

УДК 004

Анализ методов распознавания динамических объектов

# 09, сентябрь 2012

Марченков А.М.

Научный руководитель к.т.н. Волосатова Т.М.  
МГТУ им. Н.Э. Баумана, Москва, Россия

МГТУ им. Н.Э. Баумана  
[bauman@bmstu.ru](mailto:bauman@bmstu.ru)

### Введение

Простейшим способом обнаружения объектов движущихся на постоянном фоне является вычитание изображений. Он вполне применим, например, когда ставится задача определить нахождения объектов на ленте производственного конвейера.

На рисунке 1 приведена схема алгоритма, с помощью которого на изображении можно выделить область, где произошли изменения.

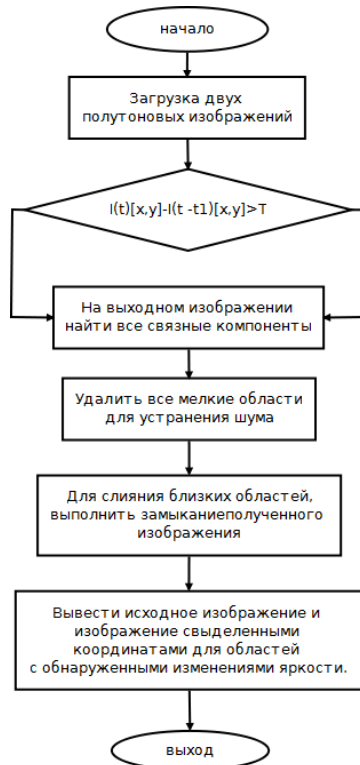


Рис. 1 - Алгоритм вычитания двух изображений

Важным параметром в описанном выше алгоритме является порог, которому определяется разница между двумя изображениями. Если порог взять минимальным, то есть равным 0, то возникнет множество паразитных пикселей, что иллюстрируется рисунком 3. При сравнении двух изображений с различными человеко различиями (рис. 2), при применении минимального порога результат остается сильно зашумленным и

содержит множество ложных пикселей. Но при применении большего значения количество ошибок резко уменьшается, в тоже время становится очевидным смещение объектов на последовательности кадров.

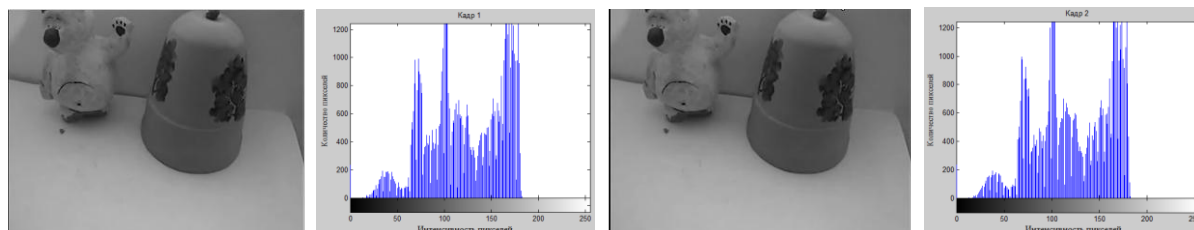


Рис. 2 - Сравнимые изображения с явными различиями и их

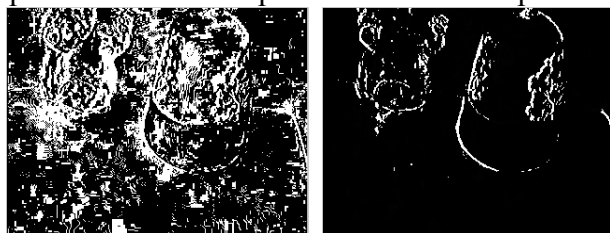


Рис. 3 - Гистограммы результаты вычитаний с различными значениями порога (первое  $T = 0$ , второе  $T = 5$ )

На рисунке 2 исходные изображения приведены вместе с их гистограммами.

### Вычисление векторов перемещения.

Поле движения называется двумерный массив двумерных векторов, представляющих движение точек трехмерной сцены. Вектора перемещения описывают смещения образов движущихся трехмерных точек. Вектор перемещения строится следующим образом: начало вектора – координаты образа трехмерной точки в момент времени  $t$ , конец – координаты образа той же точки в момент  $t + \Delta t$ .

Поле движения используется как для распознавания объектов, так и для анализа их движения. На значения интенсивности пикселей изображения при вычислении векторов перемещения обычно накладывается ограничение в одной из двух форм. Можно предположить, что интенсивность образа трехмерной точки  $P$  и некоторой ее окрестности в течение временного интервала  $(t_1, t_2)$ , на котором выполняется оценка перемещения точки, остается примерно постоянной. Или же, можно предположить, что на временном интервале  $(t_1, t_2)$  постоянна разность интенсивностей на образах краев объекта.

Поток изображения – это поле движения, вычисленное в предположении о постоянстве интенсивности изображений в окрестностях соответствующих точек.

