

## Обработка перекрытий в задачах отслеживания объектов в видеопотоке

© К.Л. Тассов, Д.Е. Бекасов

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

*Представлено описание задачи отслеживания объектов в видеопотоке, введены основные понятия проблемной области, освещены типовые решения задачи. Более подробно рассмотрены несколько решений с позиции проблемы перекрытий как одной из наиболее весомых. Приведены основные направления будущих исследований.*

**Ключевые слова:** компьютерное зрение, отслеживание объектов в видеопотоке, перекрытия в задачах отслеживания объектов, выделение признаков изображения, трекер, детектор, CAPOA, VMTM, TLD.

**Введение.** Отслеживанием объектов, трекингом (*tracking*), называется определение местоположения движущегося объекта (нескольких объектов) во времени с помощью камеры. При этом анализируются кадры видеопоследовательности и определяется положение движущихся целевых объектов относительно кадра. Данная задача имеет широкий спектр применений: система обеспечения безопасности, анализ потока машин на автомагистрали, освещение и анализ спортивных мероприятий, организация прозрачной работы офиса, система дополненной реальности, различные научные исследования, робототехника, «умный дом», интерактивные и игровые приложения. При решении задачи используются методы математического анализа, математической статистики, обработки изображений, теории принятия решений и систем искусственного интеллекта. Строго говоря, сама задача отслеживания перемещения произвольного объекта в видеопотоке является математически не разрешимой и (без необходимых ограничений) слабо формализуемой. Поэтому при ее решении обычно накладывают ряд ограничений, с помощью которых предполагается работа некоторого метода с заданной точностью. Помимо подразумеваемых ограничений на адекватность входных данных, могут применяться ограничения на количество отслеживаемых объектов, классы объектов, количество и степень перекрытий, продолжительность полного перекрытия. Данные ограничения почерпнуты из основных проблем, возникающих при трекинге объектов. Одной из важнейших подзадач отслеживания является решение проблемы перекрытий отслеживаемого объекта другими. Настоящая статья посвящена рассмотрению алгоритмов трекинга именно с точки зрения проблемы перекрытия объектов.

**Основные понятия.** Для дальнейшего рассуждения необходимо ввести основные понятия, используемые в данной проблемной области.

*Сцена* — область пространства, обозреваемая камерой или системой камер. Она содержит:

- наблюдатель (камеру) — точку наблюдения, субъект;
- статический объект — объект, абсолютное положение которого на сцене не изменяется в заданном временном интервале;
- задний фон — совокупность статических объектов, которые в данной задаче можно рассматривать как единое целое, так как они не перекрывают сопровождаемые объекты на заданном промежутке времени;
- динамический объект — объект, абсолютное положение которого изменяется в течение заданного временного интервала.

При этом динамический объект, принадлежащий заданному классу, называется *объектом интереса*, который сопровождается в данный момент *целью* (*target*). Под *окружением* будет пониматься совокупность объектов сцены, не являющихся целями. Совокупность характеристик, полностью описывающих цель в рамках заданного способа ее представления, будет называться *состоянием объекта*, а функция изменения состояния с течением времени — *траекторией*. Тогда *сопровождение* (*отслеживание*) *цели* — это построение траектории движения цели. При сопровождении может возникнуть явление *перекрывания объектов* (*occlusions*), состоящее в том, что объекты, имеющие меньшие значения координаты  $z$  в системе координат наблюдателя, полностью или частично закрывают от наблюдателя объекты с большим значением  $z$  при близких значениях координат  $x$  и  $y$ .

При описании различных алгоритмов часто используют такое понятие, как *признак* (*feature*) — характерная особенность объекта интереса, по которой осуществляется различение объектов между собой (таких признаков может быть несколько).

Алгоритм, решающий задачу трекинга объекта, будет называться *трекером* (*tracker*).

**Типовые решения, используемые в задачах отслеживания объектов.** Как показало изучение различных источников единой классификации алгоритмов, отслеживания объектов не существует. Это неудивительно, учитывая сравнительно молодой возраст дисциплины (основной объем исследований приходится на последнее десятилетие, что объясняется повсеместным распространением систем видеонаблюдения и общей компьютеризацией). Так, в [1] дается следующая классификация трекеров:

- ручной — отслеживание цели ведется в ручном режиме;
- автоматический — полностью автоматическое отслеживание (в том числе автоматический захват целей и корректировка ошибок);

• интерактивный — компромисс между двумя предыдущими (например, первичное выделение целевого объекта проводится оператором).

