

WEEK 4

Object Tracking



Đặt vấn đề :

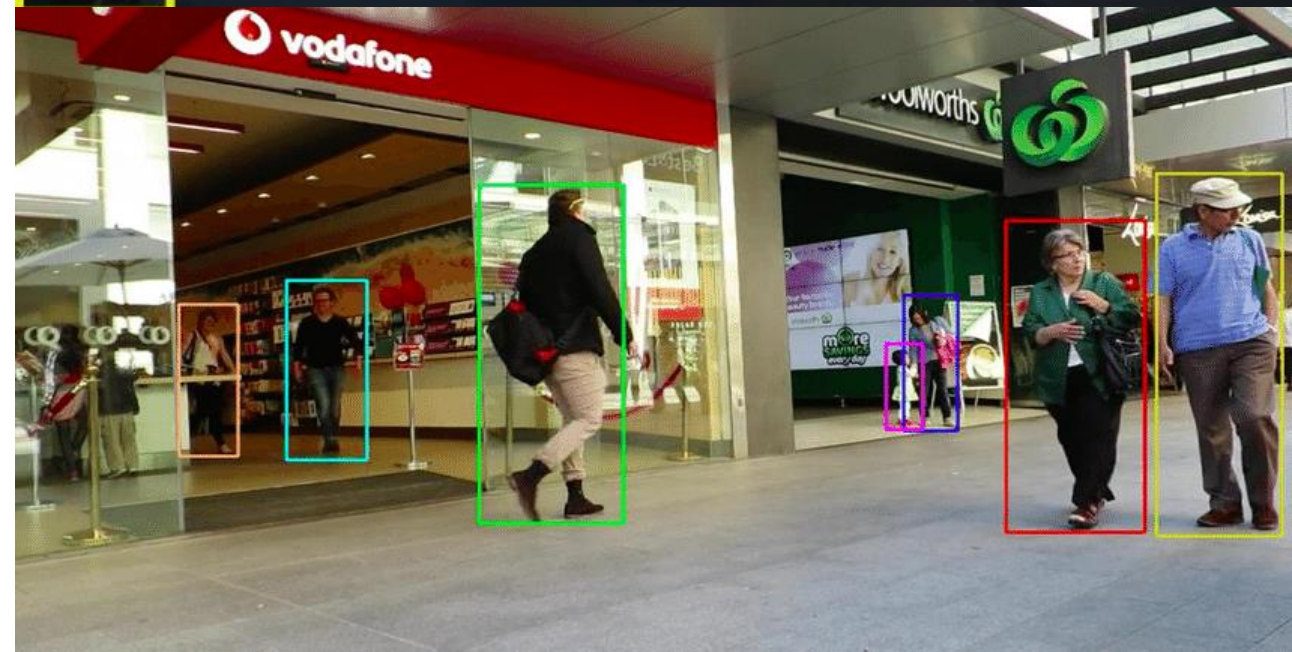
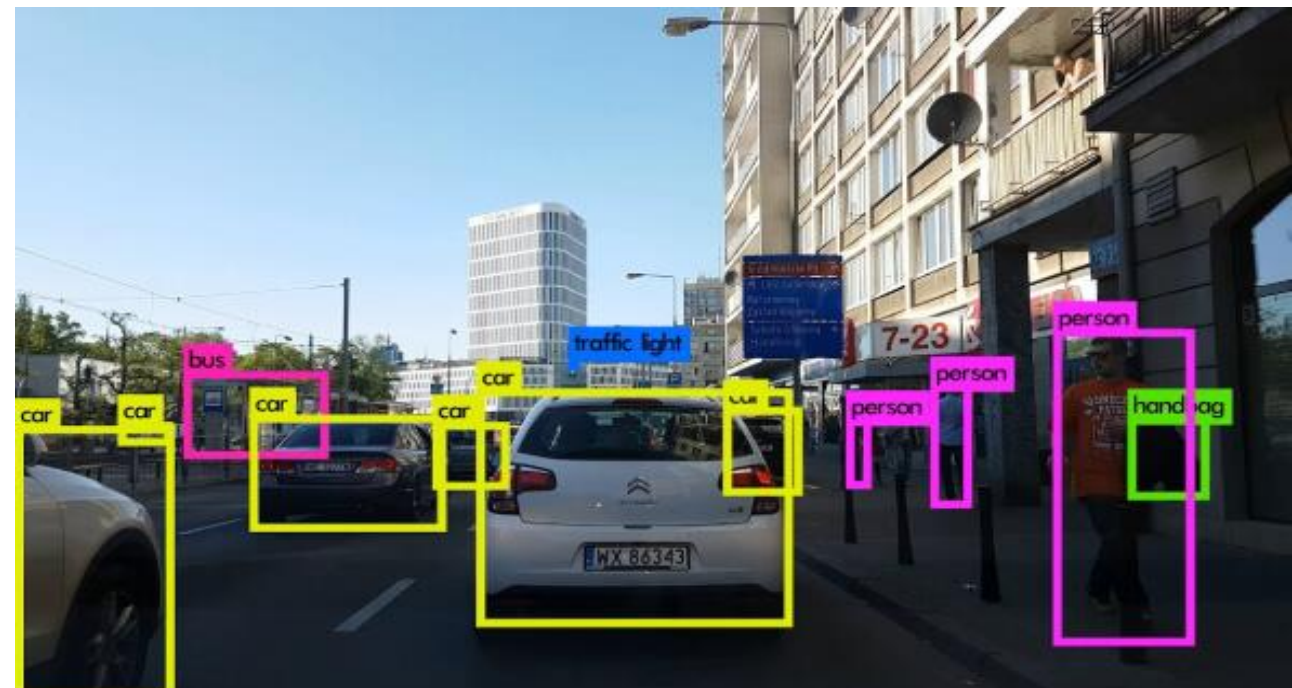
Lấy một ví dụ đơn giản: Việc lắp đặt camera giám sát những nơi đường phố công cộng là hoàn toàn cần thiết để quản lý trật tự xã hội.

