

BÁO CÁO PROJECT II

BỘ MÔN HỆ THỐNG THÔNG TIN

ĐỀ TÀI : MULTIPLE TRACKING

GV hướng dẫn : TS. Nguyễn Thị Oanh

Nhóm sinh viên thực hiện

Lữ Mạnh Hùng

20142078

Nội dung chính

Báo cáo tập trung vào xử lý ảnh ở mức thấp và mức giữa được trình bày theo trình tự thực hiện để tracking nhiều đối tượng:

1. Detect nhiều đối tượng
2. Tracking các đối tượng vừa detect được
3. Phân biệt các đối tượng để gán nhãn

GIỚI THIỆU VÀ MÔ TẢ ĐỀ TÀI

1. Giới thiệu đề tài

- Theo dõi nhiều đối tượng (multiple tracking) là cơ sở của hệ thống giám sát thông minh bằng hình ảnh
- Minh họa: các hệ thống giám sát giao thông, dự báo cháy,...

2. Các bài toán cần giải quyết trong đề tài

- Bài toán 1: Phát hiện các đối tượng chuyển động: là tách các đối tượng chuyển động ra khỏi nền của chúng
- Bài toán 2: Theo dõi đối tượng đó là công việc đưa ra một chuỗi các hành vi của đối tượng chuyển động
- Bài toán 3: Phân biệt các đối tượng để gán nhãn. Bài toán này giải quyết tình huống độn độ giữa các đối tượng

PHƯƠNG PHÁP PHÁT HIỆN ĐỐI TƯỢNG

► Phát hiện đối tượng chuyển động

Các khung hình video

PHÁT HIỆN CÁC VÙNG ẢNH NỔI
(FOREGROUND DETECTION)

XỬ LÝ CÁC VÙNG ẢNH NỔI
(FOREGROUND PROCESSING)

Các đối tượng chuyển động,
các tính chất của chúng

PHƯƠNG PHÁP PHÁT HIỆN ĐỐI TƯỢNG

Phát hiện vùng ảnh nổi

1. Phương pháp trừ nền (Background subtraction)

- Xét một điểm ảnh $I(x,y)$ trong một khung hình mới và $B(x,y)$ là điểm ảnh trên ảnh nền đều có tọa độ (x,y) . Điểm ảnh I được coi là điểm ảnh nổi (foreground) nếu:

$$|I(x,y) - B(x,y)| > \tau$$

Trong đó τ là một ngưỡng được định nghĩa từ trước

- Ảnh nền B được cập nhật bằng cách sử dụng phương pháp lọc Infinite Impulse Response (IIR) theo công thức sau:

$$B_{t+1} = \alpha I_t + (1 - \alpha) B_t$$

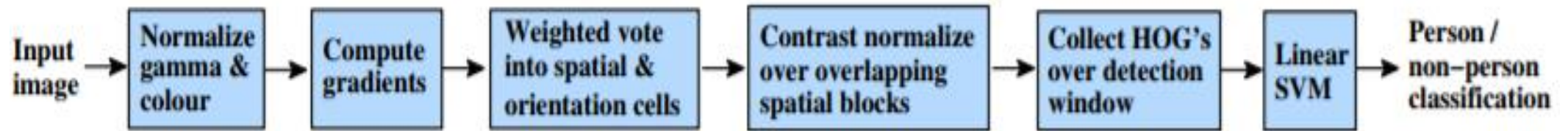
- Trong đó I_t là điểm ảnh nổi tại thời điểm t , B_t là điểm ảnh nền tại thời điểm t , α là tham số được định nghĩa trước

PHƯƠNG PHÁP PHÁT HIỆN ĐỐI TƯỢNG

► Phát hiện vùng ảnh nổi

2. Sử dụng vectơ HoG (Histograms of Oriented) để xác định người

- Các bước thực hiện:



- So sánh với Background Subtraction: phương pháp này đem lại kết quả detect chính xác hơn nhưng đòi hỏi tài nguyên tính toán lớn nên kết quả hoạt động chậm, không phù hợp với bài toán live tracking.

