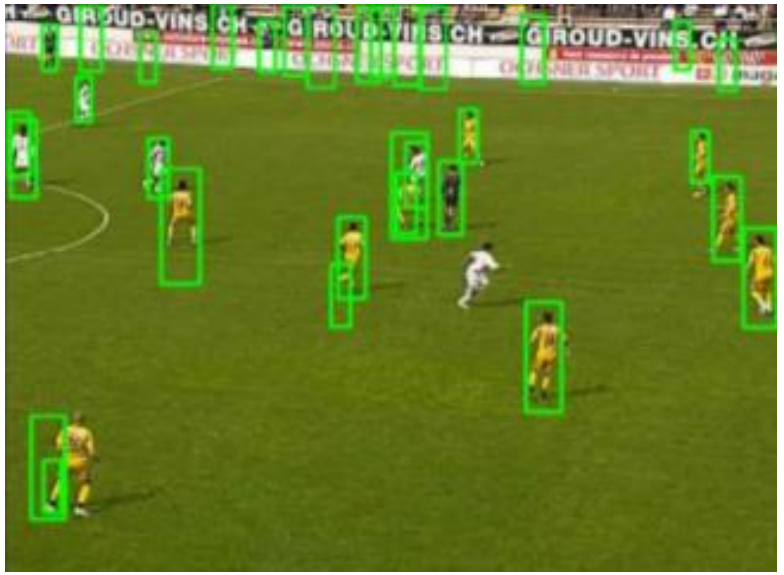
Отслеживание движения

Движение – смещение одних объектов относительно других.

В задачах компьютерного зрения выделяется несколько принципиально разных случаев движения:

1. Неподвижная камера, постоянный фон (*системы видеонаблюдения*).
2. Движущаяся камера, относительно постоянный фон (*создание панорамы из последовательности изображений*).
3. Движущаяся камера, постоянно изменяющийся фон (*системы автоматического управления автономными роботами*).

Задача видеонаблюдения - выделение «объектов интереса» в видео и отслеживание их движения в последующих кадрах.



1. Выделение объектов интереса:
    - Результат - ограничивающий прямоугольник
    - Или попиксельная маска
2. Отслеживание (video tracking)
    - Вход – положение объекта на первом кадре
    - Результат – траектория движения объектов (след или “track”)

Проблемы

Масштабируемость

Видео гораздо больше одного изображения, гораздо выше вычислительная нагрузка

Изменение по времени

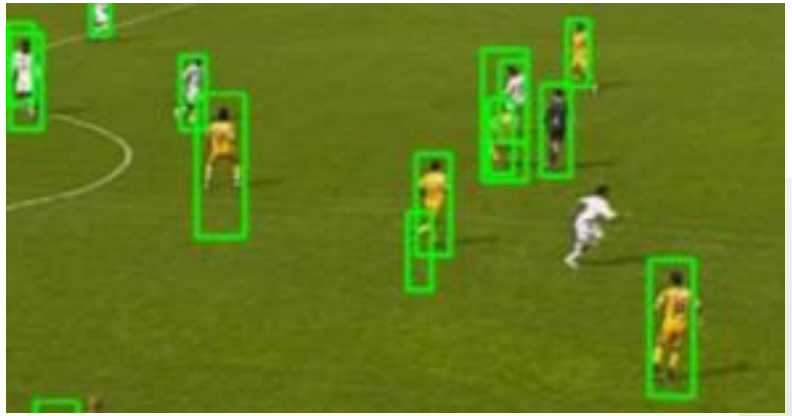
Вид объекта меняется от кадра к кадру из-за ракурса, изменения освещения, внутренних изменений (идущий человек)

Несколько объектов

В сцене могут быть несколько объектов, которые могут быть похожи друг на друга, перекрывать друг друга и т.д.

Выбор объекта интереса

1. Инициализация вручную



2. Детектор объектов

- Детектор «пешехода»
- Детектор «лиц»

