

Анализ влияния метода вычитания фона на конечную эффективность систем компьютерного зрения

проф., д.т.н. Я.Я. Петричкович, А.В. Хамухин.

Вычитание фона – это категория алгоритмов обработки изображений, предназначенных для классификации объектов на последовательности изображений как подвижные и неподвижные. Чаще всего вычитание фона используется в детекторах движения на основе видеокамер в системах видеонаблюдения. Существует множество методов вычитания фона, данной проблеме посвящено тысячи публикаций, среди которых особо можно выделить работы [2–6], описывающие основные типы методов. Алгоритмы вычитания фона отличаются как по вычислительной сложности, так и по точности идентификации, поэтому возникает потребность в сравнении данных методов. В работе [1] предложено сравнивать методы вычитания фона с помощью тестов на фиксированных последовательностях изображений разного вида с заранее отмеченными точками движения в форме маски для каждого кадра последовательности. В качестве метрики сравнения используются метрики совпадения результатов обработки алгоритмов, классифицирующих точки кадра как движущиеся или неподвижные, с заранее введёнными шаблонами, считающимися эталонами.

В современных видеосистемах вычитание фона – это лишь один из первых этапов обработки изображений, за которым на основе полученной маски движения следует дальнейший анализ происходящего в поле зрения видеокамер. Цель данной статьи – провести анализ того, насколько работа алгоритма вычитания фона влияет на достоверность конечного результата. Для этого мы, во-первых, опишем метрики, введённые в [1], во-вторых, приведём в качестве примера типичный алгоритм сопровождения движущихся объектов в поле зрения видеокамер, в-третьих, введём новые метрики для оценки качества работы алгоритмов сопровождения, и в-четвёртых, проведём испытания алгоритмов сопровождения объектов, в основе которых лежат разные методы вычитания фона и приведём анализ полученных результатов.

Принципы оценки методов вычитания фона changedetection.net

Тестовая выборка изображений changedetection.org состоит из 31 видеоролика, в которых содержится в общей сложности 90 000 кадров изображений. Видеоролики разделены на 6 различных категорий: базовый тестовый набор, изображения с динамически изменяющимся фоном (водоёмы, ветви кустарника и деревьев на ветре и т.п.), изображения с не закреплённых жёстко трясущихся камер, изображения с тенями, тесты на обработку изменений фона благодаря перемещаемым на изображениях предметам и изображения тепловизионных камер. Примеры изображений каждой из шести категорий приведены на рис. 1 (рисунок заимствован из работы [1]).



Рисунок 1. Примеры изображений тестовой выборки changedetection.net (сверху) с результатом обработки одним из распространённых методов вычитания фона (снизу). Слева направо примеры изображений из категорий: основной набор, динамический фон, тряска камеры, тени объектов, перемещение элементов фона, тепловизионные изображения.

Каждое изображение тестового набора размечается в каждой точке, то есть каждая точка может принадлежать одному из пяти классов: подвижная точка, неподвижная точка, состояние точки неопределено, точка находится вне области интереса, и точка неподвижна, но находится в тени от движущегося объекта. Если обозначить эти состояния, соответственно, как $C = \{m, s, u, o, h\}$, то разметкой является характеристическая функция $\chi_T(x, y)$, которая каждой точке изображения (x, y) сопоставляет её класс C . Пример разметки, заимствованный из [1] приведён на рис. 2. Состояние o (вне области интереса) введено для того, чтобы лучше разграничить категории тестовых наборов изображений. Состояние u (не определено) введено для устранения неоднозначности разметки в граничных точках изображений движущихся объектов: в граничных точках любому оператору, который проводит разметку, сложно однозначно отнести точку

к классу движущаяся(m) или неподвижная(s). Наконец, состояние h (тень подвижного объекта) специально введено для отдельной метрики, позволяющей оценить возможность метода вычитания фона классифицировать именно такие точки как неподвижные точки фона.