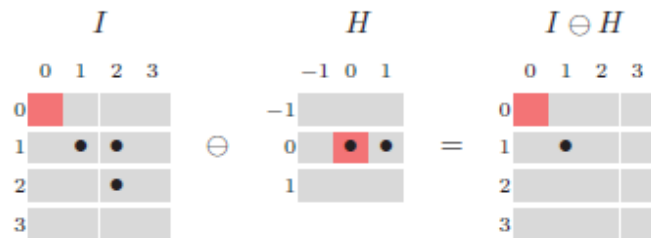
PHƯƠNG PHÁP PHÁT HIỆN ĐỐI TƯỢNG

► Xử lý các vùng ảnh nổi (xử lý nhiễu)

1. Phép co ảnh – Erosion

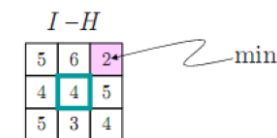
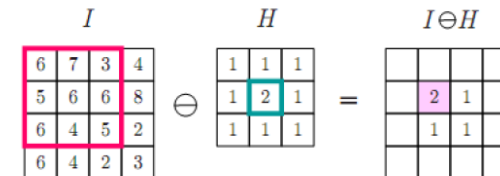
$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

- Phép co ảnh nhị phân của tập hợp A bởi phần tử cấu trúc B là tập hợp các điểm z (z nằm ở tâm điểm của phần tử cấu trúc B) sao cho B_z là tập con của A.



- Phép toán co ảnh của ảnh xám I với cấu trúc phần tử không phẳng H tại vị trí (x, y) của ảnh được xác định như sau [1][2][4][5][6]:

$$(I \ominus H)(x, y) = \min(I(x+i, y+j) - H(i, j) \mid (i, j) \in D_H)$$



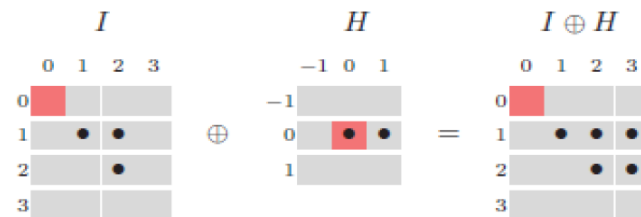
PHƯƠNG PHÁP PHÁT HIỆN ĐỐI TƯỢNG

► Xử lý các vùng ảnh nổi (xử lý nhiễu)

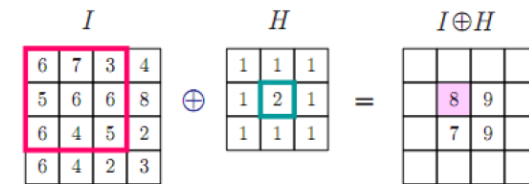
2. Phép giãn ảnh – Dilation

$$A \oplus B = \left\{ z \mid \left[\left(\hat{B} \right)_z \cap A \right] \subseteq A \right\}$$

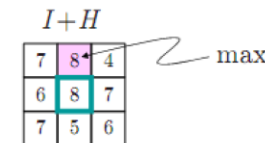
- Phép giãn ảnh nhị phân của tập A bởi phần tử cấu trúc B là tập hợp của các điểm z (z là tâm của phần tử cấu trúc B trên tập A) sao cho phản xạ của Bz giao với tập A tại ít nhất một điểm.



$$(I \oplus H)(x, y) = \max(I(x+i, y+j) + H(i, j) \mid (i, j) \in D_H)$$



- Phép toán co ảnh của ảnh xám I với cấu trúc phần tử không phẳng H tại vị trí (x, y) của ảnh I được xác định như sau [1][2][4][5][6]:



