

Санкт-Петербургский государственный университет

Кафедра системного программирования

Антропов Игорь Михайлович

Система слежения автомобиля на малых и средних дистанциях

Бакалаврская работа

Научный руководитель:
к. т. н., доц. Литвинов Ю. В.

Консультант:
инженер-программист компании "Тиквижн" Осечкина М. С.

Рецензент:
генеральный директор ООО "СКЗ" Пименов А. А.

Санкт-Петербург
2018

SAINT PETERSBURG STATE UNIVERSITY

Software Engineering

Antropov Igor

Car tracker on small and medium distance

Graduation Thesis

Scientific supervisor:
associate professor Yurii Litvinov

Consultant:
Software Engineer at Tickvision Mariia Osechkina

Reviewer:
CEO Alexander Pimenov

Saint Petersburg
2018

Оглавление

Введение	4
1. Постановка задачи	6
2. Анализ требований	7
2.1. Функциональные требования	7
2.2. Нефункциональные требования	10
3. Обзор	11
3.1. Методы систем слежения	11
3.2. Подходы к вычислению дескриптора особенностей	15
3.3. Анализ методов	17
3.4. Существующие алгоритмы	18
4. Архитектура и реализация системы	23
5. Апробация прототипа	25
5.1. Критерии тестирования	25
5.2. Сравнение с существующими решениями	25
5.3. Сравнение с требованиям	27
5.4. Выводы	28
Заключение	29
Список литературы	30

Введение

Технический прогресс стремительно развивается, и при текущем технологическом уровне идея абсолютного, массового и повсеместного перехода на беспилотное управление автомобилями не только не кажется абсурдной, но и знаменует собой самое ближайшее будущее. Несмотря на неоднозначность полной автоматизации, у сторонников подобного развития находится множество логичных, очевидных и вполне обоснованных аргументов, среди которых упрощение парковки, минимизация пробок и заторов благодаря исключительно дисциплинированному и упорядоченному движению, значительное сокращение количества дорожно-транспортных происшествий и повышение эффективности использования дорог за счет отсутствия человеческого фактора.

Вместе с тем, их противники отмечают, что полная автоматизация автомобиля приведет к ощущению ложной безопасности, а следовательно — абсолютной невнимательности водителя и неконтролируемости ситуации. Существует мнение, что человеческий фактор по-прежнему будет играть роль: безусловно, большую часть процесса вождения беспилотный автомобиль возьмет на себя, но количество ситуаций, в которых будет требоваться управление человека, сильно зависит от качества программного обеспечения.

Именно поэтому система управления должна быть предельно точной и отказоустойчивой. При этом фактор безопасности ставят на первое место как создатели комплексной системы автоматизации автомобиля, так и разработчики отдельных режимов его автоматического управления в пробках, по скоростной магистрали, автоматической парковки. Чрезвычайно важно максимально точно фиксировать движение автомобиля, определять его местоположение, измерять направление ускорения и замедления, определять угол крена кузова. Все перечисленные данные необходимы для полной автоматизации, но во время движения определяющую роль играют не они.

Одной из важнейших подзадач системы управления беспилотным автомобилем является корректное построение модели текущей ситуа-

ции на дороге. Построить актуальную модель дорожной ситуации без самых точных данных невозможно, поэтому обязательное условие корректности результата заключается в том, что система в точности должна знать местоположение и поведение других участников дорожного движения рядом с ней.

Добиться корректного построения модели текущей ситуации на дороге возможно с помощью постоянной локализации (детекции) и классификации автомобилей. Локализация это процесс нахождения координат объекта, а классификация — процесс определения класса объекта. Алгоритмы классификации заранее обучают делить объекты на определенные классы, например: автомобили, пешеходы и знаки дорожного движения. Система должна работать в режиме реального времени, и так как операции детекции и классификации являются ресурсозатратными, применяются алгоритмы слежения (tracker) за уже опознанными объектами. Алгоритмы слежения позволяют по начальным входным данным выделить некоторые особенности объекта (features) и, используя их, определять местоположение объекта на последующих кадрах.