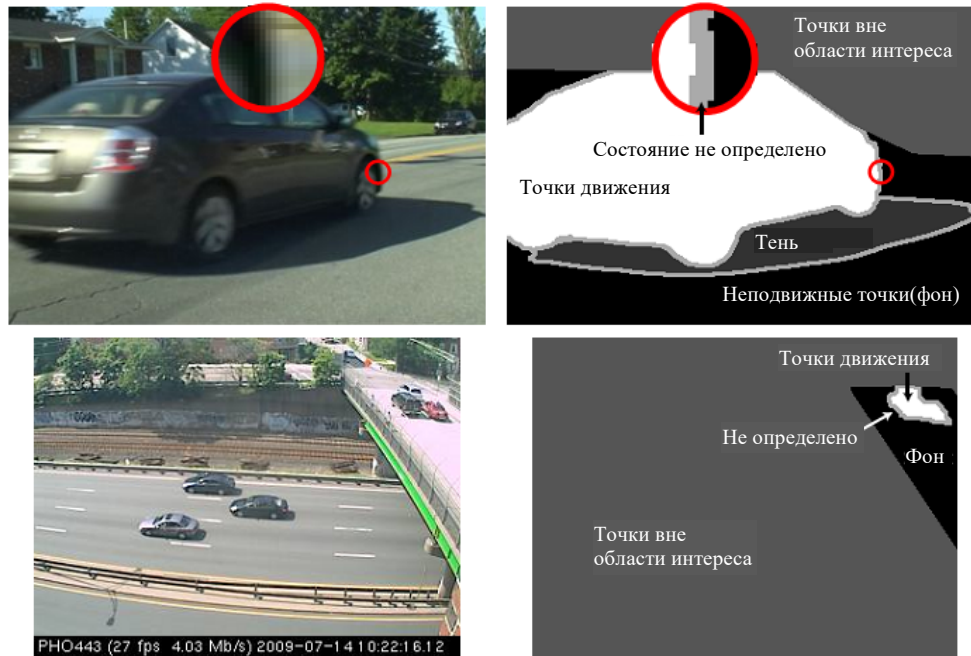


Рисунок 2. Пример разметки кадров из тестового набора.

Тестируемые алгоритмы вычитания фона выдают результат в форме индикаторной функции $\chi_A(x, y)$ для каждой точки каждого кадра, в отличие от функции χ_T , данная функция может принимать только два значения: m (точка движения) или s (неподвижная точка). Примеры данной функции для одного из наиболее простых алгоритмов вычитания фона приведены на рис. 1.

В рамках `changedetection.net` метрики качества работы алгоритмов вычитания фона строятся на основе функций χ_T и χ_A следующим образом. Множество точек, корректно определённых алгоритмом как движущиеся, определяется как

$$TP = \{(x, y) | \chi_T(x, y) = m \text{ при условии, что } \chi_A(x, y) = m\}.$$

Точки, корректно определённых алгоритмом, как неподвижные, обозначаются

$$TN = \{(x, y) | \chi_T(x, y) = s \text{ или } \chi_T(x, y) = h \text{ при условии, что } \chi_A(x, y) = s\}.$$

Множество точек, в которых алгоритм допустил ошибку первого рода, обозначается как

$$FP = \{(x, y) | \chi_T(x, y) = s \text{ или } \chi_T(x, y) = h \text{ при условии, что } \chi_A(x, y) = m\}.$$

Наконец, множество точек с ошибкой второго рода в результатах обозначается

$$FN = \{(x, y) | \chi_T(x, y) = t \text{ при условии, что } \chi_A(x, y) = s\}.$$

Чтобы избежать большого числа обозначений, далее под TP , TN , FP , FN подразумевается мощность одноимённых множеств. В рамках `changedetection.net` предлагается использовать следующие метрики для одной видеопоследовательности кадров в каждой из категорий.

1. $Re = TP / (TP + FN)$ – доля точек движения в шаблоне разметки, корректно определённых алгоритмом как движущиеся.

2. $Sp = TN / (TN + FP)$ – доля точек фона в шаблоне разметки, корректно определённых алгоритмом как неподвижные.

3. $FPR = FP / (FP + TN)$ – доля точек движения по результатам алгоритмов, ошибочно определённых движущимися точками.