PHƯƠNG PHÁP PHÁT HIỆN ĐỐI TƯỢNG

Xử lý các vùng ảnh nổi (xử lý nhiễu)

3. Phép mở ảnh – Opening

- Gọi A là hình ảnh gốc và B là phần tử cấu trúc, (\odot) là ký hiệu của phép mở ảnh giữa tập hợp A và phần tử cấu trúc B, phép mở ảnh được xác định bởi công thức:

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

- Tác dụng phép mở: Phép co giúp xóa các điểm nhiễu hạt không mong muốn > làm co toàn ảnh > dùng phép giãn để giãn lại ảnh.

PHƯƠNG PHÁP TRACKING CÁC ĐỐI TƯỢNG

Kalman filter

- Lọc Kalman nhằm ước lượng giá trị đích thực của một đối tượng trong ảnh, bằng cách dự đoán giá trị của nó và tính độ tin cậy (hay độ bất định) của dự đoán đó.
- Đồng thời đo đạc giá trị (nhưng bị sai số vì có các nhiễu), sau đó lấy một trung bình có trọng giữa giá trị dự đoán và giá trị đo đạc được, làm giá trị ước lượng.
- Mô hình toán học:

Gọi x_k là vector giá trị thực sự của một đối tượng trong ảnh tại thời điểm thứ k . Ta sẽ giả sử x_k biến đổi theo qui luật sau:

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$

trong đó F_k là ma trận thay đổi trạng thái (state transition matrix), u_k là vector điều khiển, B_k là ma trận điều khiển, còn w_k là nhiễu ngẫu nhiên, với giả sử là nó có phân bố Gaussian (phân bố normal nhiều chiều) $N(0, Q_k)$, trong đó Q_k là ký hiệu của ma trận hiệp phương sai tương ứng.

PHƯƠNG PHÁP TRACKING CÁC ĐỐI TƯỢNG

Kalman filter

Tại mỗi thời điểm thứ k , có một đo đạc (measurement) trạng thái x_k cho kết quả là

$$z_k = H_k x_k + v_k$$

trong đó H_k là ma trận của mô hình quan sát, còn v_k là nhiễu trong lúc đo đạc, và ta giả sử nhiễu này cũng tuân theo một phân bố Gaussian $N(0, R_k)$.

Các ma trận F_k, B_k, H_k được coi là đã biết. Ta giả sử thêm là các nhiễu $\{v_1, \dots, v_k, w_1, \dots, w_k\}$ là một bộ biến ngẫu nhiên độc lập và cũng độc lập với trạng thái ban đầu x_0 .

PHƯƠNG PHÁP TRACKING CÁC ĐỐI TƯỢNG

Kalman filter

Thuật toán ước lượng như sau: Có thể chia nó thành 2 bước, bước dự đoán ban đầu (predict) và bước điều chỉnh sau đó (update)

Predict:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k + w_k,$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

- Ở đây $\hat{x}_{k|k-1}$ là ký hiệu dự đoán giá trị của x_k dựa trên thông tin về giá trị tại thời điểm $k-1$, còn $\hat{x}_{k|k}$ là ước lượng giá trị của x_k sau khi đã sử dụng mọi thông tin tại thời điểm k . Ma trận P dùng để chỉ (ước lượng) ma trận hiệp phương sai của ước lượng của x .

PHƯƠNG PHÁP TRACKING CÁC ĐỐI TƯỢNG

Kalman filter

Update:

- Độ lệch so với quan sát (measurement residual): $y_k = z_k - H_k \hat{x}_{k|k-1}$
- Thặng dư hiệp phương sai (residual covariance): $S_k = H_k P_{k|k-1} H_k^T + R_k$

Kalman tối ưu: $K_k = P_{k|k-1} H_k^T S_k^{-1}$

Ước lượng được điều chỉnh (updated estimate): $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k$