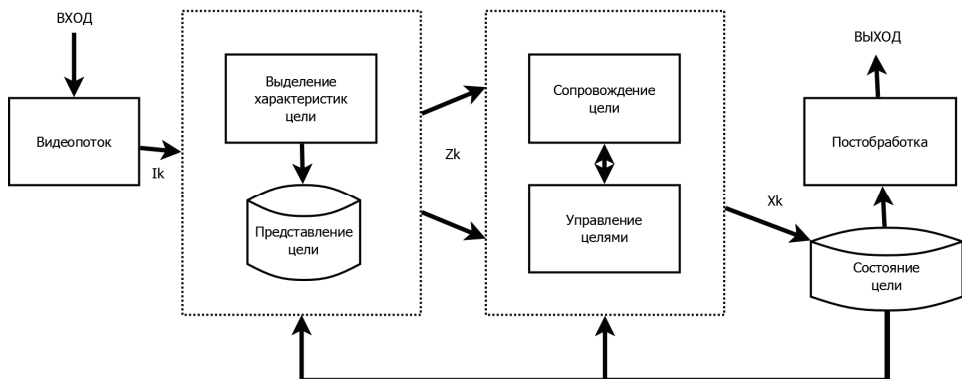


Рис. 1. Типовая структура трекера

На рис. 1 представлена типовая структура трекера, описанная в [1]. Согласно указанной структуре, выделяются следующие основные подзадачи:

- выделения значений признаков объектов (feature extraction);
- выбора (математического) представления цели (target representation);
- сопровождения объекта (localisation);
- управления целями (захват, сопровождение, освобождение) (track management);
- пост-обработки (приведение полученной информации к виду, удобному для использования).

Такая декомпозиция позволяет сформировать блочную структуру трекера с возможностью простой модернизации метода. Более того, можно провести классификацию методов по конкретному решению подзадач.

#### 1. Выделение признаков:

- низкоуровневые признаки (low-level):
  - цвет (приведение к различным цветовым пространствам, например HSL [2],  $C_1C_2C_3$  [1]);
  - градиент или лапласиан (например, с помощью оператора Собеля [1]);
  - движение (например, с помощью векторного поля или поля оптических потоков [1]);
- среднеуровневые признаки (mid-level):
  - границы (например, с помощью детектора Кэнни (Canny) [1]);

- особые точки (interest point) (например, с помощью детектора Моравека (Moravec)[3] или оператора Харриса (Harris) [4]);

- единообразные области (кластеризация низкоуровневых признаков [5–7]);

- высокоуровневые признаки (high-level):

- модель заднего фона (использование математического ожидания и дисперсии [8] или смеси гауссиан [9]);

- модели объектов (обучение на некоторой выборке объектов интереса, например алгоритмы аппроксимации границ объекта (edgelets) [10, 11] или вейвлетных преобразований Хаара [1]);

2. Модели представления цели, основные из которых представлены на рис. 2:

- представление формы (shape):

- простые формы:

- аппроксимация точкой [12];

- аппроксимация занимаемой области (прямоугольник, эллипс) [13];

- аппроксимация занимаемого объема [14];

- каркасные модели (аппроксимируются множеством жестких, неизменяемых блоков) [15];

- деформируемые модели:

- мягкая модель (fluid model) (без ограничений на перемещение отдельных элементов) [16];

- контурные модели [17];

- точечные модели (point distribution models) [18];

- представление внешнего вида:

- шаблоны (для каждого изменения цели) [19];

- гистограммы (цвета [13], ориентации [20], SIFT [21]);

- адаптируемая модель (набор функций распределения для каждого признака) [22].

3. Определение положения цели.

- методы с одной гипотезой (SHL);

- методы на основе вычисления градиента (например, KLT, использующий шаблоны или Mean shift, применяющий цветные гистограммы) [13];

- байесовские методы [1];

- методы с несколькими гипотезами (MHL):

- регулярная сетка (гипотезы для каждого узла сетки) [1];

- частотный фильтр (PF) [1];

- гибридные методы (комбинация с SHL) [23].

Различные комбинации указанных алгоритмов порождают большое число вариантов алгоритмов отслеживания объектов для разнообразных прикладных задач и ограничений. Часто в литературе класс

трекера определяют именно по используемому алгоритму выявления признаков или по способу представления цели. Однако встречаются и другие классификации.

Например, в статье [24] алгоритмы отслеживания четко разделяют:

- на трекеры — отслеживают изменение состояния цели от кадра к кадру;
- детекторы — определяют положение цели на конкретном кадре, без учета временного контекста.

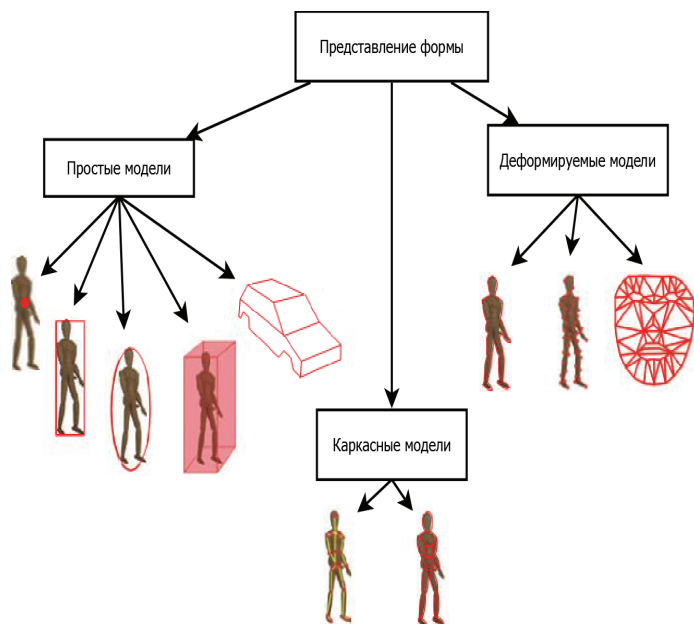


Рис. 2. Модели представления цели

Трекерами могут быть:

- детекторы, определяющие положение цели на конкретном кадре, статическими (модель цели не изменяется с течением времени, например [13]);
- детекторы, определяющие положение цели на конкретном кадре, адаптивными (модель цели адаптируется под изменения изображения цели на видеопоследовательности, например [19]);
- детекторы, определяющие положение цели на конкретном кадре, комбинированными [25];
- детекторы, определяющие положение цели на конкретном кадре, использующими неизменные части шаблонов [26];
- детекторы, определяющие положение цели на конкретном кадре, порождающими (генерируется множество вариантов модели цели) [27].

В свою очередь, порождающие трекары могут генерировать вариации моделей цели:

- в реальном времени (в процессе отслеживания) [28];
- заблаговременно (на этапе инициализации трекара) [27].

Трекары могут моделировать:

- только объект интереса;
- объект интереса вместе с окружением.

Трекары, моделирующие окружение объекта интереса, разделяют:

- на выделяющие объекты, чьи траектории коррелируют с траекторией цели [29];
- вычитающие фон [30].

В свою очередь, детекторы разделяются:

- на использующие локальные признаки изображений [31];
- использующие скользящее окно (sliding window) [32].

**Перекрытия как одна из главных проблем отслеживания объектов.** Задача отслеживания объектов является сложной и многоаспектной, и при ее решении возникает большое число проблем. Ниже перечислены некоторые из них [1, 33]:

- неопределенность — состояние наблюдателя, в котором он не может различить:

- цели и другие объекты интереса;
- цели и другие динамические объекты;
- цели и статические объекты;
- цели и задний фон;

- изменение состояния цели — положение в глобальной системе координат, изменение внешнего вида, смена класса цели;

- изменение освещения — обычная суточная смена освещения, изменение погоды, появление и изменение состояния точечных искусственных источников освещения;

- шум:

- цифровой шум — дефект изображения, вносимый фотосенсорами и электроникой устройств, которые их используют;
- аналоговый шум, возникающий при передаче сигнала;
- управление оптической системой — фокусировка нескольких целей, изменение фокуса при сопровождении цели, перевод фокуса с заднего фона на цель и обратно;

- перекрытие другими объектами:

- по классу объекта:
- перекрытие целевого объекта статическим объектом;
- динамическими объектами;
- другими целевыми объектами;
- по степени перекрытия:

- частичное;
- полное;
- по времени перекрытия:
  - долгосрочное;
  - мгновенное;
  - изменение состояния окружения при изменении состояния наблюдателя.

С неопределенностью, как правило, борются путем выделения специальных, характерных только для конкретной цели признаков [1, 19]. Проблему изменения состояния цели и освещенности решают постоянной адаптацией модели к изменяющимся условиям — обучением трекера или рассмотрением модели в инвариантных для подобных изменений цветовых пространствах (например, HSL [2]). Проблема шума решается применением разнообразных фильтров (к примеру, медианного или ранжирующего [34]), задача управления оптической схемой — физически (системы с оптической стабилизацией) или программно (адаптацией модели).

На проблеме перекрытия объектов стоит остановиться подробнее, так как она является одной из самых нетривиальных и трудно формализуемых. Действительно, отслеживаемый объект может при своем перемещении частично или полностью скрыться из поля зрения на неопределенное время, а трекер должен продолжать отслеживание цели в обычном режиме. Перекрытия оказывают большое влияние на качество отслеживания, часто значительно ухудшая показатели трекеров. Кроме того, существует известная проблема «курицы и яйца» [35] — ситуация перекрытия должна быть определена до того, как цель будет точно локализована в кадре путем маскирования перекрытой части, в то время как определить перекрытую часть цели можно будет только после того, как цель будет локализована в кадре. Ниже представлен обзор существующих решений данной проблемы. Подход к решению задачи во многом определяется типом трекера.

**Адаптация к изменениям цели. WSL Appearance Model.** В [26] предлагается алгоритм адаптивного трекинга, использующего для представления цели три компонента-распределения: стабильный  $S$  (обучаемый за весь период времени), переходный  $W$  (за два кадра) и компонент выбросов  $L$ . При этом перекрытия характеризуются компонентом выбросов, имеющим равномерное распределение. Алгоритм адаптирует модель к медленным изменениям цели, обучая таким образом стабильный компонент. Он наиболее надежен и используется для оценки движения, а переходный — как дополнительное ограничение. Оба этих компонента имеют распределение Гаусса. Обучение модели производится с помощью *EM*-алгоритма, адаптированного под использование в реальном времени. При этом тре-

кер может реализовывать любые признаки изображений. Однако в [26] предлагается применение комплексных коэффициентов специального полосового фильтра для обработки изображений управляемой пирамиды (steerable pyramid). Такая вейвлетная модель обеспечивает стабильность алгоритма при изменении масштаба и пространственного окружения. Недостатки предложенного алгоритма: не учитывает цветность, работает только с медленно перемещающимися объектами, не справляется с задачей, если в окружении имеются похожие объекты или происходит полное перекрытие.