3. Сегментация видео

- Выделение движущихся объектов

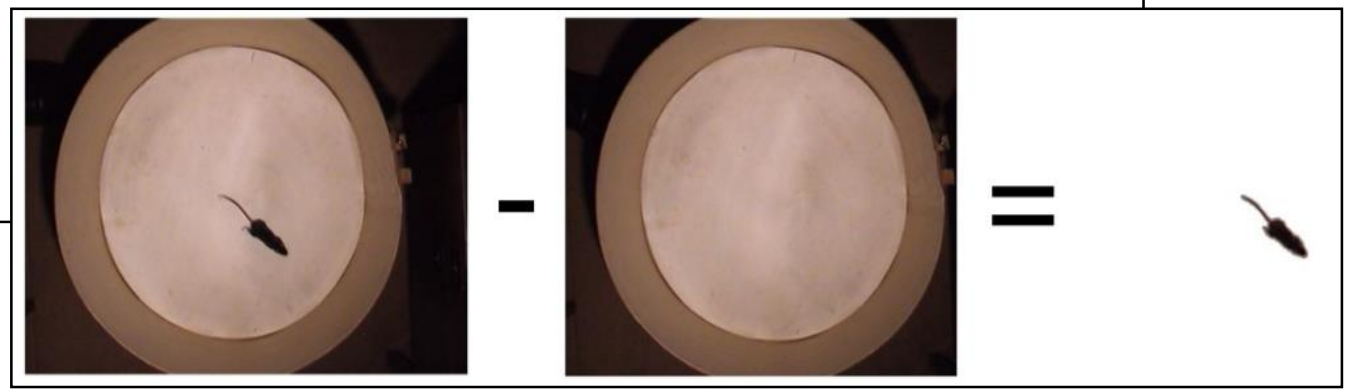
Система видеонаблюдения

Построим стандартную, простую систему видеонаблюдения

- Упрощение 1: стационарная камера
- Упрощение 2: стабильный фон

Вычитание фона (Background subtraction)

- Возьмем изображение без объектов (фон, background)
- Вычтем фон из новых изображений с объектами
- Сравним разницу для каждого пикселя с порогом – Порог – параметр алгоритма
- Если разница больше порога - то пиксель принадлежит «переднему плану» (foreground)
- Получаем маску «переднего плана»



Реальная картина

Яркость изменяется по времени

Наличие периодически изменяющихся объектов

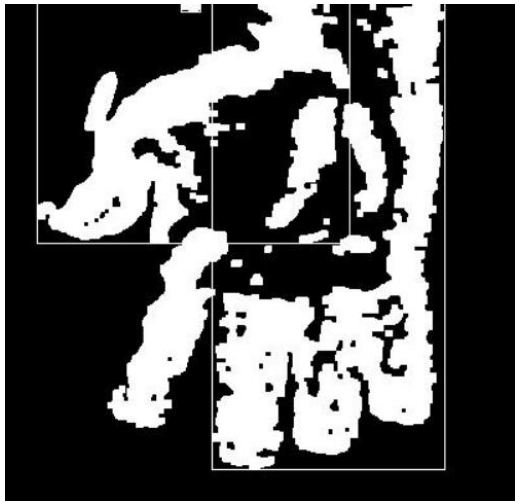
Возможные решения

1. По начальным кадрам строится модель яркости(цвета) пикселя фона. Если на новом кадре яркость (цвет) пикселя не удовлетворяет модели фона - значит это пиксель принадлежит движущемуся объекту.
2. Усреднение кадров. Возьмем N последних кадров и попиксельно усредним интенсивности. (*Однако усреднение не работает: в кадре всегда есть движущиеся объекты или случайные и резкие изменения яркости*)
3. Вместо усреднения берем медиану.
4. Смесь Гауссовых распределений.

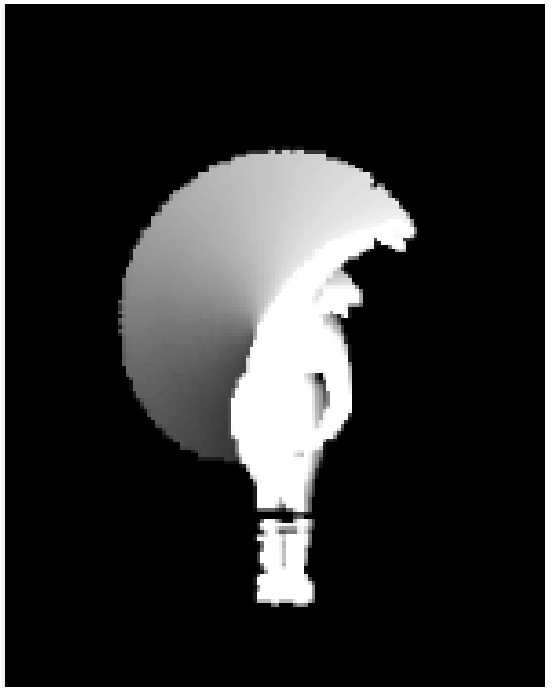
Интерпретация результатов

Метод вычитания соседних кадров

В общем случае на вход подаются два кадра одного размера, на выходе получаем маску активности, которая рассчитана на основе разности входных кадров, и представляет собой кадр с белыми и черными областями, где белые области означают зону движения.



Метод вычитания модели фона позволяет выделить весь движущийся объект, чтобы можно было судить о геометрических параметрах объекта.



motionHistoryImage в OpenCV

1. Сглаживаем текущий кадр фильтром Гаусса, чтобы избавиться от шумов.

2. Из сглаженного текущего кадра вычитаем предыдущий.