Hiệp phương sai cho ước lượng mới (updated estimate covariance): $P_{k|k} = (1 - K_k H_k) P_{k|k-1}$

PHƯƠNG PHÁP GÁN NHÃN ĐỐI TƯỢNG

So sánh histogram

Histogram của ảnh đa mức xám: $[0; L-1]$ là hàm rời rạc:

$$h(r_k) = n_k$$

Với r_k là thành phần mức xám thứ k

n_k số lượng pixel có mức xám là r_k

Dạng chuẩn hóa:

$$h(r_k) = \frac{n_k}{n}$$

Với n : tổng số pixel trong ảnh

Biểu diễn ảnh = 1 vector đặc trưng

- Số chiều vector = số bin histogram
- Giá trị mỗi phần tử = giá trị mỗi bin histogram

PHƯƠNG PHÁP GÁN NHÃN ĐỐI TƯỢNG

► So sánh histogram

- Ưu điểm
 - Phản ánh phân bố màu sắc trong ảnh
 - Bất biến với phép quay ảnh (không làm méo)
 - Bất biến với phép dịch ảnh
- Nhược điểm
 - Không phản ánh tính không gian
 - Nhạy với phép thay đổi ánh sáng



PHƯƠNG PHÁP GÁN NHÃN ĐỐI TƯỢNG

► So sánh histogram

- Để so sánh 2 histogram (H_1 và H_2), đầu tiên chúng ta chọn một độ đo ($d(H_1, H_2)$) để thể hiện khi nào hai histogram khớp nhau:

1. **Correlation (CV_COMP_CORREL)**

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I (H_1(I) - \bar{H}_1)^2 \sum_I (H_2(I) - \bar{H}_2)^2}}$$

where

$$\bar{H}_k = \frac{1}{N} \sum_J H_k(J)$$

and N is the total number of histogram bins.

2. **Chi-Square (CV_COMP_CHISQR)**

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)}$$

3. **Intersection (method=CV_COMP_INTERSECT)**

$$d(H_1, H_2) = \sum_I \min(H_1(I), H_2(I))$$

4. **Bhattacharyya distance (CV_COMP_BHATTACHARYYA)**

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_1 \bar{H}_2 N^2}} \sum_I \sqrt{H_1(I) \cdot H_2(I)}}$$