Другой подход определяет перекрытие пикселей при превышении ошибки некоторого порога. Такие алгоритмы хорошо работают, если статистические параметры перекрытий совпадают с ожидаемыми. К сожалению, в большинстве случаев ожидания не выполняются, так как в реальных ситуациях перекрытия могут быть похожи на саму цель или перекрывать ее в течение длительного времени [35].

**Tracking a multivalue template.** Данный алгоритм [36] использует шаблоны для представления цели. Адаптация шаблона к изменению цели проводится путем сглаживания выделенных признаков во времени с помощью фильтра Калмана для каждого пикселя изображения. При своей простоте, алгоритм позволяет отслеживать объекты в реальном времени с частичными перекрытиями, изменениями освещенности, ориентации, перемещениями камеры и является достаточно надежным. Он ставит ограничение на изменение состояния цели и допускает только аффинные преобразования.

В этом методе шаблон представляется вектором признаков для каждой точки. В [36] в качестве признаков предлагается использовать цвет в RGB, нормализованном RGB или в более подходящих к конкретной задаче пространствах. Все векторы обрабатываются независимо друг от друга, вероятность для каждого вектора считается в соответствии с байсовским предположением по всей совокупности измерений. Тогда с помощью гауссова шума для заданного математического ожидания можно получить предсказание для вектора признаков в следующий момент времени. Однако при возникновении перекрытий данное предсказание становится неверным. Для уменьшения влияния перекрытий в выражении для распределения Гаусса корень заменяется нормой ошибки измерений, которая определяется как расстояние Махаланобиса (Mahalanobis) между шаблоном и выделенной в данный момент целью.

Помимо этого в функцию распределения входит робастная функция Хубера (Huber) для обработки ошибки измерений. Ее смысл заключается в том, что если ошибка превышает заданный порог, то очередное измерение (текущее состояние цели) не подчиняется распределению Гаусса, и тогда в формуле распределения используется



не квадратичная, а линейная норма. Поиск совпадения по шаблону заменяется поиском локального минимума функции Хубера от ошибки между шаблоном и предполагаемой целью, задаваемой параметрически. Вектор параметров, по которому проводится минимизация, включает параметры для всех аффинных преобразований: перенос, масштабирование, поворот.

Как было отмечено, если ошибка превышает порог, значит, пиксел перекрыт. Если количество перекрытых пикселов («выбросов») превышает заданное число, то включается режим «перекрытия», и адаптация шаблона отключается. Алгоритм работает только с кратковременными перекрытиями, так как длительное перекрытие начинает оказывать влияние на шаблон и в итоге изменяет его так, что, когда объект перестанет быть перекрываемым, становится невозможным установление сходства между ним и шаблоном. Еще одним недостатком является возможность обнаружения ложных перекрытий.

**Обнаружение перекрытий благодаря изменению движения отдельных областей. Tracking with adaptive block matching.** В [37] представляется блочный контурный трекер. В этом алгоритме ситуация перекрытия анализируется путем сравнения параметров движения между целью и блоками изображения, которые не могут быть скомпенсированы движением (последующими кадрами). Цель представляется маской, инициализирующей в несколько шагов: сначала выполняется многозначная сегментация по четырем направлениям [37], а затем — поиск области одинаково движущихся сегментов (таким образом, анализируется пространственно-временной контекст объекта). Алгоритм предполагает небольшую интенсивность движения в кадре, а потому анализирует только каждый третий кадр. В процессе работы алгоритм ищет блоки изображения, принадлежащие цели, границе цели и заднему фону, и дальше работа идет с полученными блоками. Движение между кадрами определяется путем применения аффинных преобразований к начальным блокам, которыми инициализировалась цель трекинга.

Интересной особенностью алгоритма является рассмотрение событий перекрытия и освобождения от перекрытия как взаимобратных: для определения освобождения от перекрытия (перекрытия), текущий кадр последовательности компенсируется движением предыдущего (следующего) кадра для получения открытых (перекрытых) областей. В случае освобождения от перекрытия (перекрытия) открытые (перекрытые) области, обладающие предложенными характеристиками движения, идентичны (различаются) с объектом. Алгоритм ищет такие сходства (различия) движения. Если в какой-то области движение прекратилось (или различается с движением цели),

значит, эта область перекрыта. Область обозначается неперекрытой, если она возобновляет движение. Возобновление движения определяется путем кластеризации векторов движения областей цели и вычислением расстояния от вектора движения рассматриваемой области до центра этого кластера. Если движение какой-то из областей возобновилось, маска обновляется.