В сферу деятельности компании «Тиквижн» входит непосредственное создание системы построения модели ситуации на дороге. Для выполнения поставленной задачи компании потребовался алгоритм слежения за автомобилями, способный уменьшить ресурсозатратность системы, необходимость создания которого и стала основой для данной работы.

1. Постановка задачи

Целью данной работы является создание прототипа системы слежения за автомобилями на средних и малых дистанциях. В связи с поставленной целью исследования, были определены следующие задачи.

1. Проанализировать существующие требования к системе слежения от компании «Тиквижн» и разработать дополнительные.
2. Проанализировать существующие подходы и технологии систем слежения за объектами.
3. Разработать архитектуру системы на основе выбранных подходов, технологий и сформулированных требований.
4. Реализовать прототип системы.
5. Провести тестирование прототипа, сравнение с другими решениями и проверить на соответствие требованиям.

2. Анализ требований

В данной главе будут изложены требования к прототипу системы слежения за автомобилями на средних и малых дистанциях. Особенность этих требований заключается в том, что на их основе необходимо разработать прототип, который позволит ответить на вопрос: «Возможно ли уменьшить ресурсозатратность системы помощи управления транспортным средством, используя методы слежения за объектами, не ухудшив точность?».

Если данный прототип покажет успешные результаты, то на его основе будет создан действующий модуль для системы помощи управления транспортными средствами.

Требования к прототипу разделены на функциональные и нефункциональные. Требования точности и устойчивости были разработаны для определения необходимого минимума для дальнейшей разработки модуля. Остальные требования были выдвинуты представителем компании «Тиквижн» для дальнейшей интеграции прототипа в разрабатываемую систему помощи управления транспортным средством.

2.1. Функциональные требования

2.1.1. Требования входных параметров

Входными данными прототипа системы слежения за объектами являются начальное описание целевого объекта и последовательность кадров, на которой необходимо произвести слежение.

За последовательность кадров отвечает одна камера с неопределенным заранее разрешением выходных изображений. Кадры для работы прототипа трехканальные (цветные), они передаются без предварительных обработок. Большое разрешение входных изображений может сильно влиять на скорость работы, поэтому такие ситуации должны быть обработаны.

Начальным описанием целевого объекта служит результат работы модуля локализации объектов. Он представляет из себя координаты об-

ласти кадра, содержащей большую часть объекта или весь объект целиком (описывающий прямоугольник). Одной из особенностей такого представления является отсутствие информации о точной форме объекта.

2.1.2. Дальность работы алгоритма

Требуется корректная работа системы слежения на малых и средних дистанциях. Система должна начинать слежение сразу после получения начальных координат объекта от системы детекции и до момента отдаления объекта более чем на 20 метров.

2.1.3. Результат работы системы

Результат работы системы слежения за объектами должен быть представлен в виде координат объекта на новом кадре. Он должен быть получен после обработки каждого следующего кадра.

Возможно использование данных, полученных после работы модуля детекции, для валидации результатов прототипа системы слежения, а также для корректировки ее работы.

2.1.4. Точность работы

Одним из важных требований к работе системы слежения является точность работы.

Для определения параметра точности вычисляется средний процент перекрытия размеченных данных и результата работы прототипа, когда кадр был успешно обработан. Данное определение параметра также позволит использовать его как критерий сравнения прототипа с другими решениями.

За минимально необходимую точность была взята точность алгоритма KCF [13]. Прототип работает корректно, если его средняя точность работы больше минимально необходимой.

2.1.5. Устойчивость алгоритма

В процессе транспортного движения часто возникают ситуации, когда целевой объект частично или полностью покидает область видимости камеры, или происходит перекрытие другими объектами. Также возможны потери объекта из-за его быстрого перемещения или резкого изменения цвета.