Для вычисления разреженного поля движения можно пользоваться парами соответствующих точек, найденных на двух изображениях для различных моментов времени  $t_1$  и  $t_1 + \Delta t$ . Эти точки должны обладать некоторым характерным признаком, чтобы их можно было обнаружить на обоих изображениях. Например, и на полутоновых, и на цветных изображениях в качестве соответствующих точек можно выбрать угловые или граничные точки. На сегментированных полутоновых изображениях с движущимися областями постоянной формы в качестве соответствующих точек можно искать центры тяжести этих областей. Для угловых точек этот оператор вычисляет изменение яркости по вертикальным, горизонтальным и диагональным соседям пикселя  $P$ . Только, если минимальная из этих четырех разностей превышает некоторый порог, то пиксель  $P$  считается точкой интереса (признаковой точкой). Можно использовать и другие признаковые операторы, например, текстурный оператор.

После обнаружения интересных точек  $\{P_j\}$  на изображении  $I_1$  для момента времени  $t$  надо найти соответствующие точки изображения  $I_2$  для момента времени

$t+\delta$ . Вместо выполнения аналогичных вычислений и подбора соответствующих пар точек на изображении  $I_2$  проще искать новые положения точек интереса, найденных на  $I_1$ . Для этого можно пользоваться корреляционным методом. Для очередной интересующей точки  $P_j$  с изображения  $I_1$  нужно взять небольшую область вокруг этой точки из  $I_1$  и найти для нее положение с максимальной корреляцией на изображении  $I_2$ , предполагая, что возможное перемещение точки ограничено. Поиск точки нового положения  $P_j$  ограничен прямоугольной областью с размерами  $S \times R$ . С уменьшением размеров области ускоряется поиск и уменьшается вероятность ошибки из-за обнаружения неверной точки. Но уменьшать область имеет смысл, только если есть априорная информация о предельной скорости объектов [1]. Подробное описание развития алгоритмов слежения приведено в [3].

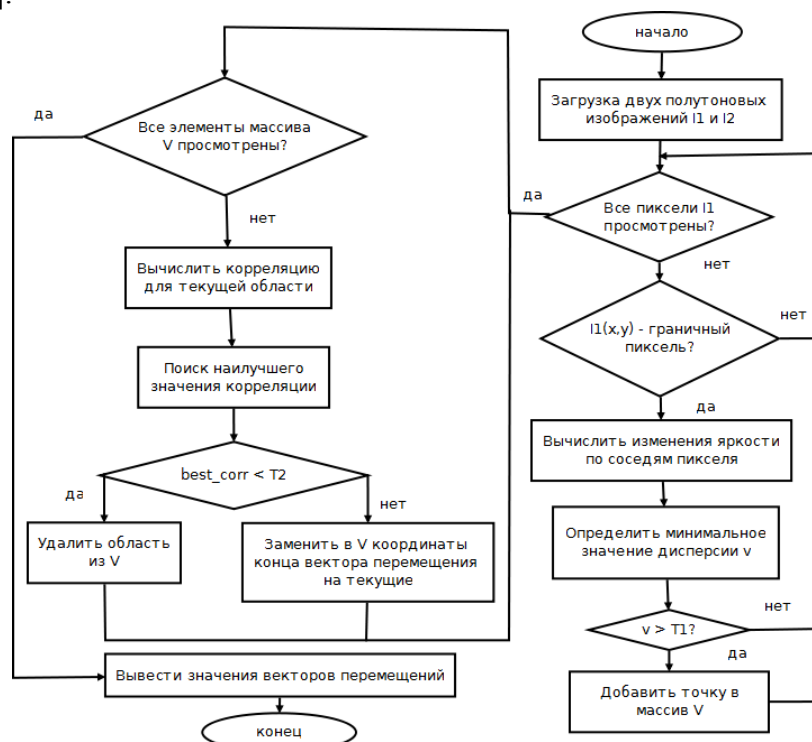


Рис. 4 - Алгоритм построения векторов перемещения  
( $T_1$  – яркостный порог для оператора интереса,  $T_2$  - порог значения  
кросс-корреляции по окрестности точки интереса)

На рис. 4 представлен алгоритм нахождения векторов перемещений, его можно применять для циклической обработки пар кадров для непрерывного прослеживания точек интереса на видеопоследовательности с большим количеством кадров. При этом угловые точки, обнаруженные на кадре  $t+\delta$ , можно после очередного применения алгоритма запомнить в качестве точек, обнаруженных на очередном кадре, но для вычисления кросс-корреляции лучше будет использовать новые, возможно, изменившиеся в окрестности. Подобным образом признаковые точки можно прослеживать на динамической сцене в условиях постепенного изменения окрестностей этих точек. В общем случае нужно предусмотреть действия алгоритма в случае появления новых или исчезновения старых угловых точек из-за перекрытия другими объектами [1].