**Использование карты перекрытий.** Алгоритм контентно-адаптивного прогрессивного анализа перекрытий (*CAPOA*). В отличие от обычных, учитывающих только внешний вид цели, данный алгоритм [35] поэтапно анализирует ситуацию перекрытия, принимая во внимание пространственно-временной контекст. При этом получаемая информация дважды проверяется на соответствие цели и ограничениям перемещения, что позволяет лучше разделять целевой объект с объектами-перекрытиями. Для исключения ситуаций ошибочного определения положения цели при перекрытиях используется сопоставление с адаптируемым шаблоном. Вместо того чтобы полагаться на статистическую модель изменений внешнего вида цели, *CAPOA* явно и прогрессивно определяет выбросы, используя пространственно-временной контекст цели. Проблема «курицы и яйца» решена благодаря сопоставлению с адаптируемым шаблоном (алгоритм *VMTM*), где неперекрытая часть цели используется для установки цели из ошибочного положения в правильное. В результате трекер становится устойчивым к большому числу типов перекрытий.

В начальный момент времени цель вручную выделяется определением области интереса (*ROI*). Когда приходит новый кадр, аппроксимация области интереса  $ROI_1$  находится из первого совпадения по образцу. Однако местоположение цели, определенное по первому совпадению, может быть ошибочным, так как использует шаблон, сформированный в соответствии с ситуацией с перекрытиями на предыдущем кадре. Для того чтобы исправить ситуацию, применяется *CAPOA* для анализа ситуации с перекрытиями в  $ROI_1$ , а затем на основе результатов этого анализа — *VMTM*, генерирующий новый *ROI* ( $ROI_2$ ), где ситуация с перекрытиями в котором проанализирована перед этим с помощью *CAPOA*. Ситуация с перекрытиями в  $ROI_2$  генерирует новый шаблон, который проходит поиск на совпадение по второму разу. В итоге получается окончательный *ROI* ( $ROI_3$ ). После определения  $ROI_3$  неперекрытая часть шаблона обновляется с помощью фильтра Калмана.

Необходимо отметить, что анализ ситуации перекрытия  $ROI_{2-3}$  нужен только для определения новых закрытых областей. Поиск соответствия шаблону осуществляется преобразованием системы координат для отображения предполагаемого шаблона на кадр и нахождения области, лучше всего согласующейся с шаблоном. При этом

перекрытия уже являются полностью замаскированными благодаря *CAPOA*.

Алгоритм *CAPOA* получает на вход четыре аргумента: *ROI*, цель, предыдущий кадр и предыдущую карту выбросов. На выходе алгоритм выдает карту выбросов и обновленную маску шаблона. Определение перекрытий происходит на блоках, а не на отдельных пикселях, как в [26, 36], так как пространственный контекст играет важную роль в определении перекрытий (особенно, если перекрытие имеет сходство с целью). Для компромисса между надежностью и разрешением используется прогрессивная процедура сканирования. *ROI* сканируется много раз, и с каждым проходом размер блока уменьшается вдвое. При этом анализируются только те блоки, в которых ситуация с перекрытиями не была решена на предыдущем шаге.

Сканирование заканчивается, когда ситуация с перекрытиями в *ROI* становится полностью определенной. Для того чтобы определить перекрытия, используются предыдущие кадры — для каждого блока ищется похожий на него блок на предыдущем кадре. Если это необходимо, поиск продолжается дальше, до тех пор, пока ситуация с перекрытиями для данного блока не будет точно установлена (поиск может дойти до первого кадра, где ситуация четко определена для каждого блока). Ситуация с перекрытиями на данном кадре задается бинарной картой выбросов. При сканировании *ROI* из старой карты выбросов генерируется новая, временная. Если невозможно однозначно определить, перекрыт блок или нет, то используется оценка движения для данного блока (как в [37]). Если евклидово расстояние до среднего вектора движения модели цели меньше определенного порога, то блок не перекрыт.

Модель цели на каждом шаге алгоритма при необходимости масштабируется и адаптируется посредством фильтра Калмана к изменениям в отслеживаемой цели. Основная задача алгоритма *VMTM* состоит в повышении точности определения местоположения цели. Это достигается поиском локального минимума специальной функции, зависящей от параметров аффинных преобразований (как в [36]) и, что самое важное, учитывающей предварительную карту выбросов. Таким образом, избегаются помехи, связанные с возможными перекрытиями.

Данный метод дает в целом хорошие показатели [35], однако не справляется с полными перекрытиями и объектами-перекрытиями, идентичными отслеживаемой цели.

**Совместное использование трекера и детектора. TLD (Tracking-Learning-Detection).** Метод демонстрирует хорошие показатели и имеет коммерческий успех [24]. Разработан для долгосрочного отслеживания любого класса целей. В отличие от алгоритмов, специа-

лизирующихся только на отслеживании объекта (позволяют адаптироваться к изменению цели, но не справляются с исчезновениями объекта) или только на покадровом поиске объекта (естественно справляется с исчезновением (полным перекрытием), но не позволяет отслеживать изменения, происходящие с целью), TLD использует обе идеи и скрепляет их стадией обучения, на которой с помощью специальной экспертной системы происходит корректировка ошибок отслеживания и поиска.