Для измерения характеристики устойчивости вычисляется параметр отказоустойчивости. Он показывает среднее количество подряд обработанных кадров без ре-инициализаций. В дальнейшем он будет использоваться как критерий определения устойчивости алгоритма.

За минимально необходимую отказоустойчивость была взята отказоустойчивость алгоритма КСФ.

2.1.6. Работа в режиме реального времени

Одним из требований к прототипу системы помощи управления транспортными средствами является скорость работы модулей построения текущей модели ситуации на дороге. Система помощи является твердой системой реального времени. Это означает, что допускаются незначительные нарушения предельного срока завершения работы модулей, однако, такие ситуации должны быть обработаны.

Для модуля системы слежения критическим временем, затраченным на обработку одного кадра, является 0.03 секунды.

2.1.7. Объекты слежения

Данный прототип создается для слежения преимущественно за транспортными средствами. Требования к системе не предполагают заранее известные формы и контуры. Также отсутствует какая-либо информация о цвете автомобилей. Они могут быть как однородного цвета, так и с особыми рисунками на кузове.

Помимо машин, немаловажными участниками процесса дорожного движения являются такие транспортные средства, как мотоциклы, велосипеды, прицепы. Прототип должен выполнять корректную работу

при слежении за ними.

2.2. Нефункциональные требования

2.2.1. Отсутствие графического процессора

Графический процессор — устройство, созданное для обработки графической информации. Благодаря специализированной архитектуре он намного быстрее справляется с однотипными математическими расчетами, чем центральный процессор. Широко применяется в алгоритмах компьютерного зрения для ускорения расчета сложных алгебраических операций над матрицами.

В системе помощи управления транспортными средствами графический процессор не предусмотрен.

2.2.2. Язык разработки

Прототип должен быть разработан на языке программирования C++. Язык C++ показал себя как быстрый и достаточно надежный инструмент создания систем, в которых требуется строгое соответствие времени работы предельному сроку завершения.

Также система помощи управления транспортными средствами написана на языке C++, что стало немаловажной причиной выбора языка C++ и для реализации прототипа.

2.2.3. Использование библиотеки OpenCV

OpenCV является одной из крупнейших библиотек для работы с графикой и создания систем компьютерного зрения. Она позволяет представлять изображения в виде матриц и проводить алгебраические операции над ними.

Прототип должен использовать библиотеку OpenCV для корректной обработки графической информации.

3. Обзор

Как уже было сказано, системы помощи управления транспортным средством активно развиваются. Уже есть готовые решения от компаний MobileEye [19] и Tesla [21], однако абсолютного лидера рынка нет, и у каждого решения есть свои недостатки. Так, система от компании Tesla доступна только на их машинах, а система от MobileEye недавно привела к дорожно-транспортному происшествию.

С другой стороны идеальное решение для системы слежения за объектами также не было найдено, поэтому существует множество концептуально разных подходов — как для создания каркасов систем, так и для извлечения особенностей из кадров. Далее будут рассмотрены некоторые из них.

3.1. Методы систем слежения

Чаще всего выделяют четыре основных метода (рис. 1): точечное слежение, слежение с обучением, слежение за ядром и слежение за силуэтами.