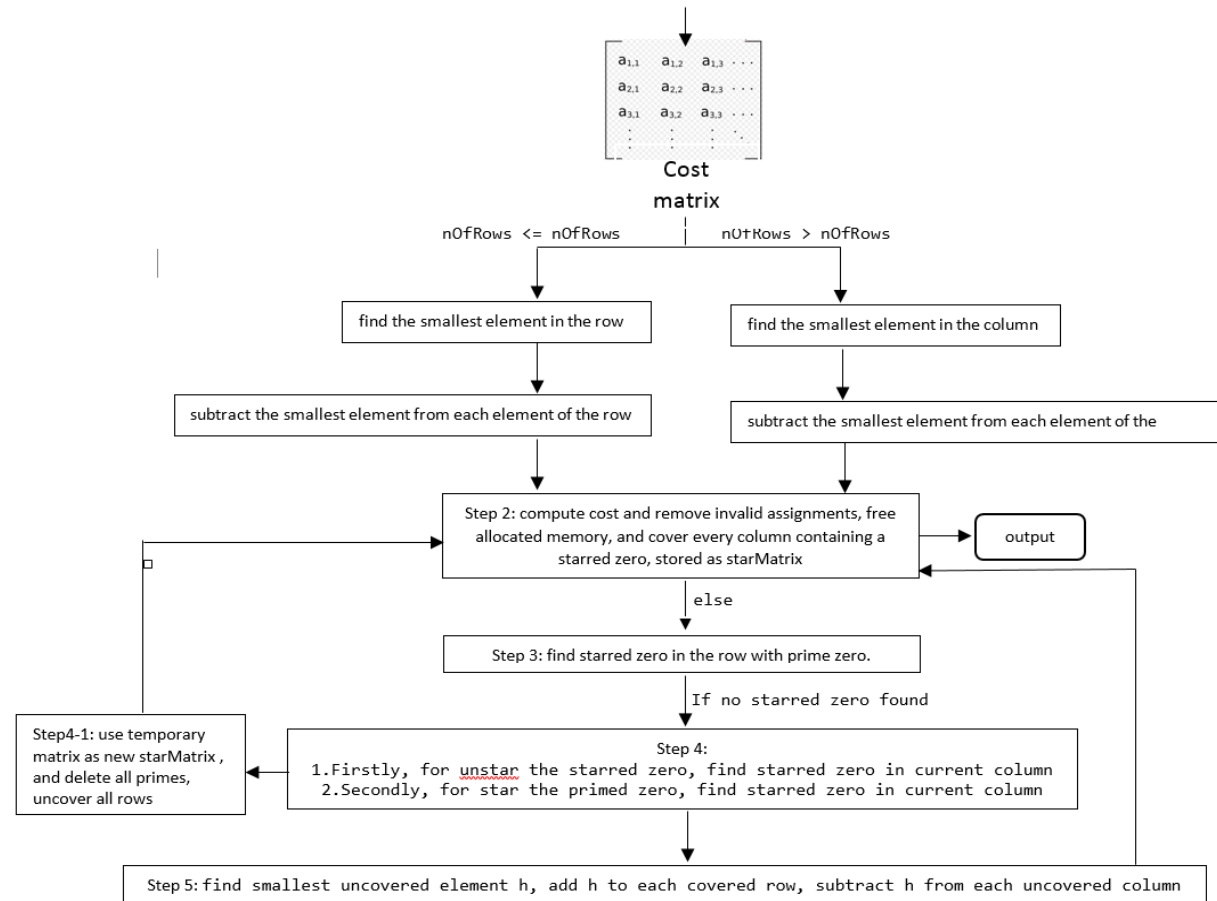
PHƯƠNG PHÁP GÁN NHÃN ĐỐI TƯỢNG

Hungarian Algorithm

- Ý tưởng của giải thuật dựa trên việc so sánh khoảng cách giữa các đối tượng của frame này với các đối tượng của frame liền trước để so khớp chúng với nhau.
- Input của thuật toán là một ma trận trọng số có:
 - Hàng là vị trí của các đối tượng ở frame trước.
 - Cột là vị trí các đối tượng ở frame hiện tại.
 - Giá trị phần tử (i, j) của ma trận là khoảng cách giữa đối tượng thứ i và đối tượng thứ j
- Đánh giá thuật toán: không xử lý được tình huống nhận biết một đối tượng xuất hiện ở frame thứ i sau đó biến mất và xuất hiện lại ở frame thứ $i+x$ ($x \geq 2$).

PHƯƠNG PHÁP GÁN NHÃN ĐỐI TƯỢNG

► Hungarian Algorithm



CÀI ĐẶT CÁC PHƯƠNG PHÁP

Background Subtraction

1. Background Subtraction (MOG2)

- Phương pháp này bản chất là Gaussian Mixture dựa trên thuật toán phân đoạn Background/Foreground.
- Một yếu tố quan trọng của thuật toán này là nó chọn một số lượng thích hợp các hàm phân bố Gaussian cho mỗi pixel (với phương pháp MOG thì cố định 3-5 hàm).
- Điều này cung cấp khả năng thích nghi tốt hơn với những cảnh có độ sáng thay đổi.
- Phương pháp này cho phép ta phát hiện được bóng của đối tượng được đánh dấu bằng màu xám