4. $FNR = FN / (TN + FP)$ – доля точек фона по результатам алгоритмов, ошибочно определённых как неподвижные точки.

5. $PWC = (FN + FP) / (TP + FN + FP + TN)$ – доля ошибок классификации состояния движения точек в ответах алгоритмов вычитания фона.

6. $Pr = TP / (TP + FP)$ – доля точек движения по результатам алгоритмов, правильно определённых подвижными.

7. $Fm = 2 \frac{Pr \cdot Re}{Pr + Re}$ – среднее гармоническое показателей Re и Pr .

Показатели Re , Sp , FNR характеризуют ошибки первого рода, показатели FPR , Pr характеризуют ошибки второго рода, и показатели PWC и $Fmeasure$ тем или иным способом оценивают среднюю ошибку.

Для объединения этих метрик по всем видеороликам используются следующие правила. Обозначим значение одной из 7 метрик, приведённых выше, в категории c для видеоролика v как $M_{c,i}^v$ (расчёт данного показателя ведётся на основе величин TP , TN , FP , FN , на всех кадрах видеоролика, i – индекс метрики), а количество видеороликов в категории c как N_c . Значение метрик в рамках одной категории рассчитывается как среднее значение метрики по видеороликам:

$$M_{c,i} = \frac{1}{N_c} \sum_v M_{c,i}^v.$$

Значение показателя по всем категориям видеороликов вычисляется с помощью ещё одного усреднения

$$M_i = \frac{1}{6} \sum_c M_{c,i}. \quad (1)$$

Далее с помощью операции ранжирования вводится два различных обобщённых показателя качества работы сравниваемых алгоритмов вычитания фона. Операция ранжирования – это операция расстановки по порядку методов по одной из введённых ранее метрик в порядке убывания качества, рангом метода называется порядковый номер метода в данной расстановке. Если метрики у двух и более методов совпадают, то при ранжировании им всем назначается среднее арифметическое разделённых методами позиций. Если в сравнении участвует Ω методов, то для каждого -го метода из числа Ω можно определить его ранг по метрике $M_{c,i}$ в каждой категории последовательности изображений как $rank_{M_{c,i}}(m)$. Усреднённая позиция методов в каждой категории с учётом всех метрик рассчитывается по формуле

$$RM_{c,m} = \frac{1}{7} \sum_i rank_{M_{c,i}}(m),$$

а общий показатель метода m во всех 6 категориях рассчитывается как

$$RC_m = \frac{1}{6} \sum_c RM_{c,m}.$$

Альтернативной обобщённой метрикой сравнения методов вычитания фона служит средний ранг при упорядочивании методов по средним метрикам, рассчитанным по формуле (1):

$$R_m = \frac{1}{7} \sum_i rank_{M_i}(m).$$

В работе [1] сравнивается 16 различных методов, в таблице 1 приведены выборочные результаты, которые соответствуют, во-первых, наиболее извест-

ным методам вычитания фона, а во-вторых, наилучшим методам по результатам, указанным в рассматриваемой работе.

Метод	<i>RC</i>	<i>R</i>	<i>Re</i>	<i>Sp</i>	<i>FPR</i>	<i>FNR</i>	<i>PWC</i>	<i>Fm</i>	<i>Pr</i>
PBAS [5]	3.00	3.29	0.79	0.990	0.010	0.009	1.77	0.75	0.82
ViBe+ [3]	4.83	5.00	0.69	0.993	0.007	0.017	2.18	0.72	0.83
GMM Stauffer-Grimson [2]	11.50	10.14	0.71	0.986	0.014	0.027	3.68	0.52	0.72
Euclidean distance	16.67	14.00	0.70	0.969	0.031	0.017	4.35	0.61	0.62

Таблица 1. Результаты сравнения методов вычитания фона, приведённые в работе [1].

