixel của đối tượng chuyển động. Tập hợp những pixel này, ta có hình ảnh của đối tượng chuyển động. Trên cơ sở xác định các pixel của đối tượng chuyển động, ta có thể phát tín hiệu cảnh báo (âm thanh, hình ảnh) hoặc có thể lưu giữ video chứa đối tượng chuyển động. Theo các yêu cầu bài toán thực tế đặt ra, ta cũng có thể bổ sung thêm việc xác định những vùng đặc biệt trong phạm vi khung video của camera. Ví dụ như phân biệt đối tượng chuyển động khi đi qua khung cửa của phòng, đi qua những vạch ngăn cách trong khu vực camera giám sát.

### 2.2.1. Giới thiệu giải thuật

Giải thuật tiến hành xử lý dữ liệu đầu vào là chuỗi video số thu được từ camera theo dõi để đưa ra được kết luận có đối tượng đột nhập trong vùng camera theo dõi hay không. Qua nghiên cứu, tìm hiểu về bài toán phát hiện chuyển động và những giải thuật trước đó, học viên nhận thấy, các giải thuật trước đây như đã trình bày ở trên đã cơ bản xác định chính xác các đối tượng chuyển động với độ phức tạp giải thuật chấp nhận được. Tuy nhiên, trong nhiều trường hợp, đối với ảnh phức tạp chứa nhiều đối tượng chuyển động, các đối tượng chuyển động với quy luật và vận tốc khác nhau... các giải thuật đã nêu trên đều gặp phải vấn đề không xác định được chính xác toàn bộ các đối tượng chuyển động, thời gian xử lý còn chậm. Học viên tìm hiểu, nghiên cứu giải thuật mới nhằm tăng tính chính xác kết quả phát hiện đối tượng chuyển động trong những ảnh phức tạp, đồng thời đưa ra một số giá trị ngưỡng của giải thuật để phù hợp với điều kiện thực tế đặt ra, bổ sung tính năng xác định những vùng giám sát trong vùng camera theo dõi.

Mô tả về giải thuật: Giải thuật này thực hiện lựa chọn các nền thích ứng phù hợp, trên cơ sở đó xây dựng các module chức năng nhằm phát hiện pixel của đối tượng

chuyển động trong các vùng đã được phân định của ảnh. Giải thuật bao gồm ba thành phần chính như sau:

(1) - Xây dựng nền tối ưu: Vấn đề quan trọng nhất đối với bài toán phát hiện chuyển động sử dụng phương pháp trừ nền là xây dựng được nền tối ưu. Nền tối ưu là những frame chứa những pixel của những đối tượng hầu như không thay đổi (đối tượng tĩnh). Vấn đề tối ưu nền bao gồm cả việc cập nhật nền mới sau mỗi lần thực hiện phép toán trừ nền. Việc lựa chọn nền tối ưu trong giải thuật đề cập được thực hiện thông qua việc sử dụng thủ tục so sánh hai nền.

(2) - Xác định những khối (vùng) có khả năng chứa đối tượng chuyển động: Thuật toán tiến hành xử lý những frame khác nhau theo thứ tự thời gian của chuỗi video thu được từ camera. Tiến hành xử lý ở mức pixel trên mỗi frame để xác định đối tượng chuyển động. Để việc tính toán chính xác và nhanh hơn, ta chia frame thành những khối pixel và tiến hành xác định chuyển động trên từng khối này. Tính giá trị Entropy của từng khối (Entropy là đại lượng xác định độ sai khác giá trị các pixel trong khối đang xét, Entropy càng lớn chứng tỏ các pixel của khối có sự sai khác nhiều và chứng tỏ có khả năng khối đó chứa các pixel của đối tượng chuyển động). Kết hợp với các phép toán hình thái học “giãn ảnh” và “co ảnh” để đưa ra những khối có khả năng chứa đối tượng chuyển động.

(3) - Xác định đối tượng chuyển động: Trên cơ sở xác định các khối có khả năng chứa đối tượng chuyển động, ta sẽ sử dụng những ngưỡng phù hợp để phân biệt đối tượng chuyển động và nền. Khi đó ta sẽ trích xuất được đối tượng chuyển động.

So sánh với các giải thuật đã được công bố trước đó, giải thuật này xử lý nền nhanh hơn và chính xác hơn, có thể phát hiện nhanh và chính xác hầu như tất cả các đối tượng chuyển động, thích ứng, linh hoạt trong các điều kiện video đầu vào khác nhau.

### 2.2.2. Nội dung giải thuật

**Giải thuật được xây dựng dựa trên 3 module chính sau:**

- Module Mô hình nền - Background Modeling Module
- Module Cảnh báo chuyển động - Alarm Trigger Module

- Module Trích xuất đối tượng - Object Extraction Module

\* Module Mô hình nền xây dựng mô hình nền tối ưu thông qua việc sử dụng thuật toán so sánh nhanh và chính xác hai nền để tạo ra pixel nền tối ưu cho mô hình nền, những pixel tối ưu này là những pixel có sự biến đổi về giá trị là ít. Việc tạo ra mô hình nền tối ưu là cơ sở cho việc xử lý phát hiện chuyển động trong các bước tiếp theo được nhanh chóng và chính xác.

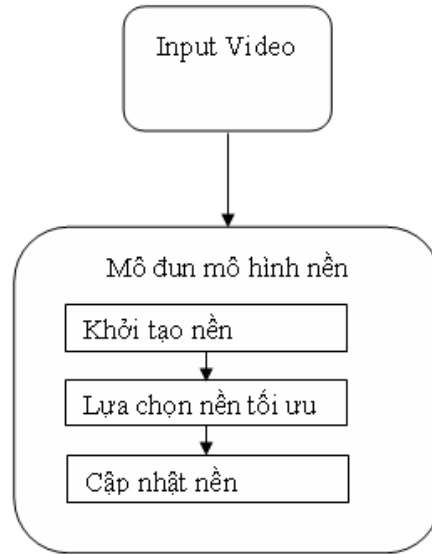
\* Module Cảnh báo chuyển động được xây dựng để xác định chính xác những khối ảnh chứa đối tượng chuyển động. Ảnh được chia thành các khối, trong đó khối chứa những pixel của đối tượng chuyển động được gọi là khối chuyển động (motion block). Yêu cầu đặt ra của bài toán là xác định chính xác và đầy đủ tất cả các khối chuyển động. Module Cảnh báo chuyển động dựa trên phương pháp đánh giá giá trị Entropy của khối và các phép toán hình thái học “co ảnh” và “giãn ảnh” (morphological erosion and dilation operations) để xác định các khối chuyển động. Module Cảnh báo chuyển động loại bỏ việc kiểm tra từng pixel của đối tượng chuyển động trên toàn bộ ảnh nền, mà chỉ tập trung vào những khối chuyển động. Vì vậy, Module Cảnh báo chuyển động đã giảm đáng kể độ phức tạp tính toán trong quá trình xử lý phát hiện chuyển động, giúp cho Module Trích xuất đối tượng chỉ xử lý, trích xuất đối tượng chuyển động từ những khối chuyển động.

\* Module Trích xuất đối tượng sẽ kiểm tra tất cả các khối chuyển động đã được phát hiện trong Module Cảnh báo chuyển động. Việc kiểm tra phát hiện chuyển động sử dụng giải thuật lựa chọn ngưỡng hiệu quả. Kết quả của Module này là xây dựng được mặt nạ nhị phân phát hiện chuyển động (Binary motion detection mask), từ đó trích xuất được đối tượng chuyển động.

### 2.2.2.1. Module Mô hình nền:

Module Mô hình nền được chia thành các phần chính sau:

- Khởi tạo mô hình nền
- Lựa chọn nền tối ưu
- Cập nhật mô hình nền



Hình 2.1. Lược đồ Mô đun mô hình nền

#### 2.2.2.1.1. Khởi tạo mô hình

Như đã trình bày ở Phần 2.1.1, video đầu vào được chia thành những frame theo thứ tự thời gian. Video thông thường có những chuẩn như: 24 fps, 25 fps, 30 fps, 48 fps... Thông thường, ta xử lý ảnh có kích thước 640 x 480 và 24 fps.



Hình 2.2. Mô tả những frame của video đầu vào

Thủ tục khởi tạo mô hình nền thực hiện theo ý tưởng của giải thuật The Modified Moving Average – MMA sẽ lấy giá trị trung bình của các khung từ 1 – K để khởi tạo mô hình nền đầu tiên (với K là số nguyên).

Với mỗi pixel (x,y), công thức tổng quát tính giá trị của mô hình nền hiện thời  $B_t(x,y)$  tại thời điểm t như sau:

$$B_t(x, y) = B_{t-1}(x, y) + \frac{1}{t}(I_t(x, y) - B_{t-1}(x, y))$$

Tương đương :

$$B_t(x, y) = B_{t-1}(x, y)\left(1 - \frac{1}{t}\right) + \frac{1}{t}I_t(x, y)$$

.  $B_t(x, y)$ : Giá trị mô hình nền ở thời điểm hiện tại  $t$ .

.  $B_{t-1}(x, y)$ : Giá trị mô hình nền trước đó ở thời điểm  $(t-1)$  - previous background model.

.  $I_t(x, y)$ : Giá trị của frame video đến hiện thời - current incoming video frame.

.  $t$ : Chỉ số frame của chuỗi video,  $t < K$ .