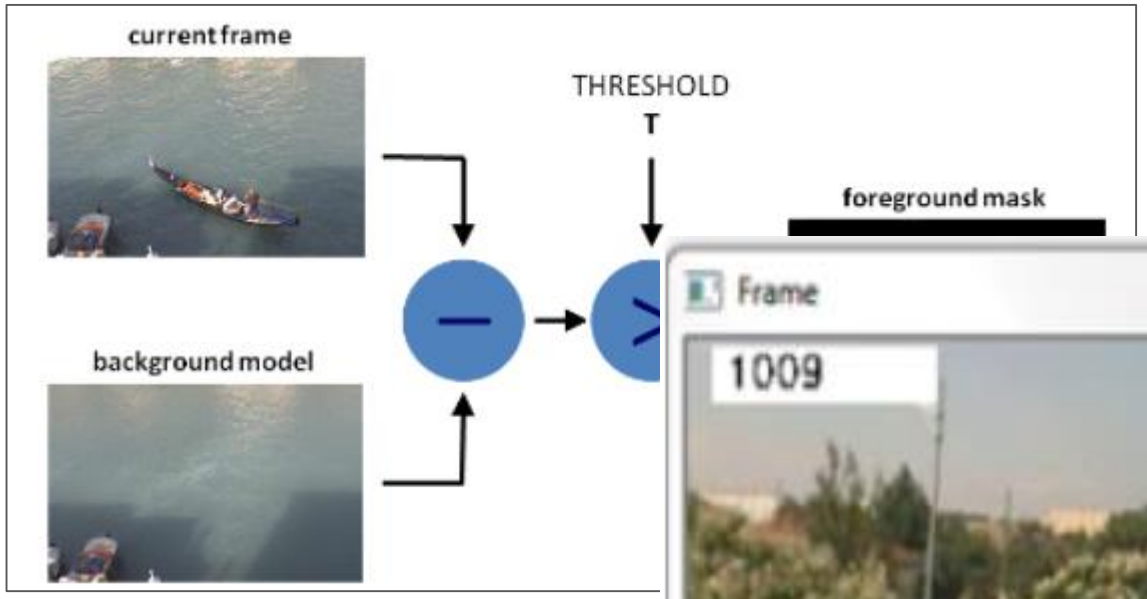
3. Сравниваем значения полученной разности с некоторым пороговым значением. Если значение пикселя больше порогового, то этот пиксель принадлежит движущемуся объекту, иначе отбрасываем его. Теперь мы получили бинарное изображение, где ноль означает, что пиксель не движется, отличное от нуля значение – пиксель движется.

4. Применяем морфологические операции закрытия и открытия, чтобы избавиться от движущихся регионов малого размера (шумы камеры). Полученное изображение и есть движущийся контур.

5. Наносим бинарное изображение на так называемое изображение истории движения (**motionHistoryImage**). На нём нарисованы движущиеся контуры за последние, например, 200 мс. **Контуры были получены через постоянные промежутки времени. Интенсивность пикселей контура обратно пропорциональна времени, которое прошло от измерения контура до данного момента.** Т.е. чем раньше был получен движущийся контур, тем он бледнее изображён на изображение истории движения.

http://robot-develop.org/archives/2328

В OpenCV реализован класс **BackgroundSubtractor**



Алгоритм

Результат работы

https://docs.opencv.org/3.4/d1/dc5/tutorial_background_subtraction.html

Многоуровневое движение

Во многих случаях визуальное движение вызвано смещением небольшого количества объектов, находящихся на разной глубине изображения. Поэтому движение пикселей можно описать более эффективно, если сгруппировать их в слои, и отслеживать многоуровневое движение (layered motion) построенных слоев.

Что такое отслеживание объектов?

Отслеживание – это поиск объекта в последовательных кадрах видео.

