

ДЕТЕКТОР СТАЦИОНАРНЫХ ОБЪЕКТОВ

Киреев Даниил Сергеевич¹

Конушин Вадим Сергеевич²

1: *Студент, факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия*

2: *Генеральный директор, ООО «Технологии видеоанализа», Москва, Россия*

E-mail: daniil.kireev@graphics.cs.msu.ru, vadim@tevian.ru

Под стационарным объектом понимается объект переднего плана, помещенный или изъятый из сцены (забытые сумки, припарковавшиеся автомобили, украденные предметы). Алгоритмы, позволяющие детектировать такие случаи, находят широкое практическое применение в разных областях: от мониторинга трафика до борьбы с терроризмом. В данный момент существует не мало подобных алгоритмов, но они не лишены проблем. Большинство методов основаны на кадровом вычитании фона. Им свойственны некоторые недостатки (изменение освещения, перемещение объектов, камуфляж и другие) [2, 3]. В рамках данной работы был разработан алгоритм, который позволяет справиться с некоторыми из этих проблем.

Алгоритм наиболее похож на [1]. Основная идея того алгоритма в поддержании двух масок — похожести на текущий фон и стационарности. Мы внесли изменения в данный алгоритм: кадры видеопоследовательности берутся разрежено, что позволяет тратить больше времени на обработку каждого конкретного изображения; мы работаем не с цветом изображения, а с градиентом, чтобы уменьшить влияние изменяющегося освещения. Алгоритм для каждой точки изображения определяет, является ли эта точка частью стационарного объекта. Если точка изначально была статичной (находилась в модели фона), а через некоторое время изменилась и стала при этом через какой-то промежуток времени вновь статичной, то данная точка принадлежит стационарному объекту. Для того, чтобы находить такие точки, на каждой итерации (для каждого кадра) обновляются две аккумулирующие маски.

Первая маска — маска схожести двух соседних кадров. Она позволяет определить, является ли данная точка статичной. Изначально она заполнена нулями, а далее обновление происходит по следующему правилу: каждая точка текущего кадра сравнивается с каждой точкой предыдущего кадра; если они похожи, то для соответствующей точки в аккумулирующей маске мы прибавляем единицу, иначе сбрасываем это значение на ноль. В качестве критерия схожести

точек используется 3σ rule. Это необходимо для того, чтобы избавиться от влияния шумов. Кроме того, изначально инициализация фона производится с помощью этой маски: точка попадает в модель фона, если значение в этой маске достигает некоторого порога.

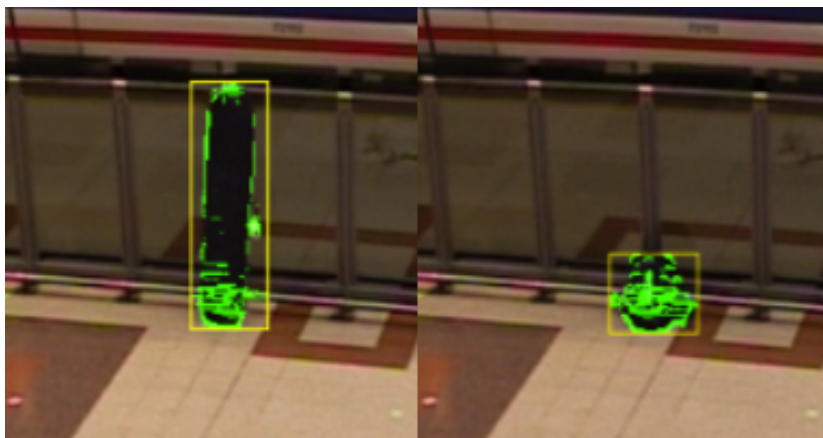
Вторая маска — маска различия текущего кадра и фона. Она позволяет определить, изменилась ли данная точка по сравнению с фоном. Изначально она так же заполнена нулями, а далее обновление происходит по следующему правилу: каждая точка текущего кадра сравнивается с каждой точкой модели фона; если они не похожи (опять же используется 3σ rule), то для соответствующей точки в аккумулирующей маске мы прибавляем единицу, иначе сбрасываем это значение на ноль.

В итоге по этим двум маскам для каждого кадра формируется маска точек, которые принадлежат стационарному объекту. В данной маске значение точки равно единице, если значения соответствующих точек в двух аккумулирующих масках достигли некоторого порога. Величина этого порога влияет на скорость реакции на событие. Точки в данной маске объединяются в объекты, которые мы и считаем стационарными (если они больше некоторого размера, размер искомым объектов — параметр алгоритма). Алгоритм объединения точек в объекты выглядит следующим образом: из пикселей, которые попали в маску стационарных точек, формируются связанные компоненты, которые расширяются по направлению своей оси инерции, в результате чего они объединяются в объекты.

После детекции алгоритм встраивает все пиксели стационарного объекта в модель фона и продолжает работу. Необходимо учесть, что различные пиксели одного и того же объекта могут быть обнаружены в разное время, соответственно необходима дополнительная задержка со вживлением. Если за то время, что мы ожидаем, размер найденного объекта остался меньше заданного порога, то мы вживляем его в фон, но не считаем данный объект стационарным.

Алгоритм тестировался на открытой базе PETS2007, а так же была собрана собственная база со сложными случаями сравнимая по размерам с открытой выборкой (поток людей, погодные эффекты, колышущиеся трава и деревья), на которой так же производилось тестирование. В качестве критериев качества использовались полнота и точность. На базе PETS2007 они обе показали 100% результат. На собственной базе результаты несколько ниже из-за наличия сложных примеров (полнота 80%, точность 20%).

Иллюстрации



Результат работы алгоритма (рамкой выделенные сами объекты, а внутри видны стационарные пиксели)

Литература

1. Diego Ortego, Juan C. SanMiguel. / Stationary foreground detection for video-surveillance based on foreground and motion history images. // *Advanced Video and Signal Based Surveillance (AVSS)*, 2013 10th IEEE International Conference. — 2013. — P. 75–80
2. YingLi Tian et al. / Robust Detection of Abandoned and Removed Objects in Complex Surveillance Videos. // *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. — 2010. — Vol. 41. — Issue 5. — P. 565–576
3. Quanfu Fan et al. / Robust Foreground and Abandonment Analysis for Large-Scale Abandoned Object Detection in Complex Surveillance Videos. // *Advanced Video and Signal-Based Surveillance (AVSS)*, 2012 IEEE Ninth International Conference. — 2012. — P. 58–63
4. Ying-Li Tian, M. Lu, A. Hampapur. / Robust and efficient foreground analysis for real-time video surveillance. // *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference*. — 2005. — Vol. 1. — P. 1182–1187