.  $K$ : Số frame đầu tiên của chuỗi video đầu vào để khởi tạo nền ban đầu. Trong phần thực nghiệm, học viên lựa chọn  $K=50$ . Chương trình sẽ tạo nền đầu tiên sau 50 frame.

```
void bgInit(uchar *d_background, uchar *d_frame, int k, long length)
{
    for(long i=0; i< length; i++, d_background++, d_frame++)
        (*d_background) = (*d_background) * (1- float(1./k)) +
                           float(1./k)*(*d_frame);
}
```

Trong phần khởi tạo mô hình nền, ta chỉ quan tâm đến hai kết quả cuối cùng đó là giá trị nền cuối cùng  $B_t(x, y)$  ( $d\_background$ ) và frame video đến hiện thời  $I_t(x, y)$  ( $d\_frame$ ) mà không cần phải lưu trữ toàn bộ những frame của chuỗi video trong suốt quá trình tính toán. Do đó giảm tải được các biến nhớ và dung lượng bộ nhớ, giúp cho việc xử lý, tính toán nhanh hơn.

#### 2.2.2.1.2. Lựa chọn nền tối ưu

Lựa chọn nền tối ưu gồm 3 thành phần chính sau:

(1) - Xác định những pixel thuộc ứng cử viên nền (background candidates) là những pixel có giá trị hầu như không thay đổi giữa 2 frame liên tiếp. Thủ tục này sử dụng thuật toán so sánh nhanh giữa những pixel của hai frame liên tiếp.

(2) - Xác định ứng cử viên nền bằng việc sử dụng module huấn luyện tín hiệu ổn định đối với từng pixel thuộc ứng cử viên nền đã được xác định ở bước (1).

(3) Xác định những pixel nền tối ưu theo thủ tục so sánh chính xác. Đây là những pixel có giá trị không thay đổi giữa những frame trước đó. Tập hợp những pixel này tạo nên nền tối ưu.

Mô tả các thuật toán, thủ tục:

\* *Thuật toán so sánh nhanh (Rapid Matching)*: Là thuật toán được sử dụng để tìm kiếm nhanh số lượng lớn các pixel của nền ứng cử viên. Thuật toán này tiến hành tìm kiếm và so sánh giá trị những pixel tương ứng của 2 frame liên tiếp nhau là frame video tại thời điểm xét  $t$ :  $I_t(x,y)$  và frame video tại thời điểm trước đó  $t-1$ :  $I_{t-1}(x,y)$ . Nếu giá trị của 2 pixel tương ứng này bằng nhau thì ta sẽ tập hợp những pixel này thành pixel của ứng cử viên nền:  $M_t(x,y)$  và đưa vào xử lý ở module huấn luyện tín hiệu ổn định.

```
void rapidMatching(uchar *d_rapid,uchar *d_prev,uchar
*d_frame,long length,int equal)
{
    for(long i=0;i<length;i++,d_rapid++,d_prev++,d_frame++)
        if(abs((*d_prev) - (*d_frame)) < equal)
            (*d_rapid) = (*d_frame);
        else
            (*d_rapid) = 0;
}
```

\* Module huấn luyện tín hiệu ổn định (Stable Signal Trainer): Tất cả các pixel thuộc tập các ứng viên nền đã được lựa chọn thông qua thủ tục so sánh nhanh ở phần (1) sẽ được xử lý thông qua module huấn luyện tín hiệu ổn định. Kết quả của module huấn luyện tín hiệu ổn định nhằm làm nổi bật những pixel ổn định này (đây có thể là những pixel của nền). Việc “huấn luyện” được mô tả bằng công thức tổng quát sau:

$$M_t(x,y) = \begin{cases} M_t(x,y) + p, & \text{if } I_t(x,y) > M_{t-1}(x,y) \\ M_t(x,y) - p, & \text{if } I_t(x,y) < M_{t-1}(x,y) \end{cases}$$

.  $M_t(x,y)$ : Là pixel ứng cử viên nền hiện tại.



.  $M_{t-1}(x,y)$ : Là pixel tương ứng của ứng cử viên nền trước đó.

.  $I_t(x,y)$ : Là frame video đến

.  $p$ : Giá trị huấn luyện, thực nghiệm đặt  $p = 1$ .

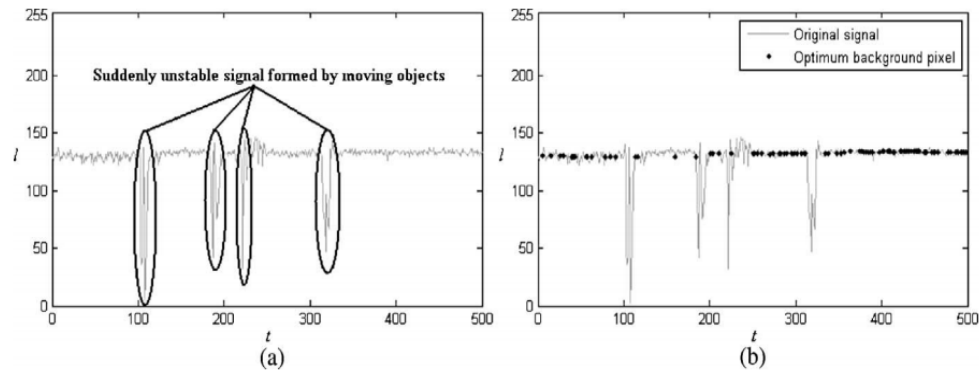
. Ứng cử viên nền khởi tạo:  $M_0(x,y) = I_0(x,y)$ .

```
void stableSigTra(uchar *d_stab,uchar *d_rapid,uchar *d_frame,long
length,int stable)
{
    for(long i=0;i<length;i++,d_stab++,d_rapid++,d_frame++)
        if((*d_rapid) == (*d_frame))
        {
            if((*d_stab) < (*d_frame))
                (*d_stab) += stable;
            else
                (*d_stab) -= stable;
        }
}
```

\* Thủ tục so sánh chính xác (Accurate matching): Module huấn luyện tín hiệu ổn định nhằm làm nổi bật những pixel ổn định của tập ứng cử viên nền. Các pixel của ứng cử viên nền này sẽ được so sánh tương ứng với những pixel của frame video hiện thời  $I_t(x,y)$  (Thủ tục so sánh chính xác). Trường hợp giá trị của các pixel tương ứng khác nhau thì những pixel này được gán giá trị 255 (màu trắng), ngược lại, những pixel là pixel của ứng cử viên nền  $M_t(x,y)$  được gán giá trị 0 (màu đen) – Hình 2.4

```
void accurateMatching(uchar *d_accr,uchar *d_stab,uchar
*d_frame,long length)
{
    for(long i=0;i<length;i++,d_stab++,d_accr++,d_frame++)
        if(abs((*d_stab) - (*d_frame)) >
ACCURATE_MATCHING_EQUAL)
            (*d_accr) = 255; //Nhưng điểm trạng thể hiện vị
tri nhưng pixel của đối tượng chuyển động
        else
```

```
(*d_accr)=0; //Nhưng điểm đen là vị trí pixel
    ung viên nền
}
```

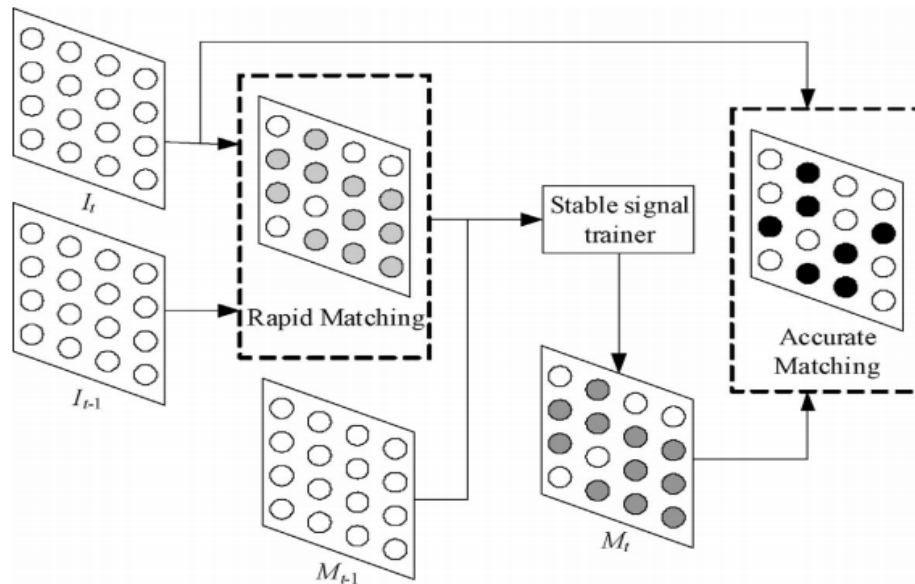


Hình 2.3. Mô tả phân bố mức xám

(a) của tín hiệu gốc

(b) của các pixel trích xuất từ module nền tối ưu.

Hình 2.3. Minh họa phân bố mức xám, bao gồm tín hiệu ổn định sẽ là các thành phần chính của nền. Những tín hiệu không ổn định chỉ thỉnh thoảng xuất hiện và điều này cho thấy sự xuất hiện của đối tượng chuyển động.