Существует множество операторов для определения границ, на рисунке 5 приведены результаты их применения к одному из кадров видео ряда. Как видно из рисунка 5 наиболее точное определение границ объектов на исходном изображении дают 2 последних вида фильтров, но для решения поставленной задачи, нам не нужно определять все границы. Еще один недостаток этих фильтров в том, что происходит удвоение границы, что впоследствии увеличит число операций в алгоритме, при его

точность значительно не повысится, так будут рассматриваться окрестности смежных точек [2].

Фильтры Собеля и Превитта дают схожие результаты, поэтому в алгоритме можно использовать любой из них. Все дальнейшие действия будут производиться с изображением, полученным в результате применения фильтра Собеля.

На рисунке 6 приведены графики значений функции кросс-корреляции, вычисленные в области поиска для трех разных особых точек, выбранных произвольно.

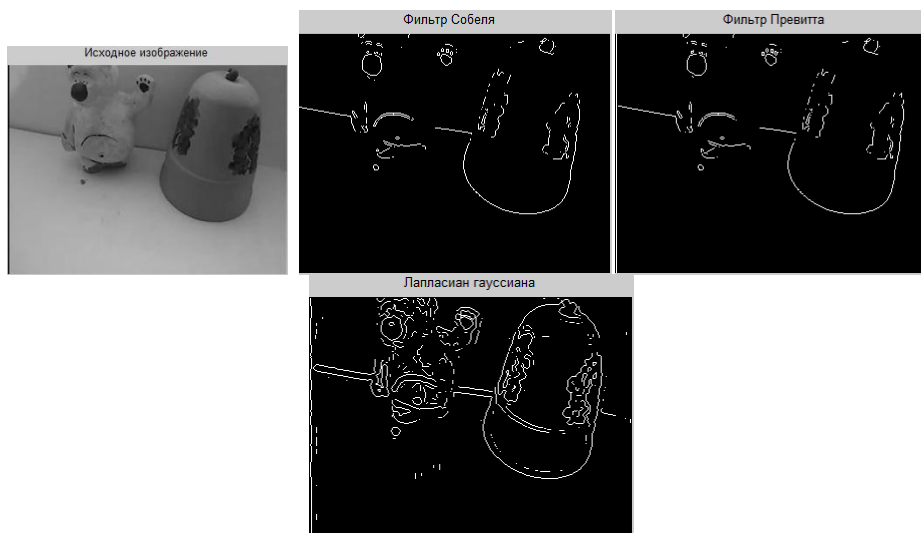


Рис. 5 - Результаты работы фильтров определения границ

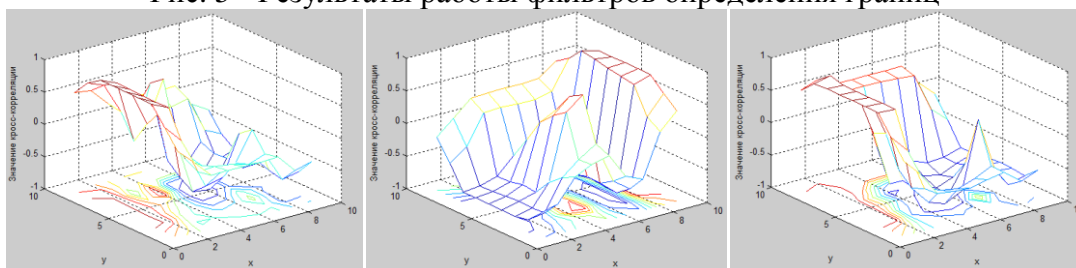


Рис. 6 - Графики значений функции кросс-корреляции для трех произвольных точек

Во многих прикладных задачах надежность и скорость слежения можно увеличить за счет дополнительной информации. Если в каждой из  $m$  точек определены некоторые характерные признаки, то к вычислению гладкости можно добавить сравнение и согласование признаков. На основе частичной траектории до текущего момента времени  $t$  можно предсказать местоположение следующей точки траектории на кадре для момента времени  $t+1$ . Это позволяет существенно уменьшить время корреляционного поиска очередной точки траектории [1].

Предложенный выше алгоритм работает на видеоряде, который состоит из изображений с незначительными смещениями особых точек. Это следует из начальных предположений. Решение задачи слежения в случае поворота или масштабирования изображения описывается в [4].

## Литература

1. Л. Шапиро, Дж. Стокман. Компьютерное зрение. Москва, «Бином. Лаборатория знаний», 752с., 2006 г.
2. Р. Гонсалес, Р. Вуде – Цифровая обработка изображений. : перевод с английского под редакцией П. Н. Чочиа – «Техносфера», Москва, 2005.

3. Антон Конушин. Слежение за точечными особенностями сцены (Point feature tracking). Компьютерная графика и мультимедиа. Выпуск №1(5)/2003. <http://cgm.computergraphics.ru/content/view/54>.
4. Виктор Гаганов. Инвариантные алгоритмы сопоставления точечных особенностей на изображениях. Компьютерная графика и мультимедиа. Выпуск №7(1)/2009. [http://cgm.computergraphics.ru/issues/issue17/invariant\\_features](http://cgm.computergraphics.ru/issues/issue17/invariant_features).