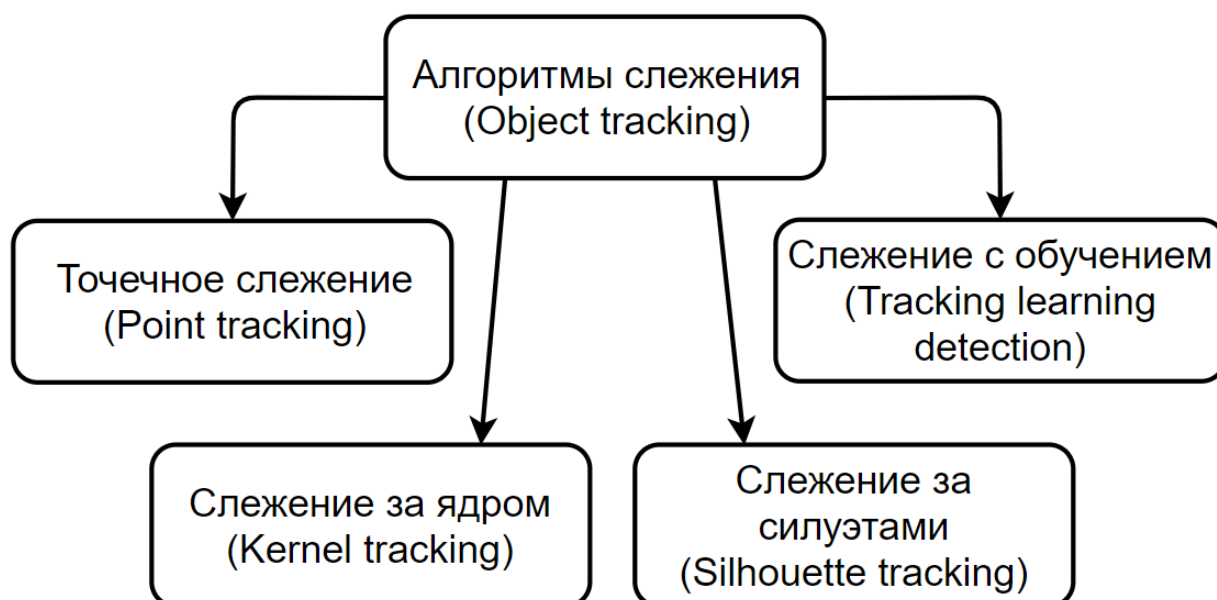


Рис. 1: Основные подходы систем слежения за объектами

3.1.1. Точечное слежение

Задача точечного слежения (point tracking) может быть определена как нахождение и сопоставление точек искомого объекта на протяжении цепочки кадров. Успех точечного подхода сильно зависит от выбора самих точек. Для улучшения качества работы следует выбирать особые точки, такие как границы формы объекта и границы разноцветных областей объекта. Большим минусом такого подхода является некорректная работа при перекрытии объекта, исчезновении и появлении объекта в кадре, ошибки при сопоставлении точек. Метод точечного слежения также делится на 2 основных подхода.

Детерминированный метод

Основной особенностью детерминированного метода [23] является сопоставление объекта на текущем кадре всем объектам на предыдущих. При этом возникает проблема однозначного соединения точек. Ее можно решить, если придерживаться определенных ограничений. Позиция объекта не должна меняться более чем на заранее заданное расстояние. После небольшого перемещения объекта не должно следовать большое перемещение. Расстояние между выбранными точками объекта не должно меняться.

Статический метод

Статический метод [2] представляет собой расчет нового положения точек с использованием информации о скорости, направлении, ускорении объекта и т.д. Способ извлечения информации такого типа из данных камеры неточен, поэтому часто используют и другие сенсоры. Также статический метод в расчетах использует вероятностные константы, так как многие объекты могут двигаться непредсказуемо.

3.1.2. Слежение за ядром объекта

Слежение за ядром объекта (kernel based tracking) [5] основано на нахождении перемещения объекта. В данном типе методов за искомым

объект принимают простейшую геометрическую фигуру, которая содержит в себе большую часть первоначального объекта. Это позволяет снизить сложность расчетов перемещения объекта.

Метод шаблонных моделей

Метод шаблонных моделей (template based models) [4] заключается в построении специального представления искомого объекта — шаблона или гистограммы. Для этого могут использоваться такие способы, как извлечения интенсивности или цвета кадра. Последующие действия метода сводятся к поиску региона следующего кадра, соответствующего шаблону, и к перерасчету самого шаблона. Недостатком данного метода является некорректная работа при большой скорости передвижения объекта.