Hình 2.4. Mô tả lựa chọn nền tối ưu

Hình 2.4. Mỗi pixel màu xám của ứng viên nền  $M_t(x,y)$  được chuyển đổi trở thành pixel màu xám đậm thông qua module huấn luyện tín hiệu ổn định trở thành nền ứng viên  $M_t(x,y)$ . Thủ tục so sánh chính xác được sử dụng để so sánh giá trị từng pixel màu xám đậm của  $M_t(x,y)$  với những pixel tương ứng của  $I_t(x,y)$ . Trường hợp, giá trị hai pixel tương ứng bằng nhau, những pixel màu xám đậm sẽ được chuyển thành màu đen, như hình vẽ. Đây chính là những pixel có mức ổn định cao và sẽ hình thành nên thành phần chính của nền.

### 2.2.2.1.3. Cập nhật mô hình nền

Việc xác định nền tối ưu có ý nghĩa quan trọng trong giải thuật phát hiện chuyển động sử dụng phương pháp trừ nền. Trong phần trên, ta đã xác định được những pixel của nền tối ưu  $M_t(x,y)$ . Và để đảm bảo việc phát hiện chính xác và nhanh các đối tượng chuyển động, mô hình nền phải được cập nhật liên tục để tạo ra nền tối ưu nhất cho việc so sánh xác định đối tượng chuyển động. Thủ tục sẽ tiến hành cập nhật nền tại những vị trí của những pixel nền tối ưu  $M_t(x,y)$  (tại những điểm đen) tương ứng với vị trí của pixel trong mô hình nền  $B_t(x,y)$  thông qua việc áp dụng công thức dịch chuyển trung bình để làm “mềm” mô hình nền. Công thức dịch chuyển trung bình như sau:

$$B_t(x,y) = B_{t-1}(x,y) + \frac{1}{\alpha} (I_t(x,y) - B_{t-1}(x,y))$$

. Với  $\alpha$  là tham số định trước, trong thực nghiệm sử dụng  $\alpha=8$ .

.  $B_t(x,y)$  là nền sau  $K=50$  frame đầu tiên

.  $I_t(x,y)$  là frame đến của video đầu vào.

```
void bgUpdating(uchar *d_background, uchar *d_accr, uchar
                *d_frame, long length, int bgupdate)
{
    for(long i=0; i<length; i++, d_background++, d_accr++, d_frame++)
        if((*d_accr)==0) //Cap nhat tai nhung vi tri pixel ung
vien nen - nhung diem ma gia tri hau nhu khong thay doi
            (*d_background) = (*d_background) +
                (1./bgupdate )*((*d_frame) -
                (*d_background));
```

```
}
```

Quá trình cập nhật nền sẽ được tiến hành đối với tất cả những frame của video đầu vào. Học viên nhận thấy, trong quá trình xử lý đối với những video đầu vào được quay trong những khung cảnh, điều kiện, thời gian khác nhau, không nhất thiết phải cập nhật nền liên tục đối với từng frame của video đầu vào và chỉ cần lựa chọn 3 frame cập nhật một lần, thậm chí trong những điều kiện ánh sáng tốt thì có thể 5 frame cập nhật một lần, chương trình vẫn đảm bảo khả năng phát hiện được tất cả các chuyển động. Việc cập nhật như vậy sẽ cải tiến tốc độ xử lý cập nhật nền của thuật toán.

```
// buoc khoi tao viec hoc nen trong 50 frame dau tien
if(k< 50)
{
    bgInit1(d_background,d_resize,(long)ws*h,k);
    //continue;
}
else
{
    // neu k<200 hoc nen full, neu k>200 thi 3 frame moi hoc nen 1
    lan
    if(k<200 || k >200 && k%3==0)
    {
        // rapid matching
        rapidMatching1(d_rapid,d_prev,d_resize,(long)ws*h,equal);

        // stable signal trainer

        stableSigTra1(d_stableSign,d_rapid,d_resize,(long)ws*h,stable);

        // accurate matching
        accurateMatching1(d_accr,d_stableSign,d_resize,(long)ws*h);

        // updata background

        bgUpdating1(d_background,d_accr,d_resize,(long)ws*h,bgupdate);
    }
```

### 2.2.3. Module cảnh báo chuyển động:

Module này được xây dựng để xác định các đối tượng chuyển động. Để làm được điều này, trước tiên, frame video sẽ được phân chia thành những khối pixel và các thủ tục tiếp theo sẽ tiến hành xử lý trên cơ sở những khối này. Ta đưa ra định nghĩa những khối chuyển động là những khối mà có khả năng chứa đối tượng chuyển động. Module này sẽ xác định tất cả những khối chuyển động để từ đó tìm ra được đối tượng chuyển động. Module được chia thành hai phần chính:

(1) - Sử dụng giá trị Entropy của khối để xác định xem khối này có khả năng chứa đối tượng chuyển động hay không.

(2) - Sử dụng phép toán hình thái học “co ảnh” và “giãn ảnh” để xác định những khối nền và những khối chỉ chứa đối tượng chuyển động.

#### 2.2.3.1. Xác định giá trị Entropy của khối

Entropy là một đại lượng toán học dùng để đo lường tin không chắc chắn (hay lượng ngẫu nhiên) của một sự kiện hay của phân phối ngẫu nhiên cho trước. Một số tài liệu tiếng anh gọi là Uncertainty Measure [1],[4] (Theo định nghĩa của Giáo trình Lý thuyết thông tin, tác giả: TS Lê Quyết Thắng, Ths Phan Tấn Tài, Ks Dương Văn Hiếu – Đại học Cần Thơ 2010).

Entropy của  $x$  là giá trị mong đợi của các độ ngạc nhiên mà  $x$  có thể nhận.

Trong trường hợp này, biến  $x$  là độ lệch về màu sắc giữa 2 frame ảnh đến và ảnh hiện thời. Nếu độ lệch màu sắc không có biến đổi gì hoặc giá trị độ lệch chỉ nằm trong 1 khoảng duy nhất (Entropy nhỏ hơn một ngưỡng cho trước) thì ta coi như không có sự thay đổi về màu sắc của  $x$ . Ngược lại, nếu Entropy lớn hơn ngưỡng trên thì có thể kết luận đã có sự sai khác về màu sắc của  $x$ .

Ví dụ: "Một dòng chữ luôn chỉ có các ký tự "a" sẽ có entropy bằng 0, vì ký tự tiếp theo sẽ luôn là "a". Một dòng chữ chỉ có hai ký tự 0 và 1 ngẫu nhiên hoàn toàn sẽ có entropy là 1 bit cho mỗi ký tự".

Khối chuyển động là khối có độ lệch màu sắc giữa 2 ảnh đến và ảnh hiện thời biến đổi ngẫu nhiên (sẽ là các giá trị khác nhau) nên  $\text{Entropy} > 1$  (Lấy ngưỡng là 1).

Ngược lại, nếu  $\text{Entropy} \leq 1$  tức là các giá trị màu biến đổi là như nhau, có thể là không đổi hoặc cùng thay đổi trong một khoảng nhỏ h nào đó.

Như vậy, trong bài toán phát hiện chuyển động, ta có thể sử dụng giá trị Entropy để xác định xem những pixel có phải của đối tượng chuyển động không. Khi giá trị Entropy của điểm ảnh có sự thay đổi ở một ngưỡng nào đó, thì ta có thể kết luận đây là pixel của đối tượng chuyển động, ngược lại, giá trị entropy này không thay đổi hoặc thay đổi trong một khoảng nhỏ nào đó (nhỏ hơn giá trị của ngưỡng) ta xác định đây không phải là pixel của đối tượng chuyển động.

### Tính giá trị Entropy của một phân phối

Xét biến ngẫu nhiên  $X$  có phân phối:

$X$	$x_1$	$x_2$	$x_3$	$\dots$	$x_M$
$P$	$p_1$	$p_2$	$p_3$	$\dots$	$p_M$

Nếu gọi  $A_i$  là sự kiện  $X = x_i$ , ( $i=1, 2, 3, \dots$ ) thì Entropy của  $A_i$  là:

$$h(A_i) = h(p_i), \quad p_i \text{ là phân phối xác suất của } X$$

Gọi  $Y = h(X)$  là hàm ngẫu nhiên của  $X$  và nhận các giá trị là dãy các Entropy của các sự kiện  $X=x_i$ , tức là  $Y = h(X) = \{h(p_1), h(p_2), \dots, h(p_n)\}$ .

Vậy, Entropy của  $X$  chính là kỳ vọng toán học của  $Y = h(X)$  có dạng:

$$H(X) = H(p_1, p_2, p_3, \dots, p_n) = p_1 h(p_1) + p_2 h(p_2) + \dots + p_n h(p_n)$$

Tổng quát Entropy của  $X$  là:

$$H(X) = \sum_{i=1}^n p_i h(p_i)$$

Định lý dạng giải tích của Entropy:

$$H(X) = H(p_1, p_2, p_3, \dots, p_n) = C \sum_{i=1}^n p_i \log(p_i)$$

$$C = \text{const} > 0$$

Cơ số logarithm là bất kỳ.

Bổ đề:  $h(p) = -C \log(p)$ .

Trường hợp  $C = 1$  và cơ số logarithm = 2 thì đơn vị tính là bit.