Object Tracking là một trong những giải pháp hiệu quả giúp ta theo dõi từng đối tượng trong frame hình từ đó giúp ta dễ dàng phát hiện ngăn chặn , quản lý tội phạm, những hành vi gây mất trật tự xã hội

Giới thiệu: *Object tracking là gì?*

Theo vết đối tượng (Object Tracking) là bài toán thuộc lĩnh vực thị giác máy tính.

Object tracking là bài toán theo dõi một hoặc nhiều đối tượng chuyển động theo thời gian thực trong một video. Hiểu một cách đơn giản, nó là bài toán cao hơn nhận dạng vật thể, khi đối tượng cần được xử lý không đơn giản là một hình ảnh mà là một chuỗi các hình ảnh, video.



Giới thiệu:

Nhận diện đối tượng thường được sử dụng để làm gì và ứng dụng như thế nào?

Đây là một trong những bài toán quan trọng và phổ biến trong lĩnh vực Machine Vision. Object tracking được ứng dụng rất nhiều và đem lại hiệu quả cao trong các lĩnh vực trong cuộc sống của chúng ta.

Trong kiểm soát an ninh: VD: ỨNG DỤNG KỸ THUẬT THEO DÕI ĐỐI TƯỢNG CHO BÀI TOÁN NHẬN DẠNG HÀNH VI CỦA KHÁCH HÀNG TRONG SIÊU THỊ .