Подходы к отслеживанию объектов

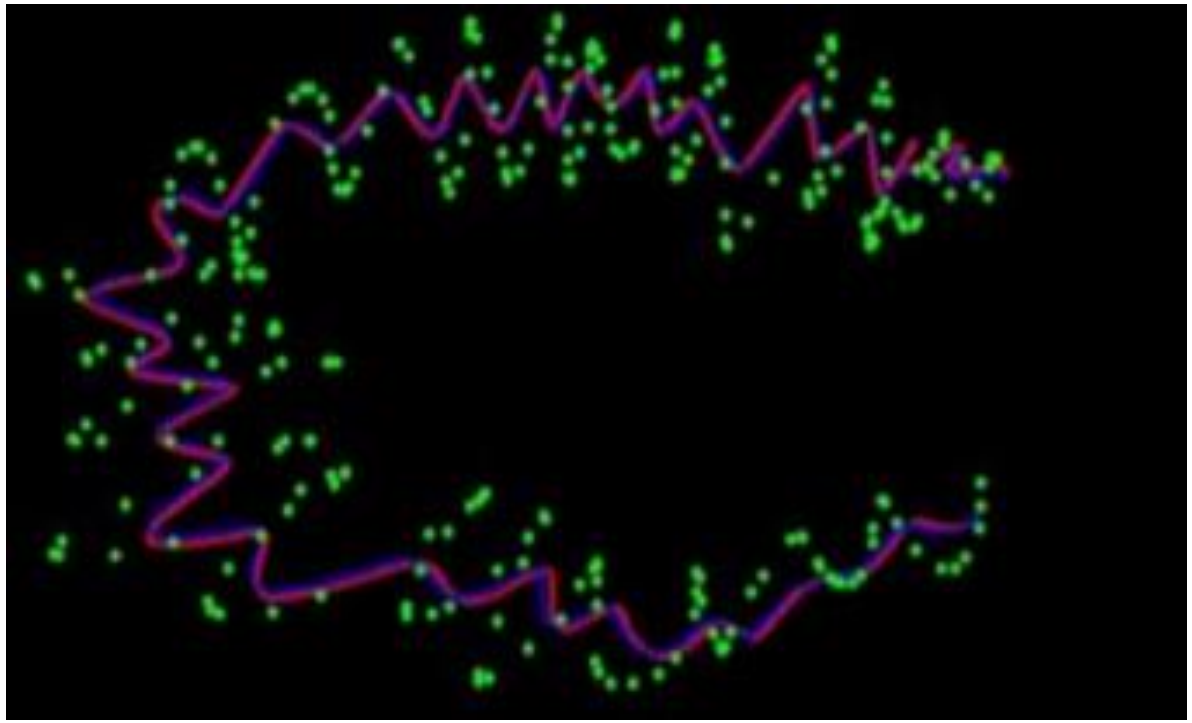
1. Плотный оптический поток: эти алгоритмы помогают оценить вектор движения каждого пикселя в видеокадре.

2. Редкий оптический поток: эти алгоритмы, такие как Tracker функции **Kanade-Lucas-Tomashi (KLT),** отслеживают расположение нескольких точек в изображении.

3. **Kalman Filtering** : очень популярный алгоритм обработки сигналов, используемый для **прогнозирования местоположения движущегося объекта на основе предшествующей информации о движении.**

Одним из ранних применений этого алгоритма было упоминание: «*бортовой компьютер, который управлял спуском лунного модуля Аполлона-11 на Луну, имел фильтр Калмана*».

Алгоритм метода состоит из двух повторяющихся фаз: **предсказание** и **корректировка**.



4. **Meanshift** и **Camshift** : это алгоритмы для определения максимумов функции плотности (центров масс).

На каждом кадре определяеися центр масс фигуры и ищется ближайший центр масс объекта со следующего кадра, такой метод может подойти, когда объекты движутся примерно с одной скоростью, их траектории не пересекаются, а камера имеет достаточное количество FPS, для того чтобы смещение быстро движущегося объекта на соседних кадрах было минимальным.

Например, такой способ может использоваться при сопровождении движения автомобилей на дороге, так как расстояние между центрами масс соседних автомобилей больше, чем смещение центра масс каждого на последовательности двух кадров.