Khi đó:  $h(p) = -\log_2(p)$  (đvt: bit) và Entropy của  $X$  là:

$$H(X) = H(p_1, p_2, p_3, \dots, p_n) = -\sum_{i=1}^n p_i \log_2(p_i)$$

Như vậy, ta sẽ sử dụng giá trị Entropy của độ lệch (hiệu tuyệt đối) giữa frame ảnh hiện thời và nền, hay nói khác đi đó chính là độ lệch về màu sắc của hai frame video. Việc xác định giá trị Entropy của độ lệch này là cơ sở để xác định những pixel của đối tượng chuyển động.

Sau khi xây dựng được nền tối ưu  $B_t(x,y)$  tương ứng với mỗi frame hiện thời  $I_t(x,y)$  như đã trình bày ở Phần 2.2.1, tiếp theo ta sẽ tính hiệu tuyệt đối  $\Delta_t(x,y)$  giữa mô hình nền tối ưu và frame hiện thời, công thức tính như sau:

$$\Delta_t(x, y) = |B_t(x, y) - I_t(x, y)|$$

Giả sử, với mỗi khối  $\omega \times \omega(i, j)$  với hiệu tuyệt đối  $\Delta_t(x,y)$  được tạo thành bởi vector xám rời rạc cấp  $V \{L_0, L_1, \dots, L_{V-1}\}$ . Hàm mật độ xác suất của những pixel có mức xám  $h$  của khối  $\omega \times \omega(i, j)$ :  $P_h^{(i,j)}$  được định nghĩa như sau:

$$P_h^{(i,j)} = n_h^{(i,j)} / \omega^2$$

Trong đó:

-  $h$  là phân tử tùy ý của  $\{L_0, L_1, \dots, L_{V-1}\}$  đại diện cho bất kỳ vector mức xám trong mỗi khối  $\omega \times \omega(i, j)$ .

-  $n_h^{(i,j)}$  là số pixel tương ứng với mức xám  $h$  tùy ý

- Coi  $h \rightarrow 0$  khi  $h < \tau$ ,  $\tau$  là giá trị ngưỡng chuyển động đối với cấp xám rời rạc. Trong thuật toán ta sử dụng giá trị  $\omega = 8$  và  $\tau = 20$

**Minh hoạ:**

Giả sử  $\omega = 4$ , và khối  $\omega \times \omega(i, j)$  có phân bố mức xám như sau :

2	3	5	6
4	4	4	5
4	4	4	4
25	2	1	2

Mức xám của khối này:  $V = 7$ , tương ứng với các vector xám rời rạc:  $\{1, 2, 3, 4, 5, 6, 25\}$

Hàm mật độ xác suất của những điểm có giá trị  $h = 4$  của khối này là:

$$P_4 = 7 / (4 \times 4)$$

( Với  $h = 4$ , ta đếm được có 7 điểm có giá trị bằng 4, giá trị  $\omega = 4$  )

Như vậy, giá trị Entropy của khối theo định nghĩa sẽ được tính bởi công thức:

$$E_{(i,j)} = - \sum_{h=0}^{L_h-1} P_h^{(i,j)} \log_2(P_h^{(i,j)})$$

Sau khi giá trị Entropy  $E(i,j)$  của mỗi khối  $\omega \times \omega$  được tính, khối chuyển động  $A(i, j)$  được xác định như sau:

$$A(i, j) = \begin{cases} 1, & \text{if } E(i,j) > T \\ 0, & \text{otherwise} \end{cases}$$

Khi tính giá trị Entropy của khối  $A(i, j)$  lớn hơn  $T$  ( $T$  là ngưỡng cho các giá trị Entropy), Khối  $A(i, j)$  được gán bằng '1', có nghĩa  $A(i, j)$  chứa những pixel của đối tượng chuyển động. Ngược lại,  $A(i, j)$  gán bằng '0', có nghĩa khối đó không chứa pixel của đối tượng chuyển động.





Hình 2.5. Minh họa các giá trị Entropy của frame với ngưỡng  $T = 1$

Hình 2.5. Minh họa khối dựa trên giá trị Entropy, mật độ thay đổi rõ rệt của các khối trong frame. Với ngưỡng  $T = 1$ , các khối có khả năng chứa đối tượng chuyển động được phát hiện (phần hình màu đỏ).

### 2.2.3.2. Sử dụng các phép toán hình thái học:

Sử dụng các phép toán hình thái học “co ảnh” (Erosion) và “giãn ảnh” (Dilation) sẽ giúp ta xác định được những khối nền (khối không chứa pixel của đối tượng chuyển động) và khối chứa chuyển động hoàn toàn.[2] Trong Phần 2.2, ta đã xác định được những khối có khả năng chứa đối tượng chuyển động. Tuy nhiên, để kết luận chính xác những khối nào chứa chuyển động ta cần sử dụng hai phép toán hình thái học “co ảnh” và “giãn ảnh”. Phép toán “co ảnh” loại bỏ một số thành phần phụ của ảnh. Trong trường hợp đối với mỗi khối chuyển động, nếu ta loại bỏ một số thành phần phụ của khối mà khối đó vẫn là khối chuyển động thì ta khẳng định rằng khối này chứa các pixel của đối tượng chuyển động. Phép toán hình thái học “giãn ảnh” nhằm khôi phục lại kích thước của khối để đảm bảo kích thước của đối tượng chuyển động sẽ không bị thay đổi.

### Các phép toán hình thái học:

- Phép “co ảnh” theo cấu trúc nguyên tố B với đối tượng chuyển động A ta sẽ được ảnh A’ của đối tượng chuyển động:

$$A' = A \ominus B$$

- Phép “giãn ảnh” theo cấu trúc nguyên tố B với ảnh A’ ta thu được ảnh A’’ chính là ảnh A của đối tượng chuyển động (khôi phục lại ảnh ban đầu):

$$A'' = A' \oplus B = (A \ominus B) \oplus B$$

Thuật toán sử dụng hàm của thư viện Opencv để thực hiện các phép toán hình thái học “co ảnh” và “giãn ảnh”:

Void **cvErode**(const CvArr\* *src*, CvArr\* *dst*, IplConvKernel\* *element=NULL*,  
int *iterations=2*)

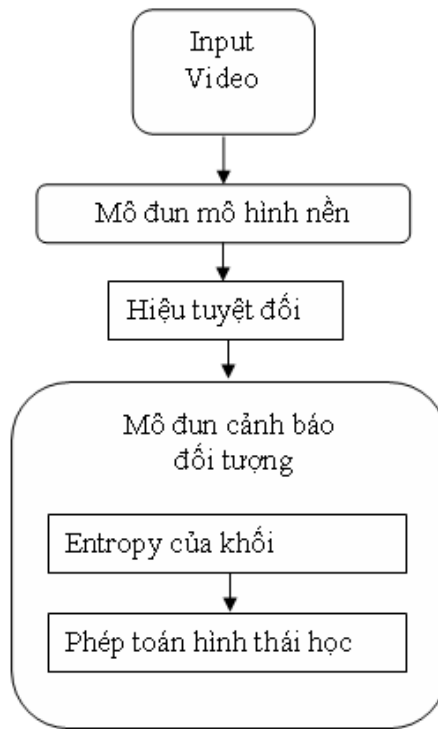
Void **cvDilate**(const CvArr\* *src*, CvArr\* *dst*, IplConvKernel\* *element=NULL*,  
int *iterations=2*)

. *src*: ảnh nguồn

. *dst*: ảnh đích

. *element*: cấu trúc nguyên tố, Opencv ngầm định (*element = NULL*) là cấu trúc nguyên tố  $3 \times 3$

. *iterations*: số lần áp dụng phép toán

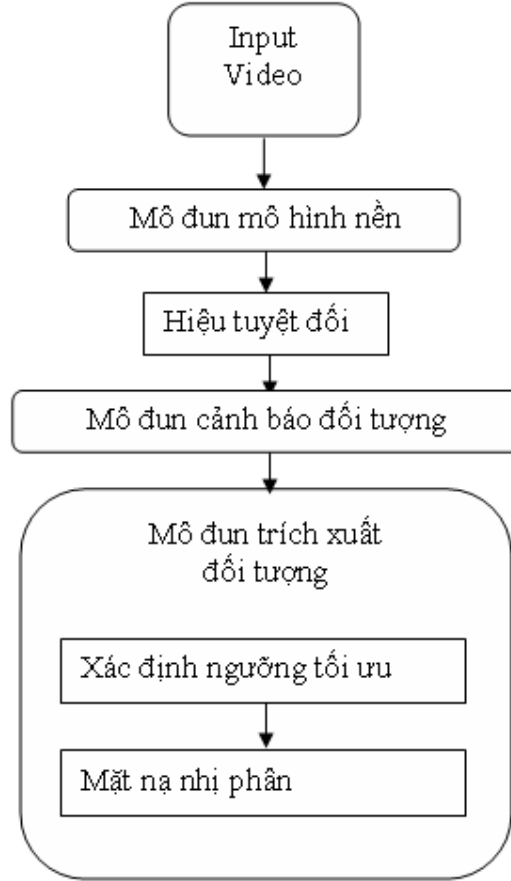


Hình 2.6. Lược đồ Module cảnh báo đối tượng

#### 2.2.4. Module Trích xuất đối tượng:

Kết quả của Module cảnh báo đối tượng chuyển động cho phép xác định được những khối pixel chứa đối tượng chuyển động thông qua giá trị Entropy và các phép toán hình thái học. Tuy nhiên, để loại bỏ các thành phần nhiễu và xác định chính xác đối tượng chuyển động, thì cần phải sử dụng một giá trị ngưỡng phù hợp. Giá trị ngưỡng này sẽ cho phép xác định chính xác những pixel của đối tượng chuyển động trên frame (thể hiện bằng biến mặt nạ nhị phân phát hiện chuyển động).