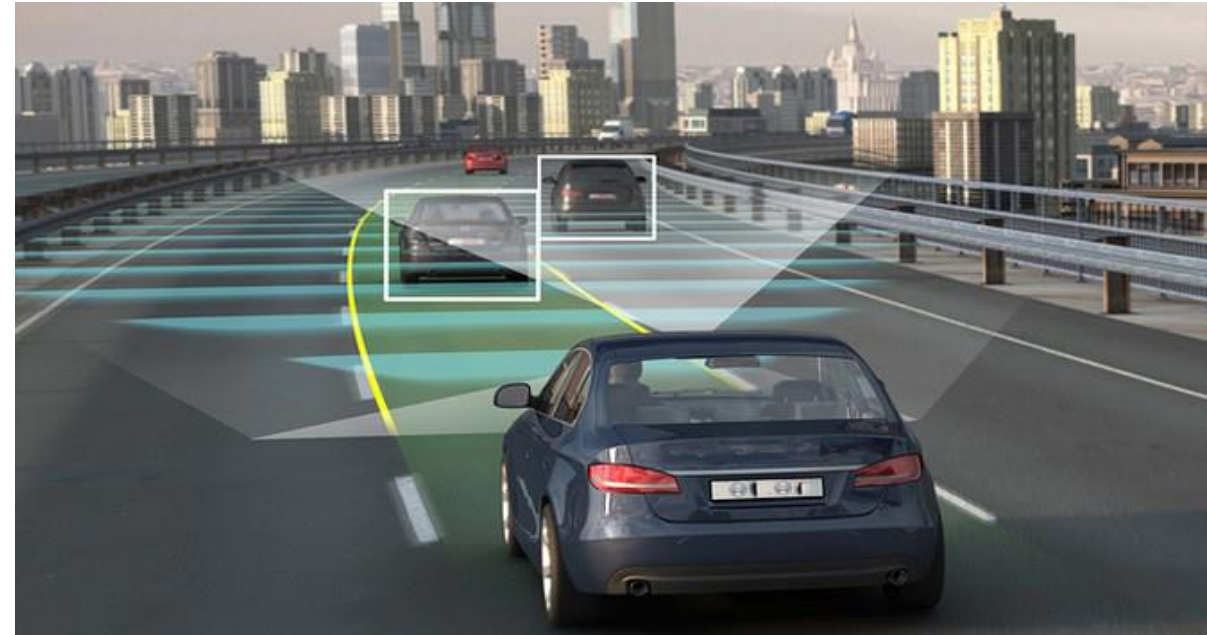
Hệ thống theo dõi đối tượng nhận vào các khung hình video thu nhận từ các camera, qua một số bước xử lý, phân tích và cuối cùng là đưa ra quỹ đạo đường đi của đối tượng theo thời gian làm cơ sở cho việc tiếp theo là nhận biết hành vi. Điều này giúp ta tiết kiệm rất nhiều thời gian và dễ dàng quản lý an ninh những khu đông người



Giới thiệu:

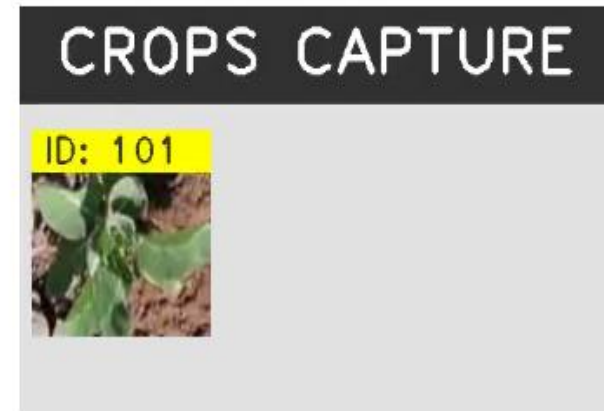
Nhận diện đối tượng thường được sử dụng để làm gì và ứng dụng như thế nào?

Trong xe tự hành: với xe tự hành , hệ thống điều hướng (navigation) cần phải nhận biết được chướng ngại vật (obstacle) trên đường đi. Và nếu đó là những đối tượng di chuyển, robot cần kích hoạt hệ thống theo vết thời gian thực để tránh va chạm.



Tracking from Drone Footage

Trong nông nghiệp: Sử dụng drone để quét và đếm nhanh Số lượng cây trồng, và gán ID cho từng cây và lưu trữ lại giúp ta dễ dàng quản lý.