TLD состоит из трекера, анализирующего движение цели от кадра к кадру (и не справляющегося с ситуацией полного перекрытия) и детектора, независимо сканирующего каждый кадр последовательности (и не справляющегося с изменением состояния цели). При этом детектор может совершать ошибки двух типов: ложные срабатывания и ложные отказы. Обучающий компонент отслеживает состояние и трекера, и детектора и генерирует обучающую выборку для уменьшения числа ошибок последнего. При этом обучающий компонент предполагает возможность одновременного выхода из строя и трекера, и детектора. В процессе обучения детектор обучается определять более широкий диапазон вариаций представления цели и четче разделять цель с задним фоном.

Особый интерес представляет предлагаемый обучающий компонент (P-N Learning). Основная идея состоит в использовании двух экспертов: P-эксперта, определяющего только ложные отказы, и N-эксперта — ложные срабатывания. Несмотря на то что сами эксперты допускают ошибки, их независимое использование компенсирует их ошибки.

Детектор включает в себя бинарный классификатор, сканирующее окно и модель цели, состоящую из обучающей выборки примеров представления цели. Каждый элемент выборки обрабатывается независимо от других. При наличии  $N$  узлов в регулярной сетке сканирования получается  $2^N$  комбинаций для данного кадра. При этом может быть ситуация, когда одной цели соответствует сразу несколько местоположений, что невозможно, так как нарушается связность движения. В этом случае возникает «ложное срабатывание». Именно связность движения является одним из главных параметров для определения ошибок детектора. P-эксперт использует временную связность и предполагает, что цель движется по траектории. Он применяет результаты работы трекера для оценки перемещения цели и, если детектор отбраковывает наиболее вероятное (с точки зрения трекера) местоположение цели, генерирует соответствующую обучающую выборку. Она генерируется из текущей цели с помощью перемещений, масштабирований и поворотов (около 100 экземпляров). N-эксперт использует пространственную связность цели и предполагает, что цель может

находиться только в единственном месте. Он также применяет данные с трекера и среди результатов детектора находит самый подходящий, который обновляет состояние трекера, а остальные добавляет в выборку как негативные экземпляры.

Как было упомянуто выше, в качестве модели цели используется коллекция негативных и позитивных экземпляров — областей изображения. Сходство между двумя экземплярами рассчитывается как нормализованная корреляция. Она же и выступает критерием в классификаторе по ближайшему соседу (Nearest Neighbor). Именно этот классификатор принимает решение об обновлении модели. Детектор использует сканирующее окно для поиска соответствия по шаблону. При большом количестве шаблонов поиск может занять очень много времени. Поэтому используется каскад из трех классификаторов: по разности моделей, классификатор ансамблей и по ближайшему соседу. На практике до последнего классификатора доходит около 50 экземпляров, а такое количество уже можно быстро обработать с помощью NN-классификатора.

В качестве трекера используется Median-Flow трекер, который является расширением трекера Лукаса — Канада (Lucas — Kanade), работающего с оптическими потоками. Расширение включает в себя определение факта потери цели: в этом случае трекер просто не возвращает местоположение (ограничивающую рамку). Факт потери цели устанавливается, если разница смещения некоторой точки цели и медианы всех точек превышает заданный порог. Это может происходить при быстром перемещении или перекрытии цели. Если на определенном шаге ни детектор, ни трекер не выдает местоположения цели, она считается потерянной. В противном случае выбирается наилучший результат (путем расчета корреляции с шаблоном).

При таком подходе детектор опирается на уже известные шаблоны, но может найти цель, если она резко сменила положение или вышла из долгого перекрытия, а трекер привносит новые шаблоны, обновляя базис детектора, и учитывает пространственно-временной контекст.

При всех своих достоинствах, метод имеет следующие недостатки [24]: плохая обработка ситуации поворотов цели, обучение только детектора (трекер остается статическим), нет возможности отслеживания нескольких целей, слабая поддержка подвижных каркасных объектов, таких как пешеходы.

**Несколько частных решений.** Иногда условия и ограничения задачи могут оказать значительную помощь. Например, поставлена задача отслеживания человеческих фигур. В [38] при поиске перекрытий используется относительная глубина объекта (для оценки глубины применяют априорные сведения об окружении и состоянии

наблюдателя). Помимо этого, в [38] используется модель человеческого тела, и каждый пиксел проверяется на соответствие данной модели.

В [39] применяется алгоритм вычитания фона, и каждый объект переднего фона представляется цветовым шаблоном и вероятностной маской. При перекрытии объекта закрытый пиксел будет фигурировать в обеих вероятностных масках. Для определения его принадлежности используется специальный классификатор.

В [40] рассматривается ситуация перекрытия с другой целью (т.е. с известным объектом, что облегчает задачу). При этом каждая цель задается шаблоном и функцией трансформации. Вероятность перекрытия рассчитывается с помощью динамической байсовской сети.