Để đưa ra giá trị ngưỡng tối ưu, thuật toán sử dụng hai biến: biến ngắn  $v_t^s(x, y)$  và biến dài  $v_t^l(x, y)$ :



Hình 2.7. Lược đồ Module trích xuất đối tượng

**- Tính giá trị biến ngắn  $v_t^s(x, y)$ :**

Áp dụng thủ tục huấn luyện tín hiệu ổn định để tính giá trị biến ngắn đối với mỗi frame với điều kiện giá trị hiệu tuyệt đối  $\Delta_t(x, y) \neq 0$ . Công thức cho việc đánh giá biến ngắn như sau:

$$v_t^s(x, y) = \begin{cases} v_{t-1}^s(x, y) + p, & \text{if } N \times \Delta_t(x, y) > v_{t-1}^s(x, y) \\ v_{t-1}^s(x, y) - p, & \text{if } N \times \Delta_t(x, y) < v_{t-1}^s(x, y) \end{cases}$$

.  $v_t^s(x, y)$ : giá trị biến ngắn hiện thời

.  $v_{t-1}^s(x, y)$ : giá trị biến ngắn trước đó

.  $N$ : tham số định nghĩa trước trong khoảng 1 – 4, thực nghiệm đặt  $N = 2$

. Giá trị khởi tạo biến ngắn:  $v_0^s(x, y) = \Delta_0(x, y)$ .

- **Tính giá trị biên dài  $v_t^l(x, y)$ :**

Tương tự như vậy, ta sử dụng thủ tục huấn luyện tín hiệu ổn định đã được điều chỉnh để tính giá trị biên dài  $v_t^l(x, y)$  cho mỗi frame. Công thức tính biên dài như sau:

$$v_t^l(x, y) = \begin{cases} v_{t-1}^l(x, y) + p, & \text{if } v_t^s(x, y) > v_{t-1}^l(x, y) \\ v_{t-1}^l(x, y) - p, & \text{if } v_t^s(x, y) < v_{t-1}^l(x, y) \end{cases}$$

.  $t$ : là bội số của  $\alpha$

.  $v_t^l(x, y)$ : giá trị biên dài hiện tại

.  $v_{t-1}^l(x, y)$ : giá trị biên dài trước đó

. Giá trị khởi tạo biên dài:  $v_0^l(x, y) = \Delta_0(x, y)$

Sau khi giá trị biên ngắn và biên dài được tính. Giá trị biên tốt nhất  $V_t(x, y)$  được tính cho mỗi frame như sau:

$$V_t(x, y) = N \times \min(v_t^s(x, y), v_t^l(x, y))$$

Cuối cùng, ta sẽ xác định những pixel của đối tượng chuyển động bằng biến mặt nạ nhị phân phát hiện đối tượng chuyển động  $D_t(x, y)$  từ các khối chuyển động của mỗi frame theo công thức tính như sau:

$$D(x, y) = \begin{cases} 1, & \text{if } \Delta_t(x, y) > V_t(x, y) \\ 0, & \text{otherwise} \end{cases}$$

$D(x, y) = 1$  chứng tỏ, pixel đó là pixel của đối tượng chuyển động, ngược lại,  $D(x, y) = 0$ , là pixel của nền. Như vậy ta đã trích xuất được đối tượng chuyển động thông qua tập các pixel có  $D(x, y) = 1$ .

```

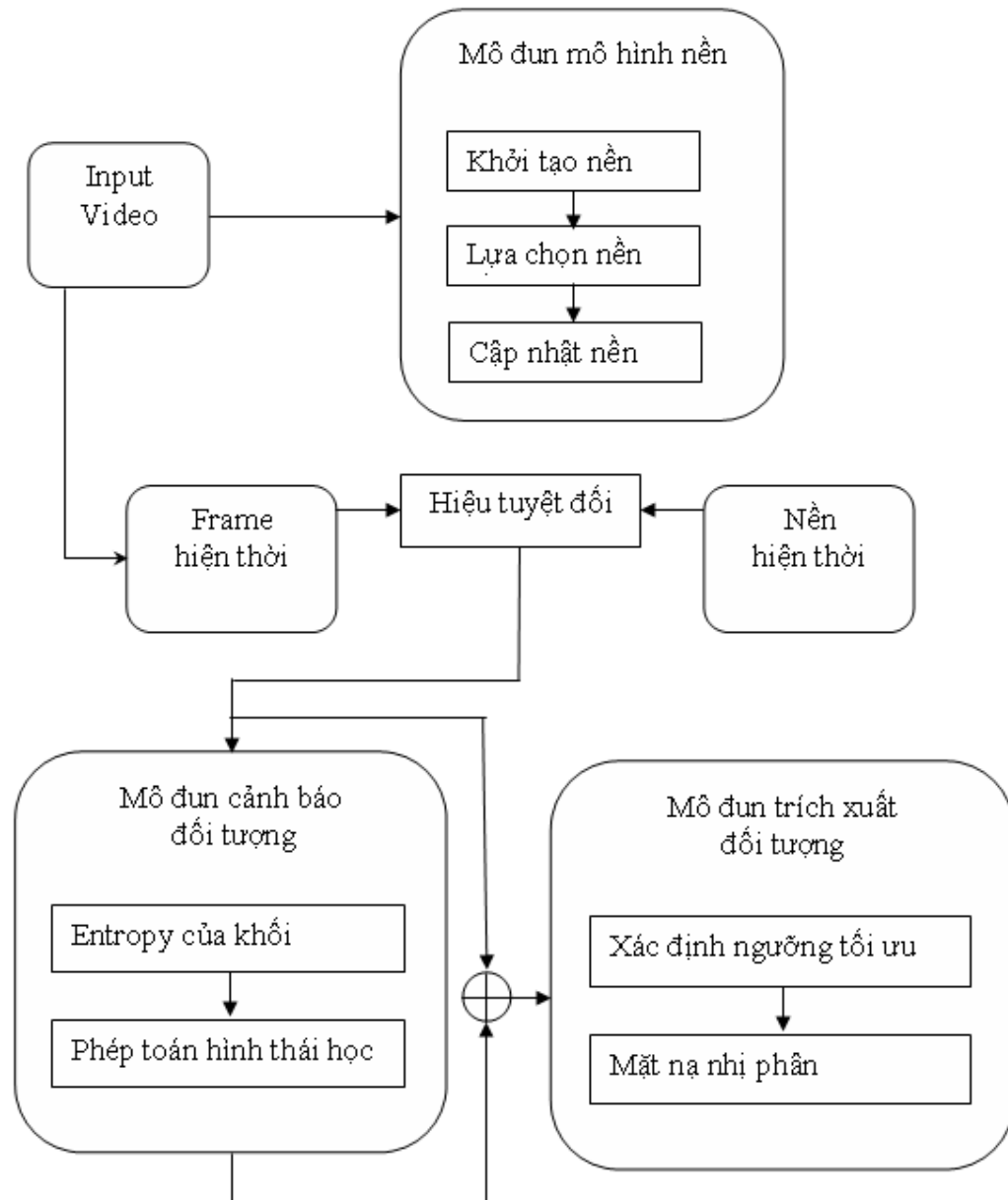
if(TERM_N * subdata[ i* ws + j] == vsmask[i*ws +j]) ;
else if(TERM_N * subdata[ i* ws + j] > vsmask[i*ws +j])
    vsmask[i*ws +j] += SHORT_TERM_P;
else vsmask[i*ws +j] -= SHORT_TERM_P;

```

```

if(vsmask[i*ws +j] == vlmask[i*ws +j] ) ;
else if(vsmask[i*ws +j] > vlmask[i*ws +j] )
    vlmask[i*ws +j]+= LONG_TERM_P;
    else vlmask[i*ws +j]-= LONG_TERM_P;
if(TERM_N * MIN(vsmask[i*ws +j],vlmask[i*ws +j]) < subdata[i*ws
+j])
    vmask[i*ws +j] = 255;
else vmask[i*ws +j] = 0;

```



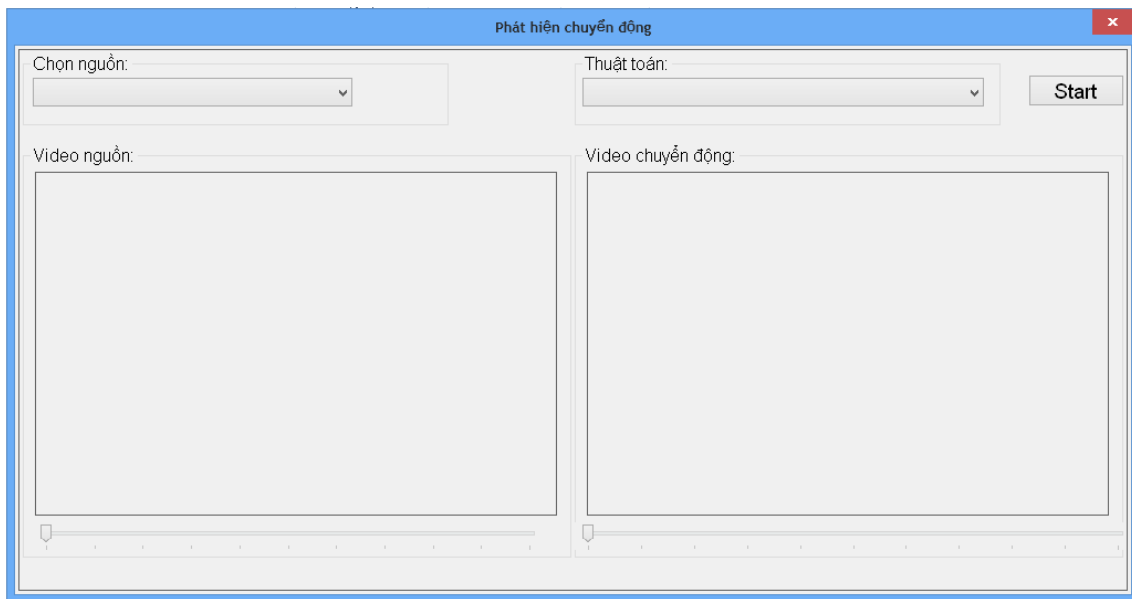
Hình 2.8. Lược đồ giải thuật phát hiện đối tượng chuyển động