8 OpenCV Object Tracking Implementations

You might be surprised to know that OpenCV includes eight (yes, *eight!*) separate object tracking implementations that you can use in your own computer vision applications.

I've included a brief highlight of each object tracker below:

- 1 **BOOSTING Tracker:** Based on the same algorithm used to power the machine learning behind Haar cascades (AdaBoost), but like Haar cascades, is over a decade old. This tracker is slow and doesn't work very well. Interesting only for legacy reasons and comparing other algorithms. (*minimum OpenCV 3.0.0*)
- 2 **MIL Tracker:** Better accuracy than BOOSTING tracker but does a poor job of reporting failure. (*minimum OpenCV 3.0.0*)
- 3 **KCF Tracker:** Kernelized Correlation Filters. Faster than BOOSTING and MIL. Similar to MIL and KCF, does not handle full occlusion well. (*minimum OpenCV 3.1.0*)
- 4 **CSRT Tracker:** Discriminative Correlation Filter (with Channel and Spatial Reliability). Tends to be more accurate than KCF but slightly slower. (*minimum OpenCV 3.4.2*)
- 5 **MedianFlow Tracker:** Does a nice job reporting failures; however, if there is too large of a jump in motion, such as fast moving objects, or objects that change quickly in their appearance, the model will fail. (*minimum OpenCV 3.0.0*)
- 6 **TLD Tracker:** I'm not sure if there is a problem with the OpenCV implementation of the TLD tracker or the actual algorithm itself, but the TLD tracker was incredibly prone to false-positives. I do not recommend using this OpenCV object tracker. (*minimum OpenCV 3.0.0*)
- 7 **MOSSE Tracker:** Very, very fast. Not as accurate as CSRT or KCF but a good choice if you need pure speed. (*minimum OpenCV 3.4.1*)
- 8 **GOTURN Tracker:** The only deep learning-based object detector included in OpenCV. It requires additional model files to run (will not be covered in this post). My initial experiments showed it was a bit of a pain to use even though it reportedly handles viewing changes well (my initial experiments didn't confirm this though). I'll try to cover it in a future post, but in the meantime, take a look at [Satva's writeup](#). (*minimum OpenCV 3.2.0*)

<https://pyimagesearch.com/2018/07/30/opencv-object-tracking/>

Các thuật toán thường sử dụng

MOSSE Tracker:

Tốc độ cao nhất, chính xác thấp nhất. Sử dụng trong các bài toán cần ưu tiên tốc độ.

KCF Tracker:

Tốc độ chậm hơn MOSSE và chính xác cao hơn MOSSE.

CSRT Tracker:

Tốc độ chậm nhất trong 3 món nhưng độ chính xác lại cao nhất trong 3 món.

OPTICAL FLOW:

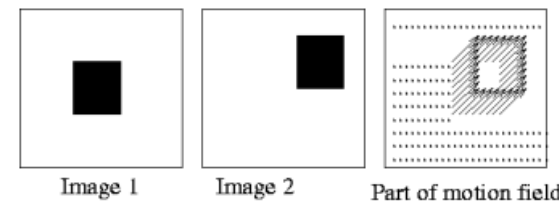
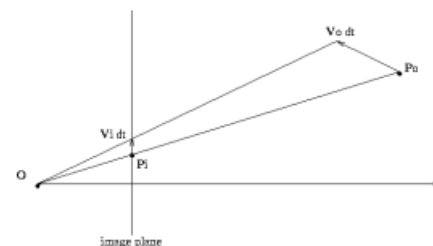
Đối tượng được theo dõi sử dụng các image brightness variations theo không gian-thời gian ở các cấp độ pixel. Trong thuật toán thì sẽ tập trung vào việc thu được vecto dịch chuyển cho các đối tượng qua các frame .