Алгоритм отслеживания движения объектов на основе метода вычитания фона

Информация о том, что в кадре есть движение, и что оно происходит в конкретных точках изображения, является слишком большой по объёму, чтобы представлять интерес с точки зрения систем так называемого семантического сжатия, или семантической обработки изображений. Обычно, эту информацию подвергают дальнейшей обработке, выделяя из полученных областей движения отдельные движущиеся объекты, одновременно происходит восстановление траектории движения объектов в кадре. Информация о масках движения на каждом кадре по объёму составляет сотни тысяч байт, информация о траекториях объектов составляет уже порядка тысячи байт, причём не на одном кадре, а на сотнях кадров. В описание траектории объекта может входить либо координаты ограничивающей изображение объекта прямоугольной рамки, либо координаты ограничивающего объект многоугольника на всех кадрах последовательности, на которых объект присутствовал.

Наиболее известный алгоритм выделения траекторий объектов на основе метода вычитания фона представлен в статье [7]. Алгоритм состоит из следующих процедур.

1. Фон и маска движения выделяются с помощью метода, похожего на представленный в работе [2]: фон моделируется функцией распределения с одним слагаемым в форме плотности распределения Гаусса, а не несколькими

слагаемыми, и дисперсия распределения оценивается с помощью скользящего среднего по модулю отклонения текущей интенсивности от текущей оценки средней интенсивности, а не с помощью скользящего среднего по квадрату отклонения. Обновление фона происходит только в движущихся точках, также, как и в работе [2].

2. Полученная маска движения подвергается сегментации по пространственному признаку: точки движения объединяются рекурсивной процедурой волновой закрашки, соседними считаются точки, у которых только одна координата изображения по модулю отличается на единицу. В результате процедуры закрашки из маски движения получаем множество 4х-связных областей движения.

3. Полученные области движения необходимо использовать для обновления информации о текущих объектах и их траекториях. Для этого строится двудольный граф из текущих объектов и найденных областей движения, области движения соединяются рёбрами с объектами, каждому ребру назначается вес. Вес ребра, соединяющего область и объект, зависит от близости области движения прогнозируемому положению объекта и от корреляции изображения объекта на предыдущих кадрах с изображением области движения на текущем кадре. Далее решается задача о назначении в двудольном графе с весами, в результате которой объектам сопоставляются те области движения, которые получены в результате отбора соответствующих рёбер в ходе решения задачи об оптимальном паросочетании. В работе [7] используется «жадный» алгоритм поиска паросочетаний, однако известно более эффективное решение [8]. Если область движения не назначена ни одному объекту, то она считается новым объектом, если объект уже давно не получал в пару область движения, то он считается пропавшим со сцены.

В результате трёх перечисленных шагов мы получаем траектории объектов движения. Возможны различные ошибки в работе данного алгоритма: могут определяться ложные движущиеся объекты в тех местах, где движения нет, могут быть пропущены объекты движения, возможны ошибки в решении задачи о

назначении, когда траектория одного объекта переходит в результатах алгоритма на траекторию другого объекта. В зависимости от того, какой метод вычитания фона применяется на первом этапе, данные ошибки будут различаться. Поскольку в данной работе мы оцениваем влияние вычитания фона на конечный результат в виде описания траекторий объектов, необходимо ввести определения, позволяющие вычислять метрики, характеризующие качество полученного результата.

Методика оценки качества алгоритмов выделения движущихся объектов и их траекторий

Алгоритмы выделения движущихся объектов принято проверять с помощью записанного видеоматериала, у которого есть так называемая аннотация или разметка. Аннотация заключается в том, что перед проверкой работы алгоритмов на некоторой видеопоследовательности каждый кадр вручную размечается. Разметка может содержать области движения, и их связь с областями движения на предыдущем кадре. Область движения обычно размечается с помощью минимального ограничивающего прямоугольника со сторонами, параллельными границам кадра, поскольку ввести такой прямоугольник достаточно просто с помощью современных устройств ввода. Раз есть связь с областями движения на предыдущих кадрах, с помощью разметки всех кадров можно восстановить траектории движения объектов во времени. Обладая подобной аннотацией для конкретной видеопоследовательности, можно сопоставить её с аннотацией, которые выдают различные алгоритмы при обработке этой видеопоследовательности. Разметка, или аннотация, сделанная вручную, является эталоном для работы алгоритмов. Поэтому мы будем называть её иногда «идеальной разметкой» или «идеальной аннотацией», имея в виду, что ожидается совпадение результатов работы алгоритмов с идеальной разметкой. Существуют системы аннотации видео в сети Интернет [9], ими можно свободно пользоваться в качестве web-приложений.