### Chương 3

## CHƯƠNG TRÌNH THỬ NGHIỆM PHÁT HIỆN CHUYỂN ĐỘNG

### 3.1. Chương trình demo phát hiện chuyển động

Như đã trình bày ở phần trên, giải quyết bài toán phát hiện đột nhập bằng camera theo dõi dựa trên kết quả giải quyết bài toán phát hiện chuyển động. Học viên nghiên cứu xây dựng phần mềm phát hiện đột nhập bằng camera theo dõi dựa trên giải thuật phát hiện chuyển động đã được trình bày ở Chương 2, học viên giới thiệu các tính năng cơ bản của chương trình phát hiện đột nhập từ những đoạn video đầu vào đã được học viên quay ở những địa điểm, thời gian khác nhau để minh họa cho chương trình:



Hình 3.1. Giao diện chính của chương trình demo

Chương trình được viết bằng ngôn ngữ lập trình Visual C++ trên nền tảng bộ công cụ Microsoft Visual Studio 2013. Chương trình cài đặt thuật toán phát hiện chuyển động đã được trình bày tại Chương 2 và bổ sung phần định nghĩa khu vực theo dõi (Definition Controlled Areas) vùng hình tròn màu đỏ.

#### Tính năng chính của chương trình demo:

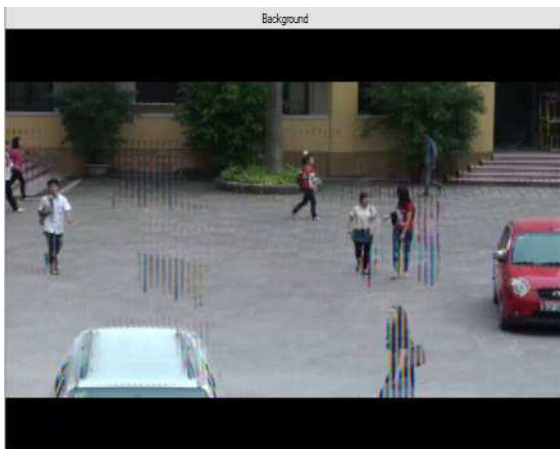
Phát hiện tất cả các chuyển động trong video dữ liệu đầu vào. Trong chương trình Demo, học viên sử dụng dữ liệu đầu vào là những đoạn video thu được từ camera ghi hình ở nhiều thời điểm và khung cảnh khác nhau. Do yêu cầu đặt ra của bài toán



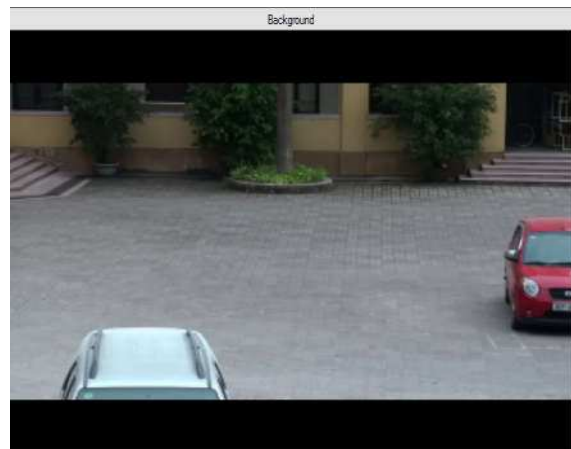
thực tế là tiết kiệm kinh phí và đảm bảo xử lý thời gian thực, nên việc sử dụng camera theo dõi không cần thiết phải dùng loại đắt tiền và hình ảnh chất lượng cao mà chỉ cần sử dụng camera thu video với dữ liệu lưu trữ dạng file .avi, kích thước: 640x480, số frame: < 25 fps (đây là dữ liệu lưu trữ cơ bản của camera kỹ thuật số).

Các lựa chọn của chương trình:

- Show background: Hiện hình ảnh về quá trình cập nhật nền liên tục của thuật toán để tạo ra nền tối ưu. Nền tối ưu là những frame chứa những pixel hầu như không thay đổi giá trị (những đối tượng không biến động) và đảm bảo thời gian tính toán nền này là nhỏ trong giới hạn chấp nhận được. Sau 50 frame bắt đầu khởi tạo nền đầu tiên, sau một thời gian tính toán, thuật toán trả về nền hầu như rất ít thay đổi.



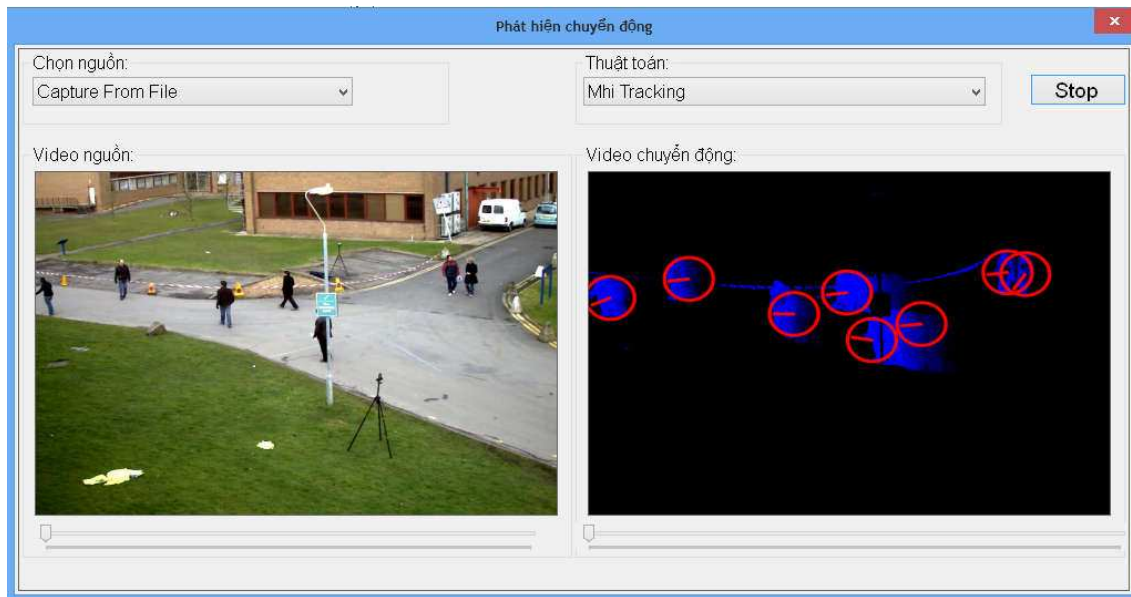
a) Nền ban đầu



b) Nền sau một thời gian ngắn

Hình 3.2. Mô tả việc xây dựng nền tối ưu của thuật toán

- Show video: Hiện hình ảnh thật thu được từ video. Sau 200 frame (khoảng 4s) thuật toán bắt đầu phát hiện đối tượng chuyển động. Những đối tượng chuyển động này được đánh dấu bằng hình tròn màu xanh.