Метод многообразных моделей

Если в предыдущем методе шаблоны обычно строятся в режиме реальной работы метода, то в методе многообразных моделей (multiview appearance models) [1] необходимо сгенерировать их до работы. Это позволяет избежать проблем со скоростью передвижения объекта, однако, метод работает только для заранее известных типов объектов.

3.1.3. Слежение за силуэтами

В методах слежения за силуэтами (silhouette tracking) подразумевается, что объект обладает сложной, выделяющейся формой, например: ладони, тело, голова. Такие методы требуют повышенной точности описания формы искомого объекта. С помощью информации, извлеченной из самого объекта (формы, цвета, гистограмм), они с каждым новым кадром генерируют модель объекта и используют ее для поиска текущего местоположения. Данный метод делится на два основных подхода.

Сопоставление формы

Метод сопоставления формы [15] выполняет поиск силуэта искомого объекта на последующих кадрах. Метод является аналогичным методу шаблонов. Перерасчет силуэта происходит с каждым новым кадром.

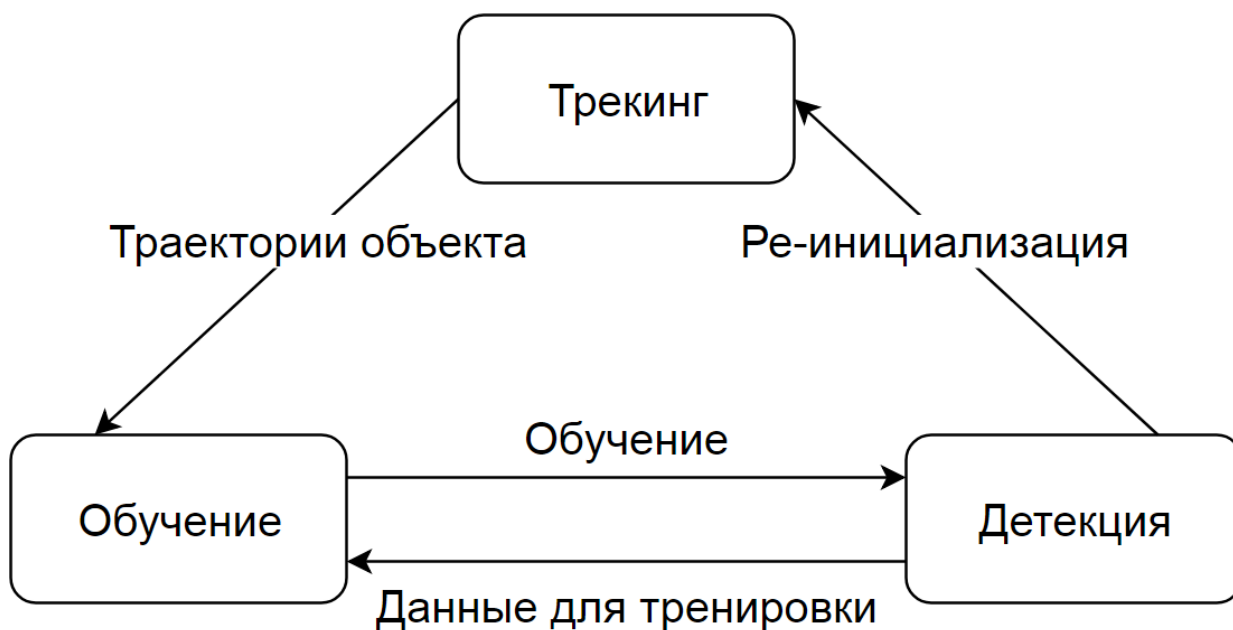


Рис. 2: Цикл работы метода слежения с обучением

Минусом алгоритма является некорректная работа при быстром изменении формы объекта.

Отслеживание контуров

Метод отслеживания контуров [14] итеративно изменяет начальный контур до его положения на новом кадре. Одним из требований является перекрытие контуров прошлого кадра и следующего. Для изменения контура применяется пространство состояний (state space model) для построения модели формы контура и его движения.