Рассмотрим основные показатели качества работы алгоритмов выделения движущихся объектов и их траекторий. Для этого введём обозначения:

– $V^n = \{V_i^n\}$ для последовательности кадров видео, где i – номер кадра, n – номер видеопоследовательности в тестовой базе;

– $\{I_l^n\}$ для объектов идеальной разметки, где l – номер идеального объекта, n – номер видеопоследовательности, на которой данный объект отмечен;

– $\{A_m^{an}\}$ для объектов автоматической разметки, где m – номер объекта, n – номер видеопоследовательности, a – номер тестируемого алгоритма;

– $P_{il}^n = P(V_i^n, I_l^n)$ и $P_{im}^{an} = P(V_i^n, A_m^{an})$ – функции, возвращающие положение объектов I_l^n и A_m^{an} на кадре V_i^n , где положение P_{il}^n и P_{im}^{an} – это множество точек объектов в кадре, которое, в связи с трудоёмкостью поточечной разметки изображений объектов, обычно представляется в виде прямоугольника наименьшего размера с параллельными границам кадра сторонами и содержащего все точки изображения объекта;

– $\rho(P_1, P_2)$ – мера совпадения двух областей кадра P_1 и P_2 , далее мы будем использовать $\rho(P_1, P_2) = \frac{|P_1 \cap P_2|}{|P_1 \cup P_2|}$ (отношение мощности пересечения множеств к мощности объединения, которое изменяется в пределах от 0 для непересекающихся множеств до 1 для совпадающих множеств);

– $\rho(P_{il}^n, P_{im}^{an}) = \rho(P_{im}^{an}, P_{il}^n) = \rho_{lm}^{ian}$ – мера совпадения положений автоматического объекта A_m^{an} и идеального объекта I_l^n на кадре i последовательности V^n ;

– $\mu(\omega)$ – это число элементов в конечном множестве ω , т.е. мощность;

– $\mu(A_m^{an}) = \mu(\{i | P_{im}^{an} \neq \emptyset\})$ – число кадров, на которых есть объект A_m^{an} ;

– $\mu(I_l^n) = \mu(\{i | P_{il}^n \neq \emptyset\})$ – число кадров, на которых есть объект I_l^n ;

Для сокращения обозначений далее будем рассматривать фиксированный алгоритм обработки и фиксированную видеопоследовательность, что позволит нам опустить индексы алгоритма a и видеопоследовательности n .

На основе введённых обозначений можно рассчитать следующие метрики работы алгоритмов.

Число ошибок первого рода можно рассчитать по формуле

$$F_P^{\alpha\beta} = \mu\left\{A_m \left| \forall l : \frac{\mu(\{i | \rho_{lm}^i < \alpha\})}{\mu(A_m)} > \beta \right.\right\}, \quad (2)$$

где $\alpha \in (0,1]$ – порог меры совпадения автоматического и идеального объекта на одном кадре для того, чтобы иметь основания эти объекты сопоставлять на данном кадре, а $\beta \in [0,1]$ – порог на долю кадров автоматических объектов, на которых идеальный объект должен быть сопоставлен автоматическому для того, чтобы автоматический объект не считался ложным срабатыванием. Чем ближе пороги α и β к единице, тем строже критерий для отбора объектов, не являющихся ложными, чем ближе эти пороги к 0, тем критерий мягче. Например, при $\beta = 0$ и достаточно малых α , например, $\alpha = \alpha_m = \frac{1}{2} \min_{i,l,m} \rho_{lm}^i$, автоматический объект будет считаться ложным только тогда, когда он ни разу не пересекался по положению ни с каким из идеальных объектов.