Optical flow theory

- Brightness consistency:** Độ sáng xung quanh một vùng nhỏ được cho là gần như không đổi, mặc dù vị trí của vùng có thể thay đổi.
 - Spatial coherence:** Các điểm lân cận trong cùng một khung cảnh thường sẽ cùng thuộc một bề mặt do đó sẽ có những sự chuyển động tương tự.
 - Temporal persistence:** Các điểm thường sẽ có sự chuyển động dần dần.
- Khi các tiêu chí trên được thoả mãn thì sẽ sử dụng phương pháp Lucas-Kanade để có được phương trình vận tốc của các điểm nhất định được theo dõi (thường ở đây là những điểm dễ phát hiện). Sử dụng phương trình cùng một số phương pháp dự đoán, thì một đối tượng sẽ được theo dõi trong toàn bộ video.

Motion Field

Motion field is an ideal representation of 3D motion as it is projected onto a camera image



Motion estimation



Intensity of a pixel in a image sequence is:

$$I(x, y, t)$$

Motion differential equation:

$$\frac{dI}{dt} = \frac{\partial I}{\partial x} \underbrace{\frac{dx}{dt}}_u + \frac{\partial I}{\partial y} \underbrace{\frac{dy}{dt}}_v + \frac{\partial I}{\partial t}$$

Unknown values

Optical flow theory

Motion assumption

1. Brightness constancy assumption

When an object moved slightly, its brightness is unchanged

$$I(x + u, y + v, t + 1) = I(x, y, t)$$

2. When u and v are small (< 1 pixel), using Taylor expansion we have:

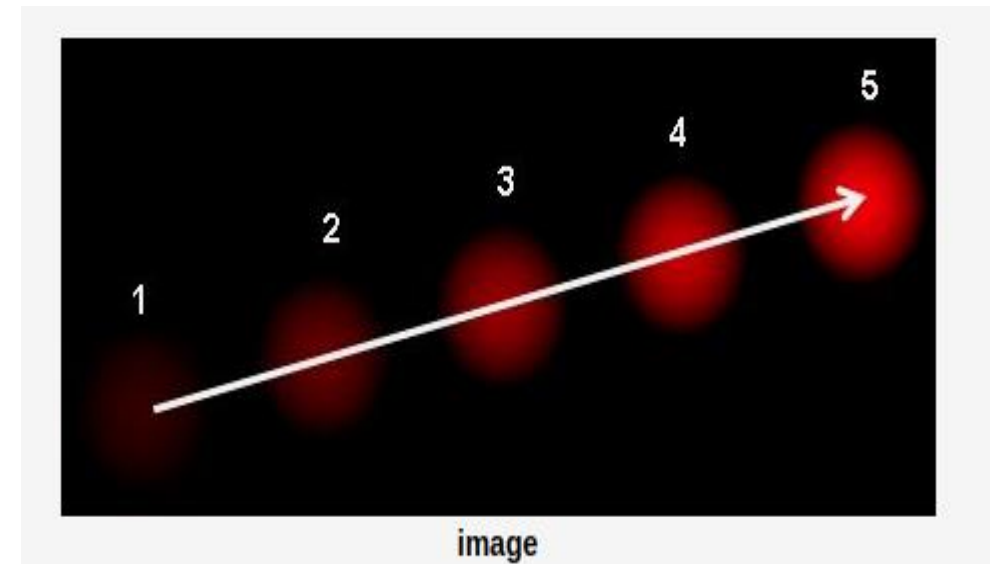
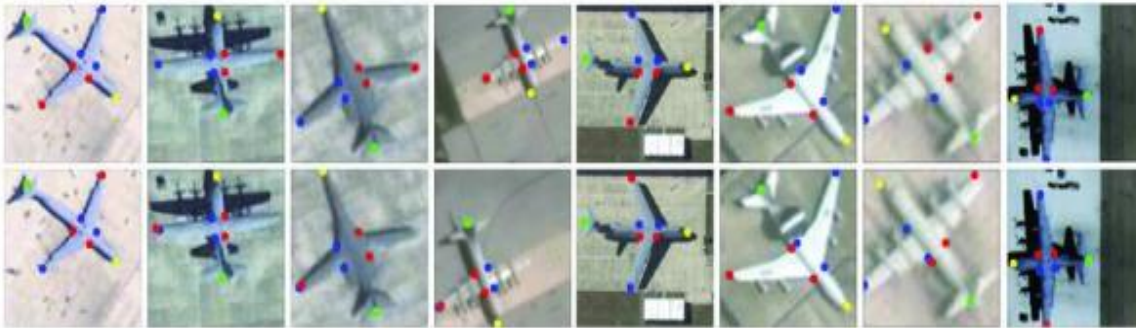
$$I(x + u, y + v, t + 1) = I(x, y, t) + I_x u + I_y v + I_t$$