3.1.4. Метод слежения с обучением

Метод слежения с обучением (tracking learning detection) [16] считается одним из самых популярных в данный момент, так как рассчитан на долгосрочную работу. Он использует модель «классификация – обучение – слежение» (рис. 2). Слежение используется для нахождения смещения объекта на соседних кадрах. Детекция выполняется независимо для каждого кадра. Результаты работы трекара и детектора анализируются и объединяются. Данный подход помогает обнаруживать ложные результаты работы и корректно реагировать на них. Последним шагом работы метода является независимое обучение детектора и

ре-инициализация трекера при необходимости.

3.2. Подходы к вычислению дескриптора особенностей

За извлечение полезной информации из объекта отвечают дескрипторы особенностей. Они упрощают изображение, отбрасывая ненужную информацию, тем самым облегчая вычисления. Рассмотрим некоторые из них.

3.2.1. Сверточные нейронные сети

Сверточные нейронные сети [18] – особый вид нейронных сетей, использующий операцию свертки. Она заключается в перемножении каждого фрагмента изображения и ядра свертки. Результат суммируется и записывается в ячейку матрицы, к которой была применена свертка. Такой подход позволяет сохранить пространственную информацию, из-за чего сверточные нейронные сети нашли свое применение в работе с компьютерной графикой. Они состоят из двух частей: сверточные слои и обычная нейронная сеть (перцептрон). Сверточные слои находят список особенностей, после чего перцептрон по ним классифицирует объект. Однако, алгоритмы слежения обрабатывают список особенностей напрямую, исключая работу перцептрона.

3.2.2. Гистограммы направленных градиентов

Метод гистограмм направленных градиентов [6] состоит из нахождения горизонтальных и вертикальных градиентов, разбиения входного изображения на ячейки определенного размера, нахождения гистограмм градиентов внутри каждой ячейки, нормализации и расчета вектора особенностей.

$$\begin{vmatrix} -1 & 0 & 1 \end{vmatrix} \text{ и } \begin{vmatrix} -1 \\ 0 \\ 1 \end{vmatrix} \quad (1)$$

Первый шаг заключается в применении фильтров с ядрами (1) к входному изображению для нахождения горизонтальных и вертикальных градиентов (рис. 3). После происходит подсчет величины и направления градиентов с помощью следующих формул.

$$g = \sqrt{g_x^2 + g_y^2}$$
$$\theta = \arctan\left(\frac{g_y}{g_x}\right)$$



Рис. 3: Градиенты изображения (изображение взято с ресурса <https://www.learnopencv.com/histogram-of-oriented-gradients/>)

Следующим шагом является разбиение входного изображения на ячейки. Шаг разбиения зависит от размера самого объекта и размера его особенностей. Для построения гистограммы необходимо сложить значения из матрицы величины тех пикселей, значения матрицы направления которых совпадают.

Нормализация необходима для уменьшения зависимости значений вектора от интенсивности освещения объекта. Обычно нормализацию проводят в пространстве L2 сразу для нескольких ячеек.

Последним шагом метода является объединение всех нормализованных векторов особенностей в один результирующий вектор.

3.2.3. Цветовой дескриптор

На практике искомые объекты часто обладают областями с одним ярко выраженным цветом. Это позволяет использовать информацию, полученную с помощью цветового дескриптора [17].

Концепцией подхода цветового дескриптора является отображение RGB представления объекта в классы, разделенные по цвету. Создателями метода была построена таблица отображения в 12 стандартных цветов.

3.2.4. Дескриптор черно-белого изображения

Существует несколько представлений цветного изображения. Одно из них состоит из трех частей, называемых каналами, и каждая точка описывается тремя значениями: R – интенсивность красного, G – интенсивность зеленого, B – интенсивность синего.

Дескриптор черно-белого изображения переводит трехканальное RGB изображение в одноканальное, где каждое значение отображает интенсивность серого цвета точки.