Число ошибок второго рода можно рассчитать по формуле, подобной (2):

$$F_N^{\alpha\beta} = \mu\left(I_l \left| \forall m : \frac{\mu(\{i | \rho_{lm}^i < \alpha\})}{\mu(I_l)} > \beta \right.\right),$$

т.е. идеальный объект считается пропущенным, если не нашлось ни одного автоматического объекта, который совпадал бы по положению лучше, чем на α , на более, чем на $(1 - \beta)\mu(I_l)$ кадрах. При $\beta = 0$ и $\alpha = \alpha_m$ идеальный объект считается обнаруженным, если хотя бы на одном кадре его положение пересекается с положением одного из автоматических объектов.

Качество сопровождения объекта можно оценить с помощью показателей

$$Q_T^\alpha = \sum_l Q_{Tl}^\alpha, \text{ где } Q_{Tl}^\alpha = \mu(\{i | \exists m: \rho_{lm}^i \geq \alpha\}),$$

$$E_T^\alpha = \sum_m E_{Tm}^\alpha, \text{ где } E_{Tm}^\alpha = \mu(I_l | \exists i: \rho_{lm}^i \geq \alpha),$$

где величина Q_{Tl}^α характеризует, насколько постоянно идеальный объект I_l сопровождается автоматическими объектами, а величина E_{Tm}^α показывает, сколько раз автоматический объект A_m ошибочно меняет сопровождаемые идеальные объекты. Для оценки качества выделения траекторий интересны интегральные соотношения $Q^\alpha = \frac{Q_T^\alpha}{\sum_l \mu(I_l)}$, $E^\alpha = \frac{E_T^\alpha}{\mu(\{A_m\})}$, которые, соответственно, показывают долю кадров, на которых идеальные объекты сопровождались автоматическими объектами, и число ошибок смены траектории по отношению к общему числу автоматических объектов.

Экспериментальная проверка качества алгоритмов сопровождения движущихся объектов

На сайте www.changedetection.net, кроме видеопоследовательностей, доступны все результаты обработки всех изображений методами вычитания фона, упомянутыми в работе [1], в форме маски. Эти результаты можно использовать для эмуляции работы алгоритма вычитания фона на первом этапе алгоритма выделения траекторий движущихся объектов, что позволит для всех методов вычитания фона получить автоматическую аннотацию, а затем метрики качества для всех категорий видеоизображений в виде

1. $F_p = F_p^{\alpha\beta}$ при $\alpha = 0.5$ и $\beta = 0.3$.
2. $F_n = F_p^{\alpha\beta}$ при $\alpha = 0.5$ и $\beta = 0.3$.
3. $Q = Q^\alpha$ при $\alpha = 0.5$.
4. $E = E^\alpha$ при $\alpha = 0.5$.

Данные метрики могут быть вычислены с помощью специального программного обеспечения, которое сопоставляет идеальную разметку с автоматической, полученной на основе алгоритмов сопровождения движущихся объектов. Используя эти метрики, можно рассчитать обобщённые показатели качества, применив тот же подход ранжирования, который производился при расчёте метрик RC_m и R_m для методов вычитания фона. Изменится только состав и количество исходных метрик. Вычисленные ранги методов конечного алгоритма слежения за движущимися объектами обозначим RTC_m и RT_m , соответственно. Итоговые результаты сравнения алгоритмов сопровождения движущихся объектов, в основе которых лежат разные алгоритмы вычитания фона, приведены в таблице 2. В таблице представлены все методы, рассмотренные в работе [1], в порядке убывания результатов по метрике RTC . Кроме того, отдельной строкой «Baseline» представлены модельные результаты, полученные для случая, когда в алгоритм сопровождения движущихся объектов подставляются шаблонные точки движения, полученные в результате ручной разметки видеоданных.