$$\Rightarrow I_x u + I_y v + I_t = 0$$

Optical flow theory

Key points

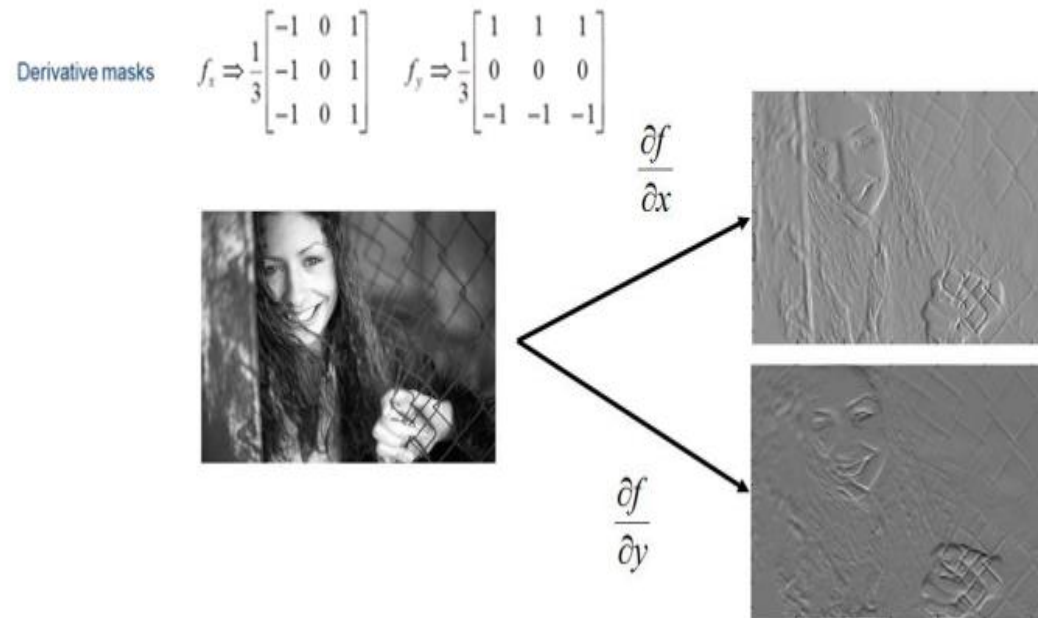
- Key points are groups of pixels that are repeatable and distinctive
- Key points should be invariant to image transformations:
 - Appearance variation (brightness, illumination)
 - Geometric variation (translation, rotation, scale)



Hình ảnh trên cho thấy sự chuyển động của quả bóng qua 5 frame liên tiếp

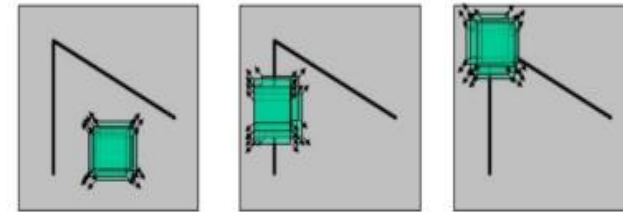
Optical flow theory

Image Derivative



Corners as key points

Harris algorithm



“flat” region:
no change in
all directions

“edge”:
no change along
the edge direction

“corner”:
significant change
in all directions

$$E(u, v) = \sum_{x, y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$$

Window
function

Shifted
intensity

Intensity



Original Image



Harris Corners