https://docs.opencv.org/master/d7/d00/tutorial_meanshift.html

В чем отличие между отслеживанием или обнаружением

Почему отслеживание быстрее, чем обнаружение

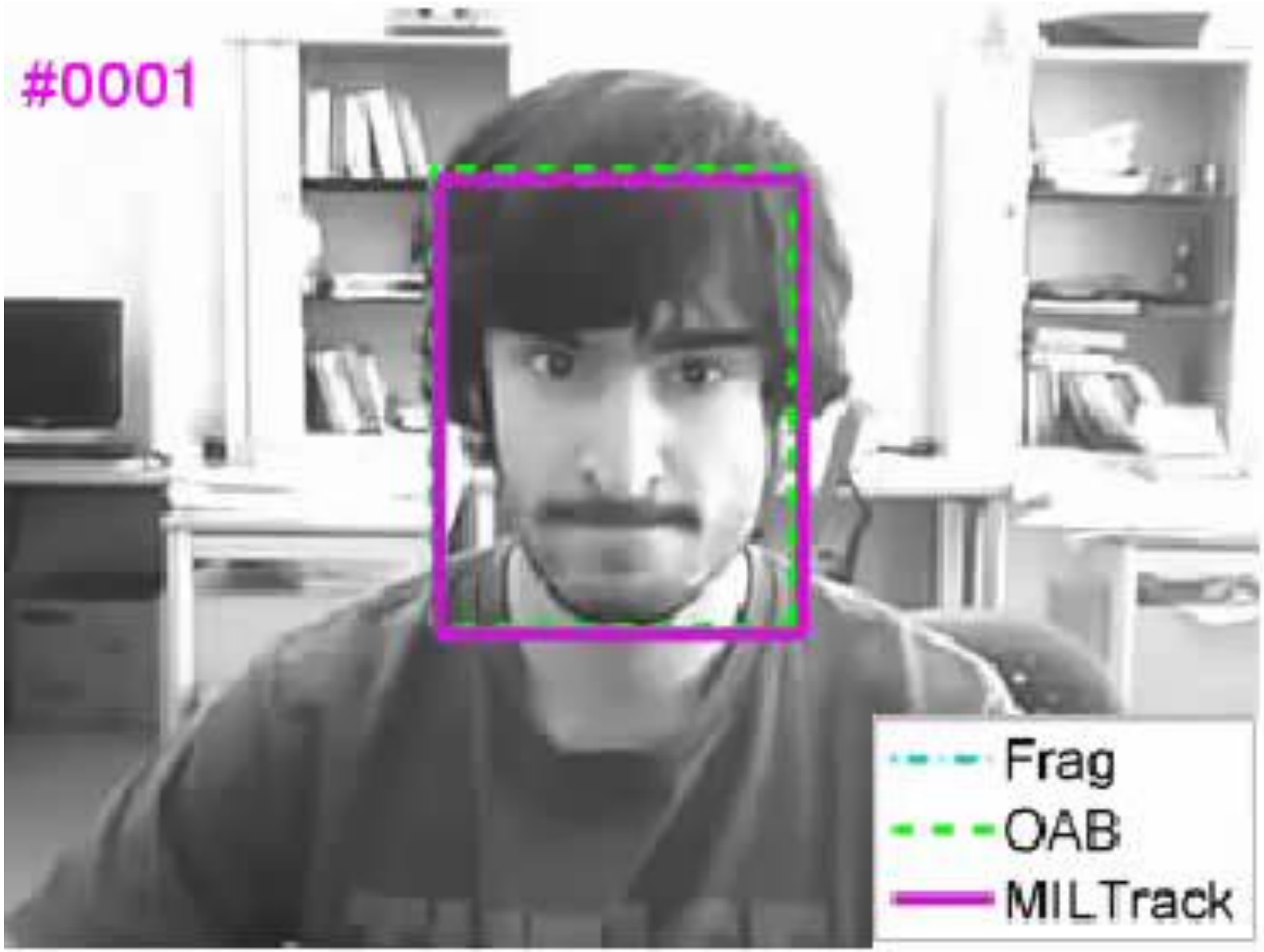
Хороший алгоритм отслеживания будет использовать всю информацию, имеющуюся у объекта до этой точки, в то время как алгоритм обнаружения всегда начинается с нуля.

НО: Постоянное слежение накапливает ошибки и рамка, определяющая объект, смещается. Чтобы избежать этого, через определенное количество кадров нужно снова выполнить обнаружение.

Отслеживание может помочь при сбое обнаружения

Если используется детектор лица на видео, а лицо закрывается объектом, детектор лица скорее всего перестанет определять лицо. Хороший алгоритм отслеживания, с другой стороны, будет обрабатывать некоторый уровень окклюзии.

Окклюзия - ситуация, в которой два объекта расположены приблизительно на одной линии и один объект, расположенный ближе к виртуальной камере, частично или полностью закрывает видимость другого объекта