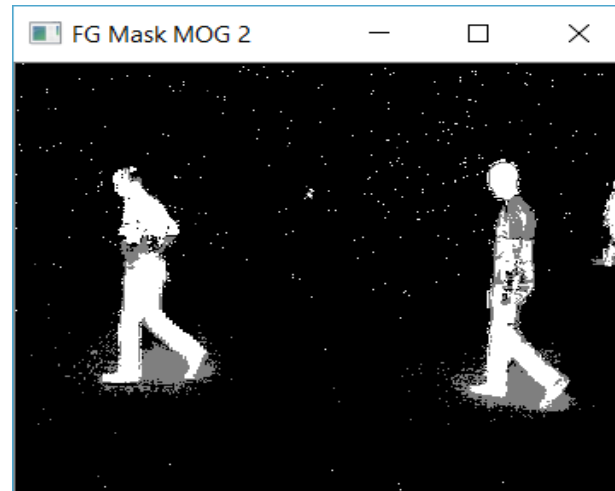
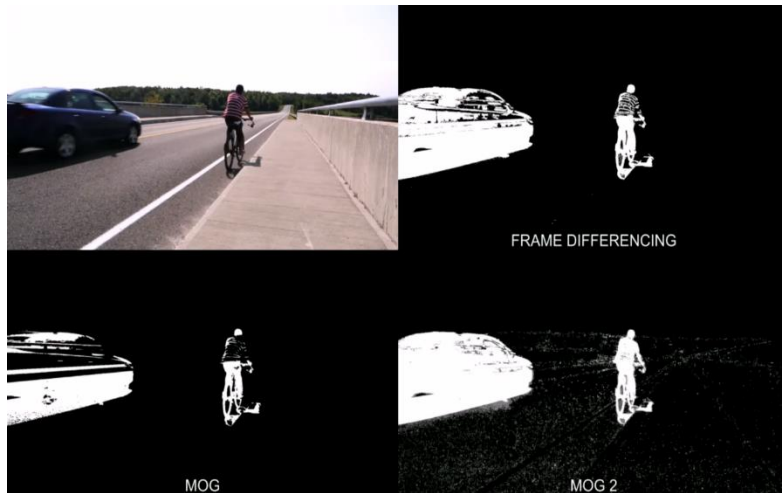
CÀI ĐẶT CÁC PHƯƠNG PHÁP

► Hungarian Algorithm

1. Background Subtraction (MOG2)

- Khởi tạo đối tượng trong OpenCV:

```
pMOG2 = createBackgroundSubtractorMOG2(500, 10, true);
```



CÀI ĐẶT CÁC PHƯƠNG PHÁP

Kalman Filter

- Các tham số khởi tạo của đối tượng Kalman filter:

- `KalmanFilter KF(stateSize = 4, measSize = 2, contrSize = 0, type = CV_32F);`

Vì đề tài cần xác định vị trí của đối tượng trên không gian 2 chiều nên `measSize = 2` chính là tính tọa độ (x,y) của đối tượng.

- Transition State Matrix F: `(Mat_<float>(4, 4) << 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1);`
- Measurement Matrix H: `Mat_<float> measurement(2, 1); measurement.setTo(Scalar(0));`

Process Noise Covariance Matrix Q: `setIdentity(KF.processNoiseCov, Scalar::all(1e-4));`

Measures Noise Covariance Matrix R: `setIdentity(KF.measurementNoiseCov, Scalar::all(10));`

- Trong bước khởi tạo cần chuyển vào tọa độ của đối tượng (x,y). Thuật toán Kalman filter sẽ thực hiện predict tọa độ của đối tượng đó và sau đó là update vị trí chính xác của đối tượng.