Следующим шагом является нормализация полученного изображения по формуле. Результатом работы дескриптора черно-белого изображения является это нормализованное представление.

3.3. Анализ методов

Выбор метода системы слежения был произведен с учетом ранее объявленных требований. Так, метод точечного слежения не подходит как решение данной задачи из-за его проблем с обработкой объектов при коллизиях. Метод слежения за силуэтами и multiview appearance models также не являются оптимальным из-за отсутствия какой-либо заранее известной информации о форме целевого объекта. Template based model требует малой скорости передвижения объекта, однако, это

можно нивелировать использованием методов машинного обучения, поэтому был выбран подход tracking learning detection.

Такой подход требует корректного нахождения дескрипторов особенностей. Дескриптору сверточной нейронной необходим графический процессор для достаточно быстрой работы. Дескриптор черно-белого изображения, в отличие от гистограмм направленных градиентов, неустойчив к изменению освещения. Также в результате работы дескриптора черно-белого изображения происходит потеря существенной информации о цвете целевого объекта. Из-за этого был выбран дескриптор цвета.

3.4. Существующие алгоритмы

Были рассмотрены алгоритмы систем слежения KCF [13], MOSSE [24], CSR-DCF [8], ECO [9], ECOhs [9].

Каждый из этих алгоритмов состоит из двух шагов. Первым шагом является нахождение дескрипторов особенностей, их обработка и вычисление новых координат целевого объекта. На втором шаге выполняется обновление внутреннего представления целевого объекта и изменение параметров фильтров.

Одна из общих идей всех перечисленных алгоритмов заключается в применении корреляционных фильтров. Кросс-корреляция такого фильтра и входного изображения позволяет построить матрицы достоверности. Значение точки матрицы достоверности показывает вероятность нахождения объекта в этой точке.

Также все перечисленные алгоритмы используют дискриминативную модель объекта. Этот подход заключается в использовании не только информации об объекте, но и о его заднем фоне.

3.4.1. MOSSE

Алгоритм MOSSE (Minimum Output Sum of Squared Error) – один из первых алгоритмов, который использует корреляционные фильтры. Для работы необходимы заранее известные серые примеры объекта для обучения корреляционного фильтра. Обучение происходит за счет ми-

нимизации суммы квадратичной ошибки. Более подробно об этом подходе написано в статье [24]. Предназначен для работы с черно-белыми изображениями.

3.4.2. КСФ

Алгоритм КСФ (Kernelized Correlation Filter) является одним из самых известных алгоритмов слежения. В нем используются циклические сдвиги и изменения масштаба для порождения большего количества примеров. Для подсчета корреляционного фильтра используется связка метода регуляризации Тихонова [22] и циркулянтов [3]. Использование преобразования Фурье позволяет значительно ускорить подсчеты. Вычисление особенностей производится путем использования гистограмм направленных градиентов. DCF (Dual Correlation Filter) – многоканальная версия алгоритма КСФ.

3.4.3. ЕСО и ЕСОhs

Алгоритм ЕСО использует сверточные нейронные сети для вычисления особенностей объектов, поэтому далее будет рассмотрена его измененная версия ЕСОhs.

Особенности и их обработка

Работа с особенностями алгоритма ЕСОhs схематично представлена на рисунке 4.

Первым шагом работы с особенностями алгоритма ЕСОhs является создание копий входного изображения, различающихся по размеру. Этот шаг необходим для более точной работы из-за заранее неизвестного размера объекта и размера его особенностей.

На следующем шаге происходит извлечение и нормализация векторов особенностей из полученных ранее изображений. Далее выполняется быстрое преобразование Фурье для ускорения расчетов алгебраических операций. Теорема Фурье о свертке утверждает, что применение фильтра к изображению эквивалентно перемножению коэффициентов разложения Фурье изображения и фильтра.