Источник: https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/

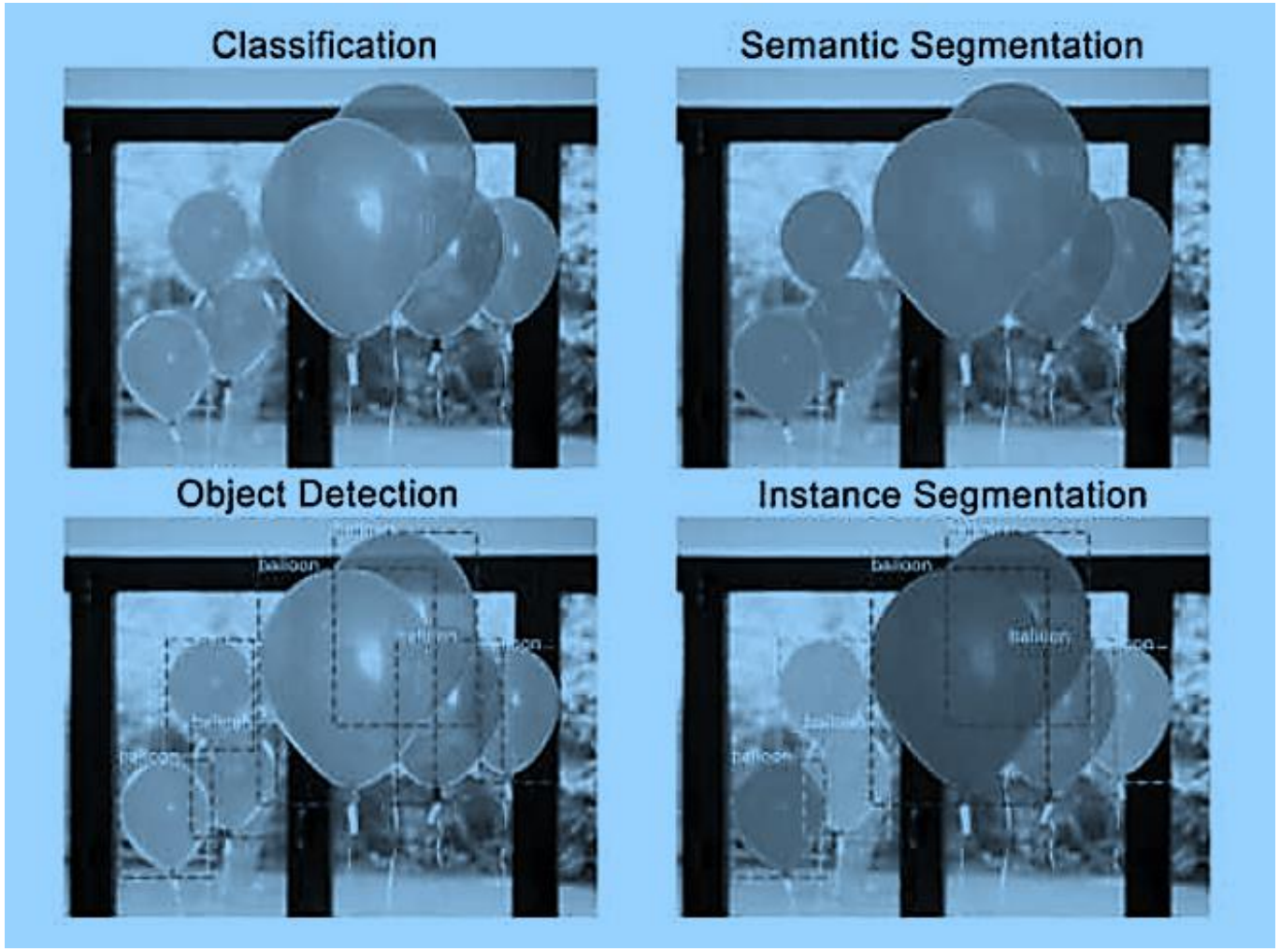
OpenCV 3.4 есть 7 различных трекеров

1. **BOOTTING** Это отслеживание объектов в режиме реального времени на основе алгоритма AdaBoost.
2. **MIL** генерирует классификатор в онлайн-режиме, чтобы отделить объект от фона.
3. **KCF** - это новая система отслеживания, которая использует свойства циркулянтной матрицы для повышения скорости обработки
4. **MEDIANFLOW** подходит для очень плавных и предсказуемых движений, когда объект виден на протяжении всей последовательности.
5. **TLD** отслеживание, обучение и обнаружение. Трекер следует за объектом от кадра к кадру. Детектор локализует все видимые явления, которые наблюдались до сих пор, и при необходимости корректирует трекер. Изучение оценивает ошибки детекторов и обновляет их, чтобы избежать этих ошибок в будущем.
6. **MOSSE** Отслеживание визуальных объектов с использованием адаптивных корреляционных фильтров
7. **GOTURN** является своего рода трекерами на основе сверточных нейронных сетей (CNN). Текущий метод GOTURN не обрабатывает окклюзии; однако он довольно устойчив к изменениям зрения, изменениям освещения и деформациям.