**Основные направления будущих исследований.** Наиболее интересным и комплексным методом из рассмотренных является TLD. Действительно, идея параллельного применения трекера, реализующего принцип пространственно-временной непрерывности траектории цели и детектора, осуществляющего поиск по шаблону, позволяет использовать сильные и нивелировать слабые стороны обоих подходов. Применение простого детектора с поиском по всему кадру естественно для решения проблемы полного перекрытия цели с последующим ее возникновением в неизвестных координатах. В то же время, такой подход не позволяет легко адаптироваться к изменениям цели и провоцирует большое количество ошибок (ложных срабатываний и отказов). Применение простого трекера, отслеживающего цели по ее траектории и использующего информацию с предыдущих кадров, позволяет производить адаптацию к изменениям внешнего вида цели и избежать ошибок, связанных с ложными срабатываниями (если выбраны соответствующие поставленной задаче признаки), однако является совершенно непригодным для борьбы с полными длительными перекрытиями. Поэтому для построения комплексного метода, успешно решающего проблему перекрытий, необходимо использовать оба подхода.

Однако это замечание касается только архитектуры разрабатываемого метода. Конкретная реализация трекера, детектора и обучающего компонента должна являться результатом тщательного исследования условий и ограничений конкретной задачи. Например, если в задаче накладывается условие перемещения наблюдателя, то хорошим выбором в качестве признаков изображения являются оптические потоки; если предполагается постоянное изменение внешнего вида, — то особые точки; если только освещения, — то цвет в пространстве HSL. Кроме того, предложенная в [24] реализация детектора далека от оптимальной: во-первых, используется каскад из трех классификаторов (постепенно отсеивающих варианты); во-вторых,

постоянно растущая модель цели, представленная в виде выборки вариаций. На каждом кадре происходит обработка действительно большого и постоянно растущего массива данных. В-третьих, процесс поиска с использованием сканирующего окна подразумевает проход по всем вариантам его расположения в кадре.

Задачу классификации можно решать с помощью нейронной или нейронечеткой сети. Вообще, использование нечеткой модели для представления постоянно изменяющейся цели довольно естественно и позволило бы достаточно просто определять принадлежность требуемому классу (и при этом хранить модель в компактном виде). Процесс поиска также целесообразно ограничить наиболее вероятными для появления цели областями, что позволило бы избежать большого числа ошибок (ложных срабатываний) детектора. Помимо этого, необходимо лучше обработать ситуацию с частичными перекрытиями, используя в трекаре идеи *CAPOA* [35] с картой перекрытий или адаптации модели (с помощью обучения трекара, как в [36]).

**Заключение.** В статье представлено описание задачи отслеживания объектов в видеопотоке, введены основные понятия проблемной области, освещены типовые решения задачи. Более подробно рассмотрены несколько решений с позиции проблемы перекрытий как одной из наиболее весомых. Несмотря на слабую формализуемость задачи и случайность процесса, проблема перекрытий для заданных ограничений и условий является решаемой [24], [35]. При разработке нового метода необходимо оценить лучшие стороны существующих подходов и предложить интегральное решение.

## ЛИТЕРАТУРА

- [1] Maggio E., Cavallaro A. *Video tracking theory and practice*. Wiley, 2011.
- [2] HSL // Wikipedia. URL: <http://ru.wikipedia.org/wiki/HSL>
- [3] Moravec H. Towards automatic visual obstacle avoidance. *Proceedings of the 5<sup>th</sup> International Joint Conference on Artificial Intelligence*. Cambridge, MA, 1977.
- [4] Harris C., Stephens M. A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147–151.
- [5] Meier T., Ngan K.N. Automatic segmentation of moving objects for video object plane generation. *IEEE Transactions on Circuits and Systems for Video Technology*, 1998, pp. 525–538.
- [6] Alatan A.A., Onural L., Wollborn M., Mech R., Tuncel E., Sikora T. Image sequence analysis for emerging interactive multimedia services-the European COST 211 framework. *IEEE Transactions on Circuits and Systems for Video Technology*, 1998, pp. 802–813.
- [7] Bezdek J.C., Ehrlich R., Full W. Fcm: The fuzzy c-means clustering algorithm. *Computers Geosciences*, 1984, pp. 191–203.
- [8] Cavallaro T. Ebrahimi. Interaction between high-level and low-level image analysis for semantic video object extraction. *EURASIP Journal on Applied Signal Processing*, 2004, no. 6, pp. 786–797.