Hình 3.3. Phát hiện và đánh dấu các đối tượng chuyển động

- Definition Controlled Areas: Định nghĩa khu vực giám sát là hình tròn màu đỏ có thể kéo thả trên nền video đang xử lý. Những đối tượng chuyển động xác định vào trong vùng này thì hình tròn cảnh báo đối tượng chuyển động màu xanh sẽ được chuyển sang hình tròn màu đỏ cảnh báo đối tượng đột nhập vào khu vực cần theo dõi.

```
if( feret.x > door.x
    && feret.y > door.y
```

```

&& (feret.width + feret.x < door.x + door.width)

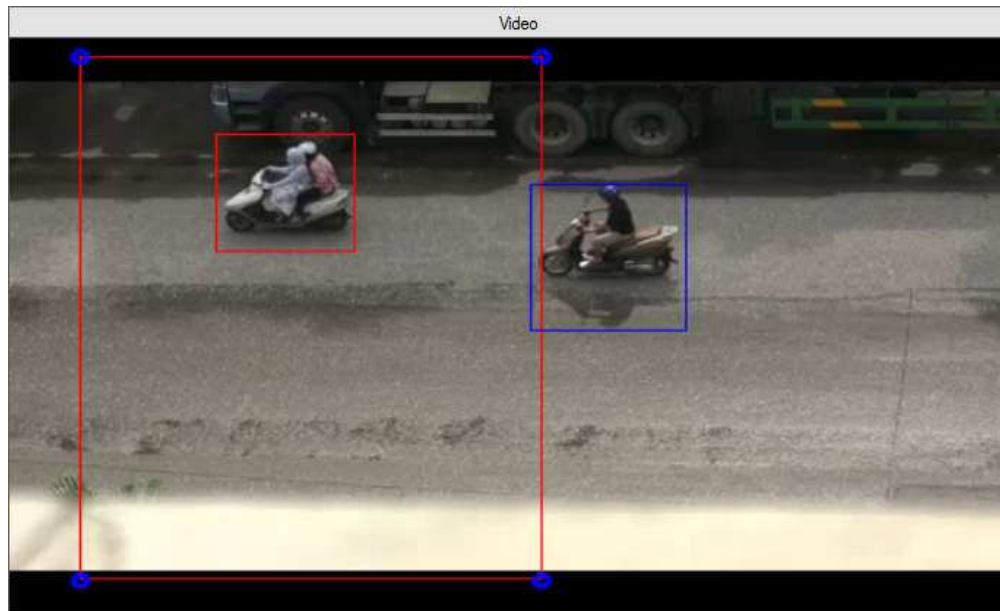
&& (feret.height + feret.y < door.y + door.height))

    DrawRect(img,feret,CV_RGB(255,0,0),1); //neu vi phạm ve mau do
else

    DrawRect(img,feret,CV_RGB(0,0,255),1); //neu khong ve mau xanh

//feret - Đối tượng chuyển động
//door - Định nghĩa vùng giám sát

```



Hình 3.4. Phát hiện và đánh dấu chữ nhật màu đỏ đối tượng vào khu vực giám sát

- Các tham số đầu vào có thể nhận những giá trị khác nhau nhằm tạo tính linh hoạt trong quá trình cài đặt phần mềm sử dụng ở điều kiện, môi trường khác nhau của video đầu vào. Người dùng sẽ thử nghiệm ở một điều kiện, môi trường nhất định với những tham số đầu vào khác nhau nhằm tìm ra bộ tham số đầu vào mà hiệu quả xử lý hình ảnh để phát hiện chuyển động là tốt nhất.

- Kết quả thử nghiệm: Chương trình thử nghiệm với dữ liệu đầu vào là những video mẫu được quay trong những thời điểm, khung cảnh và điều kiện khác nhau. Chương trình đều phát hiện được hầu như tất cả các đối tượng chuyển động, phát hiện và đánh dấu đối tượng chuyển động trong vùng cần theo dõi. Quá trình xử lý đảm bảo thời gian thực trong giới hạn có thể chấp nhận được.

Học viên đã cài đặt được thuật toán phát hiện chuyển động, trong đó bổ sung, cải tiến về tốc độ cập nhật nền (tùy vào từng điều kiện, có thể cập nhật nền theo từng frame một hoặc 3, đến 5 frame mới cập nền một lần. Điều này sẽ giảm thời gian tính toán trong quá trình tạo nền tối ưu); cải tiến thuật toán tính ngưỡng để xác định chính xác đối tượng chuyển động; bổ sung phần định nghĩa khu vực giám sát. Như vậy, chương trình demo này cơ bản đã xây dựng được những tính năng chính để giải quyết bài toán đặt ra: Phát hiện chuyển động bằng phương pháp trừ nền.

### **3.2. Bài toán Phát hiện đột nhập bằng camera theo dõi**

Giả định có một mục tiêu quan trọng cần được bảo vệ, xung quanh là tường rào chắn, có một cổng ra vào. Yêu cầu xây dựng hệ thống tự động phát hiện khi có đối tượng đột nhập vào mục tiêu bảo vệ.

*Giải quyết bài toán:*

- Khảo sát bên trong, bên ngoài mục tiêu. Xây dựng 02 mô hình camera giám sát ở vòng ngoài và vòng trong mục tiêu.

- + Vòng ngoài: Giám sát và phát hiện những đối tượng đột nhập từ bên ngoài vào mục tiêu thông qua cửa chính, vượt hàng rào bảo vệ.

- + Vòng trong: Giám sát cửa vào cầu thang máy, cầu thang bộ của các tầng, cửa những phòng quan trọng.

- Lên sơ đồ camera theo dõi bao quát hết mục tiêu, giả định dùng 8 camera lắp đặt tại các địa điểm để quan sát tất cả những hướng có thể xâm nhập mục tiêu từ bên ngoài, dùng 8 camera quan sát các điểm bên trong mục tiêu (4 cửa thang máy và cầu thang bộ, 4 phòng quan trọng).

- Xây dựng phần mềm tích hợp việc xử lý đồng thời những camera giám sát để đưa ra đối tượng xâm nhập vào mục tiêu. Việc xử lý đối với mỗi camera được thực hiện như chương trình demo đã giới thiệu ở phần trên, trong đó, mỗi camera sẽ định nghĩa những vùng giám sát riêng. Trên cơ sở lắp đặt, kiểm tra thử nghiệm với những thời điểm khác nhau để lựa chọn bộ giá trị ngưỡng phù hợp để quá trình xử lý, phát hiện đối tượng đột nhập đạt hiệu quả tối ưu nhất có thể.

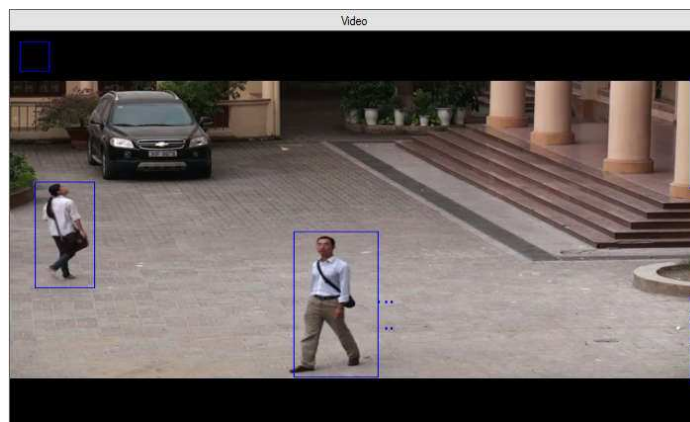
- Khi phát hiện đối tượng xâm nhập vào khu vực giám sát, chương trình phần mềm sẽ tự động cảnh báo theo các chế độ người quan sát cài đặt như: bằng âm thanh báo động, tô màu đối tượng, lưu những đoạn video có đối tượng đột nhập trong ổ cứng của máy tính...

Học viên đã giới thiệu demo chương trình với những tính năng chính phát hiện đột nhập và đưa ra hướng phát triển hoàn thiện chương trình phát hiện đột nhập bằng hệ thống camera theo dõi.

### 3.3. Một số kết quả thực nghiệm

Chương trình đã được thử nghiệm với những video đầu vào quay ở nhiều khung cảnh, điều kiện và thời điểm khác nhau. Định dạng dữ liệu đầu vào: file.avi, kích thước: 480 x 320, 25fps. Kết quả của chương trình đã phát hiện toàn bộ chuyển động của các đối tượng trong video, phát hiện tất cả các đối tượng khi xâm nhập vào vùng định nghĩa giám sát. Thời gian xử lý phát hiện đối tượng chuyển động chậm hơn không đáng kể so với tốc độ của video gốc nhưng có thể chấp nhận được. Với những điều kiện tốt, thời gian cập nhận nền rất nhanh, do đó, việc phát hiện đối tượng chính xác trong thời gian ngắn. Sau đây là một số minh họa kết quả chương trình phát hiện xâm nhập với những trường hợp video đầu vào khác nhau.

*Trường hợp 1:* Video có ít đối tượng, các đối tượng chuyển động tách rời nhau, điều kiện ánh sáng tương đối tốt, việc tạo nền tối ưu rất nhanh, phát hiện nhanh đối tượng chuyển động.

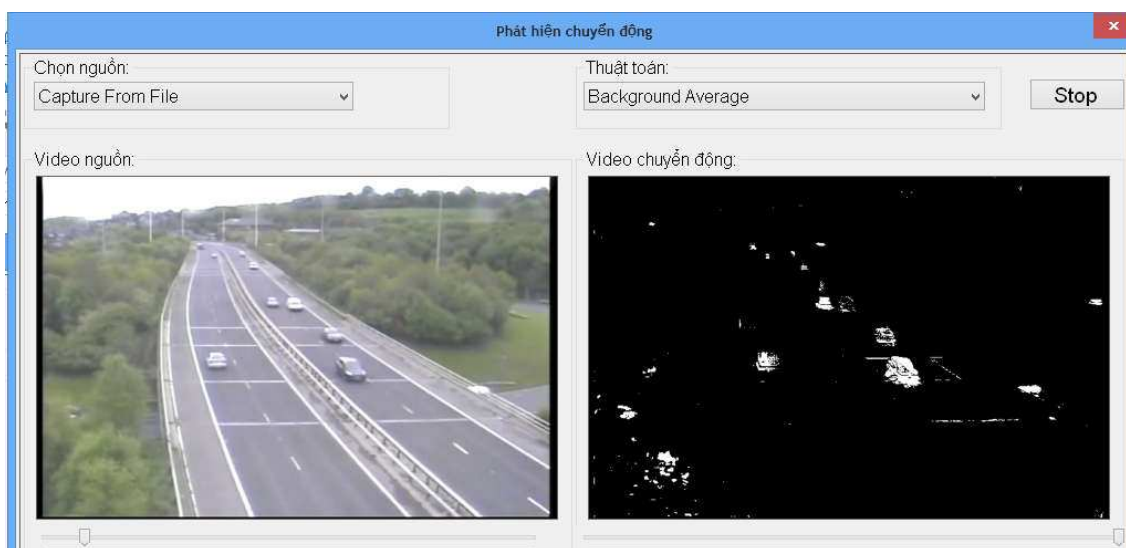


Hình 3.5. Phát hiện đối tượng chuyển động đơn lẻ trong điều kiện ánh sáng tốt

- *Trường hợp 2:* Video có nhiều đối tượng, các đối tượng chuyển động quy luật khác nhau, điều kiện ánh sáng tương đối tốt, việc tạo nền tối ưu chậm hơn so với trường hợp 1, tuy nhiên tốc độ tương đối nhanh, phát hiện nhanh và đánh dấu tất cả các đối tượng chuyển động.