https://ehsangazar.com/object-tracking-with-opencv-fd18ccdd7369

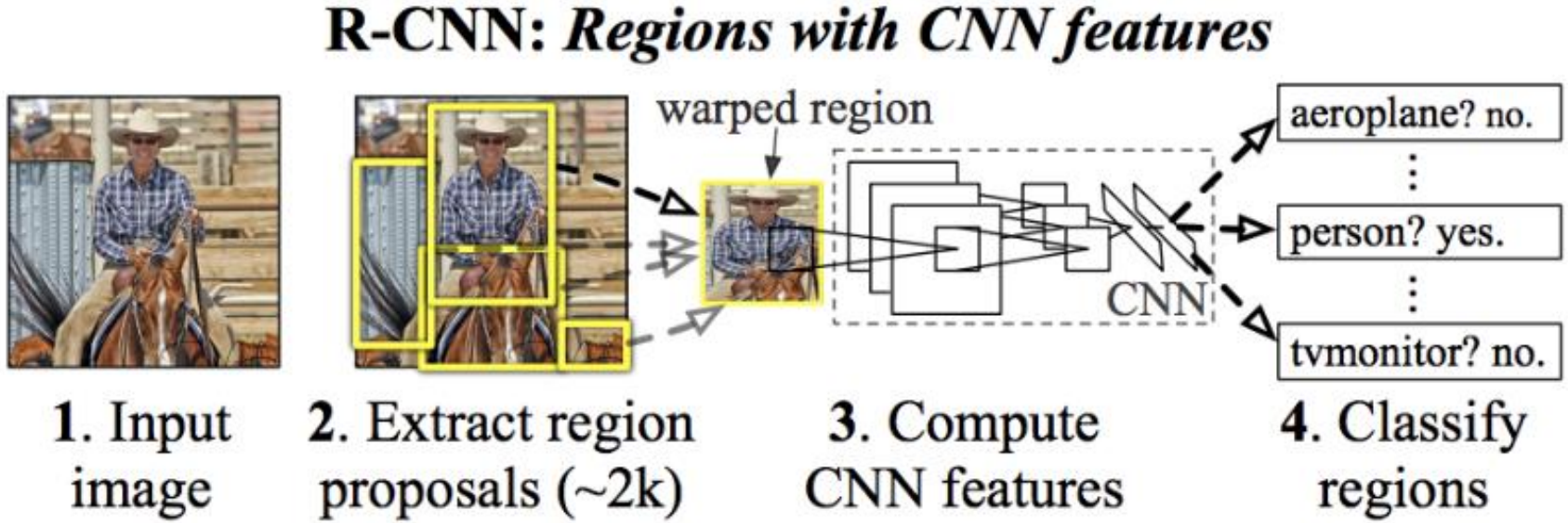
Object Tracking using OpenCV

Satya Mallick
LearnOpenCV.com



Современные задачи компьютерного зрения можно разделить на четыре вида:

- **Classification** — классификация изображения по типу объекта, которое оно содержит;
- **Semantic segmentation** — определение всех пикселей объектов определённого класса или фона на изображении. Если несколько объектов одного класса перекрываются, их пиксели никак не отделяются друг от друга;
- **Object detection** — обнаружение всех объектов указанных классов и определение охватывающей рамки для каждого из них;
- **Instance segmentation** — определение пикселей, принадлежащих каждому объекту каждого класса по отдельности;