- [9] Stauffer C., Grimson W.E.L. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, pp. 747–757.
- [10] Nevatia Wu R. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. *In Proceedings of International Conference on Computer Vision. Washington, IEEE Computer Society*, 2005, pp. 90–97.
- [11] Sundararajan K. Unified point-edgelet feature tracking. *A Thesis Presented to the Graduate School of Clemson University In Partial Fulfillment of the Requirements for the Degree Master of Science Electrical Engineering*, 2011.
- [12] Bar-Shalom Y., Fortmann T. *Tracking and Data Association*. New York, Academic Press, 1988.
- [13] Comaniciu D., Ramesh V., Meer P. *Kernel-based object tracking*. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2003, pp. 564–577.
- [14] Koller D., Danilidis K., Nagel H. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 1993, pp. 257–281.
- [15] Sundaresan A., Chellappa R.. Multi-camera tracking of articulated human motion using shape and motion cues. *IEEE Transactions on Image Processing*, 2009, pp. 2114–2126.
- [16] Beymer D., McLauchlan P., Coifman B., Malik J. A real-time computer vision system for measuring traffic parameters. *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, San Juan, Puerto Rico, 1997, pp. 495–501.
- [17] Isard M., Blake A. Condensation — conditional density propagation for visual tracking. *International Journal of Computer Vision*, 1998, pp. 5–28.
- [18] Xiao J., Baker S., Matthews I. and Kanade T. Image on 3D AAM from real-time combined 2D+3D active appearance models. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Washington, DC, 2004, pp. 535–542.
- [19] Shi J., Tomasi C. Good features to track. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Seattle, USA, 1994, pp. 593–600.
- [20] Freeman W.T., Roth M. Orientation histograms for hand gesture recognition. *Proceedings of the Workshop on Automatic Face and Gesture Recognition*. Zurich, Switzerland, 1995, pp. 296–301.
- [21] Lowe D.G. Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision*. Corfu, Greece, 1999, pp. 1150–1157.
- [22] Zhou S., Chellappa R., Moghaddam B. Visual tracking and recognition using appearance-based modeling in particle filters. *IEEE Transactions on Image Processing*, 2004, pp. 491–1506.
- [23] Maggio E., Cavallaro A. Hybrid particle filter and mean shift tracker with adaptive transition model. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2. Philadelphia, PA, 2005, pp. 221–224.
- [24] Kalal Z., Mikolajczyk K., Matas J. Tracking-Learning-Detection. *IEEE transactions on pattern analysis and machine intelligence*, 2010, vol. 6, no. 1, pp. 1–14.
- [25] Rahimi L., P. Morency, T. Darrell. Reducing drift in differential tracking. *Computer Vision and Image Understanding*, 2008, vol. 109, no. 2, pp. 97–111.



- [26] Jepson D., Fleet D.J., El-Maraghi T.F. Robust Online Appearance Models for Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, pp. 1296–1311.
- [27] Black M.J., Jepson A.D. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 1998, vol. 26, no. 1, pp. 63–84.
- [28] Kwon J., Lee K.M. Visual Tracking Decomposition. *Conference on Computer Vision and Pattern Recognition*, 2010.
- [29] Yang M., Wu Y., Hua G. Context-aware visual tracking. *IEEE transactions on pattern analysis and machine intelligence*, 2006, vol. 31, pp. 1195–209.
- [30] Babenko B., Yang M.-H., Belongie S. Visual Tracking with Online Multiple Instance Learning. *Conference on Computer Vision and Pattern Recognition*, 2009.
- [31] Lowe G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004, vol. 60, no. 2, pp. 91–110.
- [32] Viola P., Jones M. Rapid object detection using a boosted cascade of simple features. *Conference on Computer Vision and Pattern Recognition*, 2001.
- [33] Вражнов Д.А., Шаповалов А.В., Николаев В.В. О качестве работы алгоритмов слежения за объектами на видео. *Компьютерные исследования и моделирование*, 2012, т. 4, № 2, с. 303–313.
- [34] Апальков И.В., Хрящев В.В. Удаление шума из изображений на основе нелинейных алгоритмов с использованием ранговой статистики. Ярославский государственный университет имени П.Г. Демидова, 2007.
- [35] Pan J., Hu B. Robust Occlusion Handling in Object Tracking. *IEEE Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [36] Nguyen H.T., Smeulders A.W.M. Fast occluded object tracking by a robust appearance filter. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2004, pp. 1099–1104.
- [37] Hariharakrishnan K., Schonfeld D. Fast object tracking using adaptive block matching. *IEEE Trans. on Multimedia*, 2005, pp. 853–859.
- [38] L. Elgammal, Davis S. Probabilistic framework for segmenting people under occlusion. *Proc. Of IEEE 8th International Conference on Computer Vision*, 2001.
- [39] Senior A.W., Hampapur A., Brown L.M., Tian Y., Pankanti S., Bolle R. M. Appearance Models for Occlusion Handling. *2<sup>nd</sup> International Workshop on Performance Evaluation of Tracking and Surveillance systems*, 2001.
- [40] Wu Y., Yu T., Hua G. Tracking Appearances with Occlusions. *IEEE Computer Vision and Pattern Recognition*, 2003.

Статья поступила в редакцию 10.06.2013

Ссылку на эту статью просим оформлять следующим образом:

Тассов К.Л., Бекасов Д.Е. Обработка перекрытий в задачах отслеживания объектов в видеопотоке. *Инженерный журнал: наука и инновации*, 2013, вып. 6. URL: <http://engjournal.ru/catalog/it/hidden/1099.html>

**Тассов Кирилл Леонидович** родился в 1966 г., окончил МГТУ им. Н.Э. Баумана в 1991 г. Старший преподаватель кафедры «Программное обеспечение ЭВМ и информационные технологии» МГТУ им. Н.Э. Баумана. Автор научных работ в области теории распознавания образов и цифровой обработки сигналов. e-mail: ktassov@policessoft.ru

**Бекасов Денис Евгеньевич** родился в 1991 г. Студент кафедры «Программное обеспечение ЭВМ и информационные технологии» факультета «Информатика и системы управления» МГТУ им. Н.Э. Баумана с 2008 г. Область научных интересов: системы компьютерного зрения. e-mail: [bekasovdenis@gmail.com](mailto:bekasovdenis@gmail.com)