Hình 3.6. Cập nhật nền nhiều đối tượng chuyển động, điều kiện ánh sáng tốt



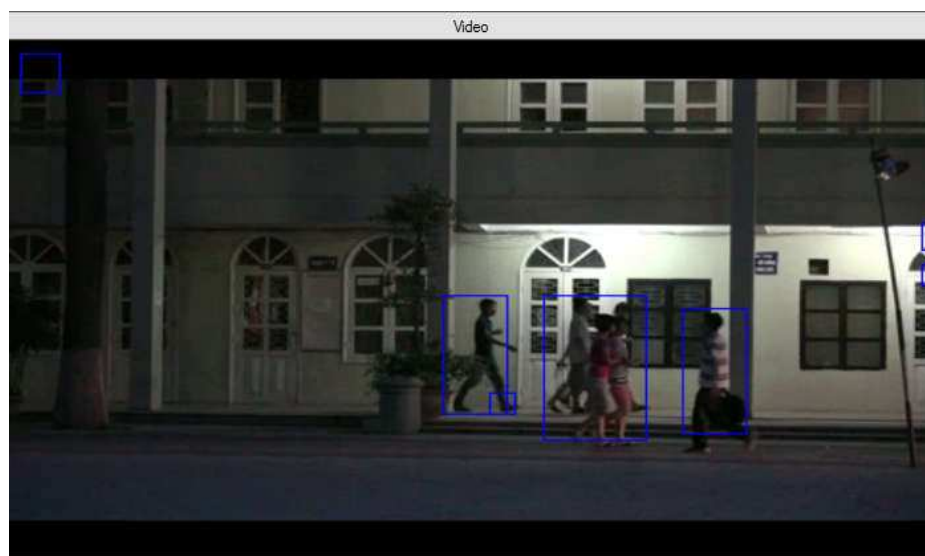
Hình 3.7. Phát hiện chuyển động trong điều kiện có nhiều đối tượng chuyển động với quy luật khác nhau, ánh sáng tốt



- *Trường hợp 3:* Video có nhiều đối tượng, các đối tượng chuyển động quy luật khác nhau, điều kiện ánh sáng tương đối yếu, việc tạo nền tối ưu chậm hơn so với trường hợp 2, phát hiện và đánh dấu đối tượng chậm hơn nhưng đã phát hiện được tất cả các đối tượng chuyển động.



Hình 3.8. Quá trình cập nhật nền trong điều kiện ánh sáng yếu



Hình 3.9. Phát hiện chuyển động trong điều kiện nhiều đối tượng chuyển động với quy luật khác nhau, ánh sáng yếu

- *Trường hợp 4:* Video đầu vào có nhiều phương tiện giao thông chuyển động với vận tốc khác nhau và tương đối cao, điều kiện ánh sáng tương đối tốt, thì quá trình tạo nền tối ưu nhanh, phát hiện nhanh tất cả các phương tiện giao thông chuyển động trong video.



Hình 3.10. Phát hiện phương tiện giao thông chuyển động với  
tốc độ tương đối cao trong điều kiện ánh sáng tốt



## KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### Kết luận

Giám sát và bảo vệ các mục tiêu quan trọng là bài toán được đặt ra từ rất sớm nhằm đảm bảo tuyệt đối an toàn cho các mục tiêu này. Trong thời đại ngày nay, không chỉ những mục tiêu truyền thống cần được giám sát và bảo vệ như: những mục tiêu quan trọng về chính trị, kinh tế, an ninh, quốc phòng mà việc giám sát, bảo vệ mục tiêu được áp dụng rộng rãi và phổ cập hóa trong xã hội như tại các tổ chức, cơ quan, công ty và cá nhân người dùng.

Khoa học công nghệ phát triển đã hỗ trợ đắc lực cho con người trong những công việc khó khăn, nguy hiểm. Công nghệ thông tin và công nghệ điện tử ngày nay đã cho phép sử dụng những thiết bị phần cứng (các loại camera giám sát, máy tính, đường truyền...) và phần mềm hỗ trợ hiệu quả cho việc giám sát và bảo vệ mục tiêu.

Bài toán phát hiện đột nhập bằng camera theo dõi quy về bài toán cơ bản là phát hiện chuyển động đã được nghiên cứu từ rất sớm, xuất hiện ngay từ khi kỹ thuật ảnh số ra đời. Đã có nhiều nghiên cứu trước đó để giải quyết bài toán phát hiện chuyển động, tuy nhiên về mặt thời gian và kết quả phát hiện chuyển động vẫn chưa được tối ưu. Qua quá trình tìm hiểu, nghiên cứu của mình về bài toán phát hiện chuyển động, học viên đã đạt được một số kết quả như sau:

- Tìm hiểu lịch sử phát triển của ảnh số và kỹ thuật xử lý ảnh số, những ứng dụng của nó.

- Tìm hiểu và nắm được những khái niệm cơ bản về ảnh số và kỹ thuật xử lý ảnh số; các loại camera giám sát; lịch sử phát triển của bài toán phát hiện chuyển động và những ứng dụng thực tế của bài toán này.

- Tìm hiểu về thuật toán phát hiện chuyển động dựa trên phương pháp trừ nền và sử dụng ước lượng Entropy của TS. Shih-Chia Huang công bố trên tạp chí IEEE tháng 1/2011 (Institute of Electrical and Electronics Engineers – Viện kỹ nghệ điện và điện tử).

- Cài đặt và thử nghiệm thành công chương trình phát hiện chuyển động dựa trên thuật toán do tác giả Shih-Chia Huang đưa ra. Trong đó, học viên cải tiến để tăng

tốc độ cập nhật nền, cải tiến thuật toán tính ngưỡng chuyển động, bổ sung tính năng định nghĩa vùng giám sát để phù hợp với yêu cầu của bài toán thực tế.

- Thử nghiệm phát hiện chuyển động với một số môi trường, điều kiện và thời điểm khác nhau để kiểm chứng tính đúng đắn của thuật toán.

### **Hướng phát triển**

Trong thời gian tới, chúng tôi sẽ tập chung nghiên cứu, hoàn thiện thêm các nội dung sau:

- Hoàn thiện chương trình phát hiện và cảnh báo đối tượng xâm nhập vào những khu vực bảo vệ với tính năng: xử lý đồng thời video từ hệ thống camera quan sát, định nghĩa vùng giám sát cho phù hợp với điều kiện thực tế, cảnh báo bằng âm thanh, hình ảnh và lưu trữ video của đối tượng đột nhập.

- Kiểm thử chương trình phát hiện chuyển động trong những môi trường, điều kiện và thời gian khác nhau để kiểm tra tính hiệu quả và ứng dụng thực tế của chương trình.

- Xây dựng mô hình chương trình phát hiện và cảnh báo đối tượng xâm nhập xử lý đồng thời với nhiều camera khác nhau áp dụng cho một bài toán cụ thể.

- Nghiên cứu thuật toán theo dõi đối tượng (Tracking Object) kết hợp với thuật toán phát hiện chuyển động trên để giải quyết các bài toán nâng cao hơn: phát hiện và theo dõi đối tượng trong vùng camera quan sát; đếm lưu lượng giao thông, sản phẩm...

## TÀI LIỆU THAM KHẢO

### Tiếng Việt

[1] Lê Quyết Thắng, Phan Tuấn Tài, Dương Văn Hiếu, *Giáo trình Lý thuyết thông tin*, Đại học Cần Thơ, Nhà xuất bản Khoa học và Kỹ thuật, 2010.

[2] Đỗ Năng Toàn, Phạm Việt Bình, *Giáo trình xử lý ảnh*, Đại học Thái Nguyên, Nhà xuất bản Khoa học và Kỹ thuật, 2008.

[3] Nguyễn Thị Lan Hương (2006), *Phát hiện – phân loại – theo dõi đối tượng chuyển động trong hệ thống giám sát*, Luận văn Đại học.

### Tiếng Anh

[4]. Shih-Chia Huang, *An Advanced Motion Detection Algorithm with Video Quality Analysis for Video Surveillance Systems*, IEEE Transactions on circuits and systems for video technology, Vol 21, No. 1, January 2011.

[5]. [Http://opencv.org](http://opencv.org)

[6] C.Stauffer and W.Grimson, "Adaptive Background mixture models for Real-time tracking" pp. 750-755, 2009.