R-CNN

2000 регионов
47 сек

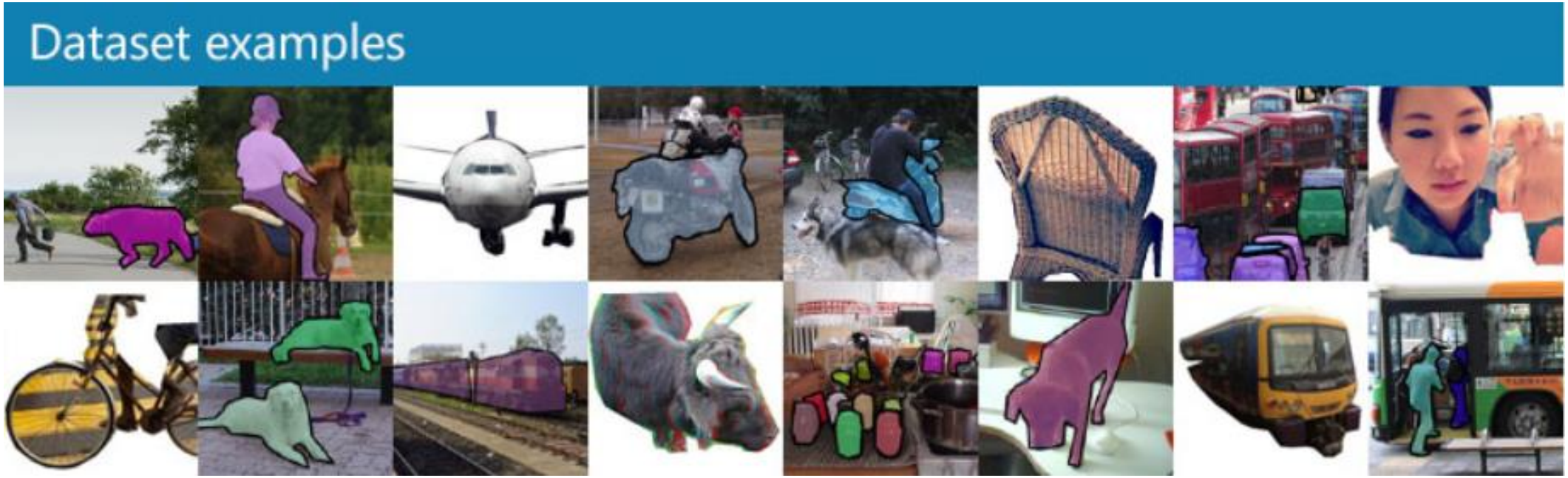
Fast R-CNN
Faster R-CNN
Mask R-CNN

Mask R-CNN

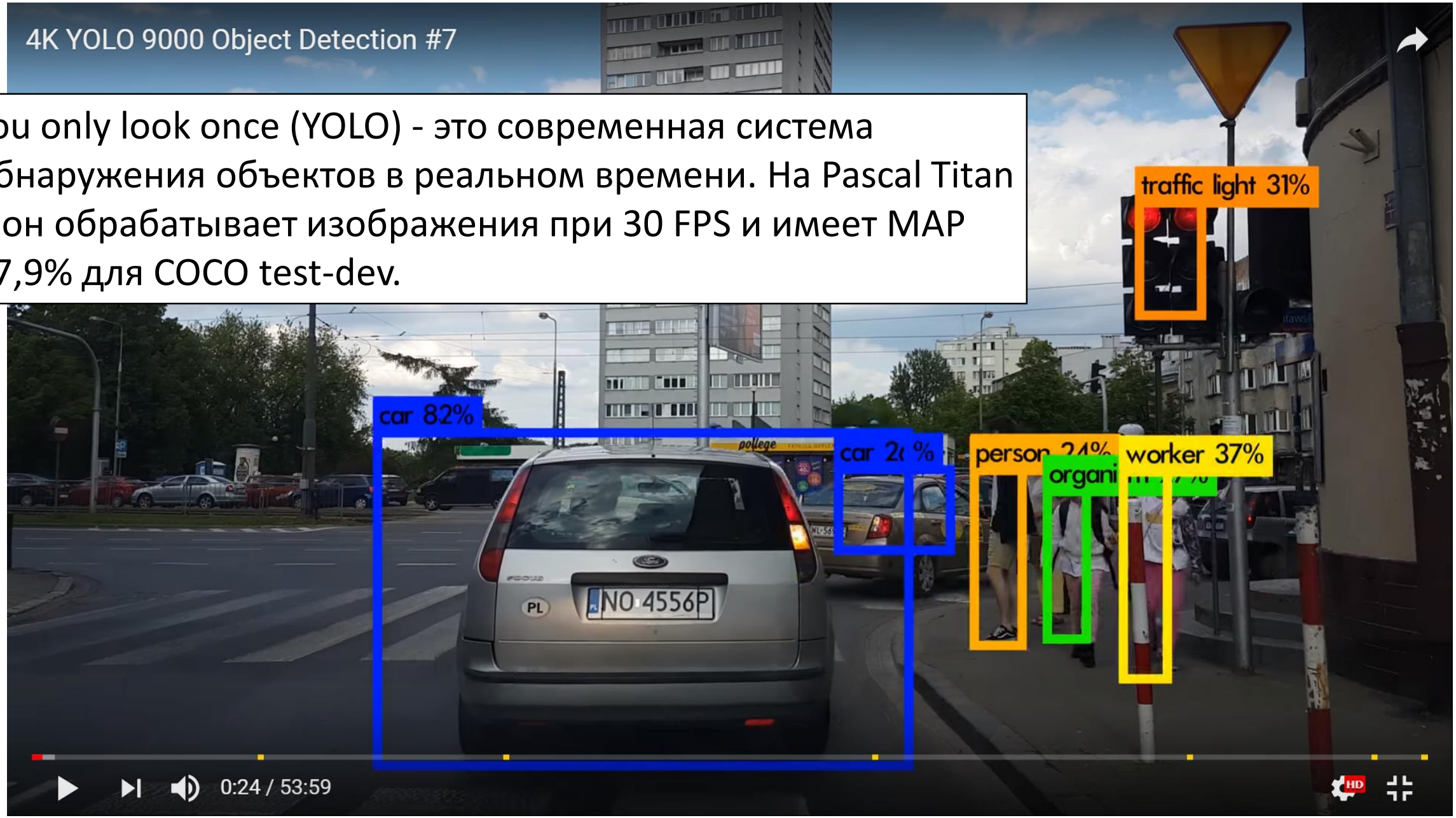
Mask R-CNN в определении поз по опорным точкам

Оценка точности методов сегментации

mAP (mean Average Precision) for Object Detection (это метрика для измерения точности детекторов объектов, таких как Faster R-CNN, SSD и т. д.)



https://pjreddie.com/darknet/yolo/



You only look once (YOLO) - это современная система обнаружения объектов в реальном времени. На Pascal Titan X он обрабатывает изображения при 30 FPS и имеет MAP 57,9% для COCO test-dev.

Tensorflow Object Detection