Метод	RTC	RT	RC	R	F_p	F_n	Q	E
Baseline	6,1875	4,25	-	-	11	80	0,35	0,31
GMM KaewTraKulPong	7,29	7,75	9,5	9,43	363	94	0,40	0,35
ViBe+	7,48	6	4,83	5	309	91	0,40	0,39
ViBe	7,58	7,75	9,33	10,17	296	102	0,39	0,35
SOBS-SC	8,5	6,5	6	6,14	246	98	0,31	0,26
Bayesian Back	8,77	5,625	11	12,57	268	109	0,40	0,27
KDE Nonaka	8,875	8,25	9,17	8,43	422	102	0,35	0,27
SOBS	9,15	7,75	8,17	8,57	295	108	0,32	0,261
KDE Elgamal	9,67	10,875	9,67	11,43	295	107	0,32	0,36
GMM Stauffer Grimson	9,99	9,5	11,5	10,14	341	110	0,36	0,29
PBAS	10,04	9,75	3	3,29	287	102	0,31	0,34
GMM Zivkovic	10,19	9,875	13,67	10,86	320	108	0,34	0,34
KDE Yoshinaga	10,94	9,75	10,67	9,29	417	113	0,39	0,28
Euclidian distance	11,06	13,125	16,67	14	351	109	0,34	0,42
PSP-MRF	11,17	12,25	4,83	5	189	122	0,30	0,35
GMM RECTGAUSS-TeX	12	13,75	13,67	13	125	165	0,20	0,44
Mahalonobis distance	12,13	15	15,5	13,43	483	116	0,33	0,36
Chebyshev probability	12,83	14,25	6,67	5,86	186	143	0,18	0,48
Local Self-similarity	16,19	18	14,67	13,14	427	191	0,10	0,42

Таблица 2. Характеристики алгоритмов сопровождения движения на основе различных методов вычитания фона.

Выводы

Качество алгоритмов вычитания фона напрямую влияет на конечное качество сопровождения отдельных объектов: это видно из того, что работа алгоритмов сопровождения на основе введённых вручную точек движения получила наивысшую оценку качества в таблице 2.

С другой стороны, отсутствует прямая корреляция между результатами по сравнению алгоритмов вычитания фона и результатами работы алгоритмов выделения траекторий объектов на их основе. Если не учитывать шаблонную разметку, то с точки зрения метрики RTC лучше всего работает алгоритм, являющийся средним по метрике R . За счёт дополнительной обработки можно получить хороший результат по алгоритму отслеживания объектов, используя не сложные методы вычитания фона.

Анализ таблицы 2 также показывает, что даже для «идеальной» разметки точек движения предложенный алгоритм трекинга объектов совершает довольно много ошибок.

Таким образом, необходимо комплексно подходить к проблеме автоматического выделения событий на изображении видеокамер. Недостаточно получить

хороший результат только для одной из вычислительных подсистем при идентификации событий в системах компьютерного зрения, поскольку существует взаимное влияние подсистем друг на друга при формировании конечного результата.

Список литературы

1. Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Kijncrad, Prakash Ishwar, “changedetection.net: A New Change Detection Benchmark Dataset”, IEEE CVPR 2012.
2. C. Stauffer and E. Grimson. “Learning patterns of activity using realtime tracking”, IEEE Trans. Pattern Anal. Machine Intell., 22(8):747–757, 2000.
3. O. Barnich and M. Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. IEEE Trans. Image Process., 20(6):1709–1724, 2011.
4. L. Maddalena and A. Petrosino. The SOBS algorithm: what are the limits? In IEEE Workshop on Change Detection, 2012.
5. M. Hofmann. Background segmentation with feedback: The pixelbased adaptive segmenter. In IEEE Workshop on Change Detection, 2012.
6. A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density for visual surveillance. Proc. IEEE, 90:1151–1163, 2002.
7. T. Kanade, R. Collins, A. Lipton, P. Anandan and P. Burt. Cooperative multi-sensor video surveillance. Proc. of the DARPA Image Understanding Workshop, May 1997, volume 1, pp. 3–10.

8. H. N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms, pages 434–443, 1990.

9. J. Yuen, B. C. Russell, C. Liu, and A. Torralba, "LabelMe video: building a video database with human annotations", IEEE International Conference on Computer Vision (ICCV), 2009