CÀI ĐẶT CÁC PHƯƠNG PHÁP

Histogram và đối tượng đựng độ

- Hình ảnh của mỗi đối tượng (là kết quả của phép trừ nền) sẽ được tách làm 3 dải màu R-G-B, và tính histogram trên mỗi mức màu này.
- Để tăng tốc độ tính toán ta tính histogram với $\text{histSize} = 8$ do đó sẽ có $8 \times 3 = 24$ miền giá trị. Việc sử dụng số bin ít cũng giúp làm nổi bật tích chất về khoảng cách giữa các đối tượng khi so sánh.
- Để phân biệt các đối tượng trước hết cần xây dựng Struct các đối tượng có thành phần là ID, và các giá trị histogram r,g,b.

for each contour:

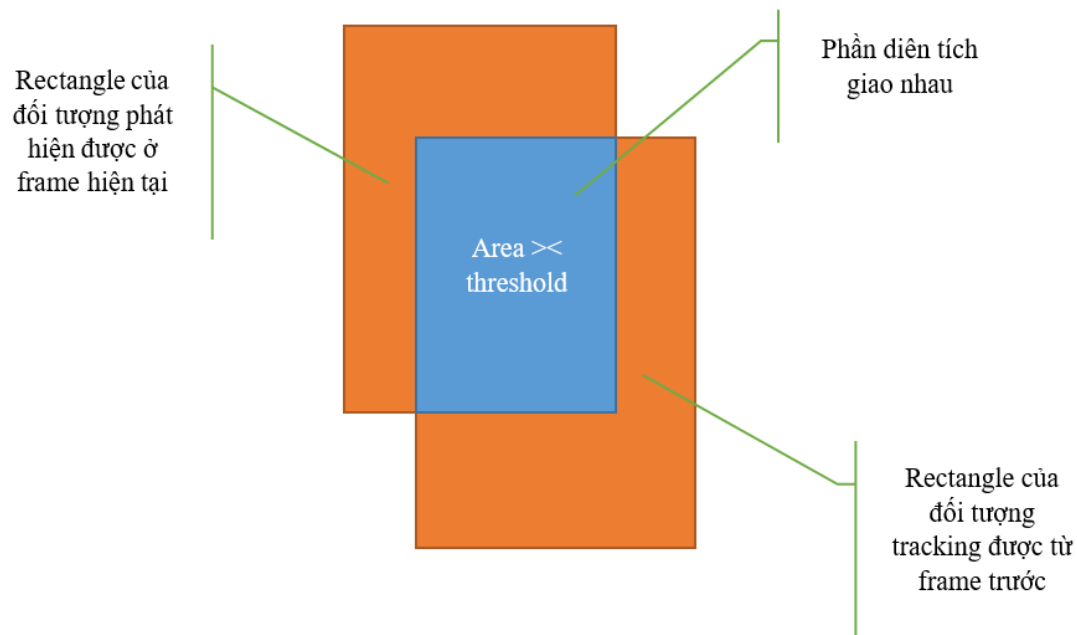
if its already tracked with ID 'A': `kalmanFilter_with_id_A.track(x,y);`

else `createNewKalmanFilterWithID A`

CÀI ĐẶT CÁC PHƯƠNG PHÁP

► Xử lý độ chồng độ trong MIL/KCF

- `MultiTracker trackers(trackingAlg);`
- Ý tưởng: sử dụng phép so sánh mức độ chồng lẫn của mỗi đối tượng
- Phép toán chồng lẫn ta sử dụng là tính diện tích phần giao nhau của hai đối tượng



TỔNG KẾT PROJECT II

Kết quả đạt được

- Đề tài: Multiple tracking.
 - Tìm hiểu cơ bản về các kiến thức nền tảng xử lý ảnh
 - Có cơ hội tiếp cận và tìm hiểu về một lĩnh vực thiết thực là xử lý ảnh.
 - Có cái nhìn mới về nhận dạng xử lý video trong các ứng dụng
 - Xây dựng được các ứng dụng xử lý ảnh thực tế sử dụng thư viện opencv
 - Có sự so sánh giữa các phương pháp tracking hiện có
 - Đề tài có khả năng ứng dụng thực tiễn với các ngành các lĩnh vực trong robot , giám sát nhận dạng qua camera ,...