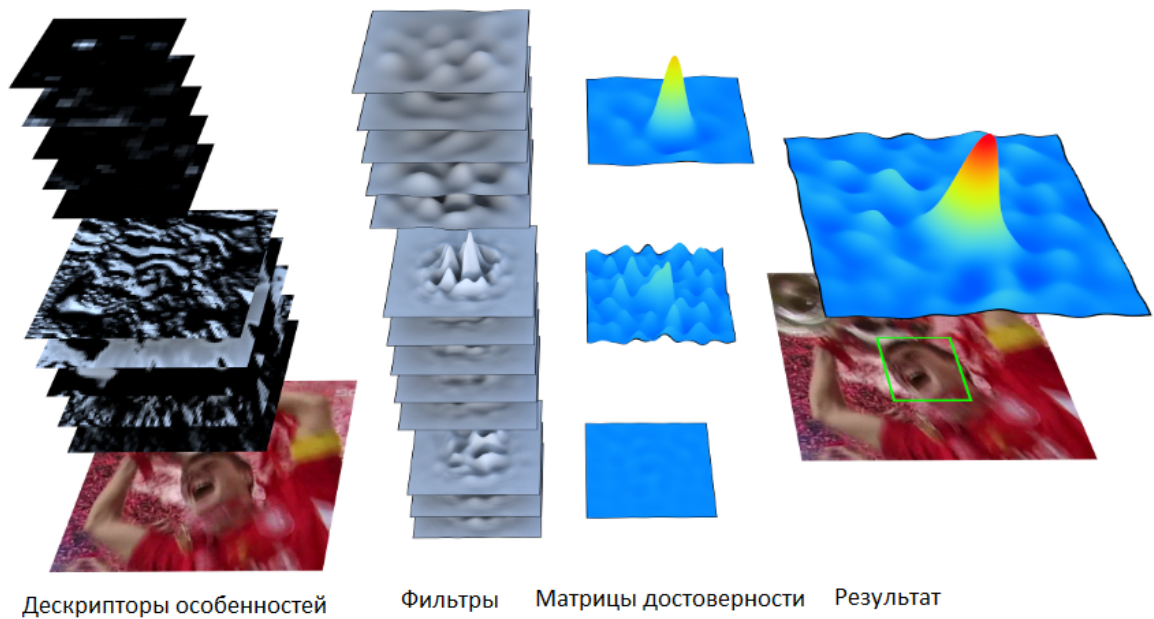


Рис. 4: Шаги работы с особенностями алгоритма ESOhs (изображение взято из статьи [9])

Полученные таким образом карты особенностей подвергаются интерполяции в непрерывную пространственную область (continuous spatial domain). В результате карты особенностей оказываются в виде непрерывных периодических функций.

Следующим шагом является применение фильтров. Для каждого слоя каждой карты особенностей используется соответствующий фильтр. В результате получают матрицы достоверности, значения в которых показывают вероятность принадлежности точки целевому объекту.

Последним шагом производится объединение матриц и подсчет координат результата.

Модель объекта и ее обучение

В качестве модели целевого объекта алгоритм ESOhs использует генеративную модель образцов. Она представляет из себя смесь компонент Гаусса [12], каждый из которых отображает разные аспекты объекта. Это позволяет уменьшить требуемое количество сохраненных примеров целевого объекта, а также снизить шанс переобучения модели.

После нахождения координат целевого объекта происходит провер-

ка на необходимость добавления данного примера в модель. Если в модели примеров меньше, чем заранее заданное значение, то происходит добавление. Иначе производится поиск элемента модели с наименьшим расстоянием до проверяемого примера. За нахождение расстояния отвечает метод Деклерка и Пиатера [7]. Если это минимальное расстояние больше заранее заданного значения, производится операция слияния двух примеров. Данная проверка проводится после каждого успешного нахождения координат объекта.

Для избежания ресурсозатратности и увеличения скорости работы алгоритма обновление фильтров производят в разреженном характере. Это означает, что обновление будет происходить только тогда, когда выполнены два условия: количество обработанных кадров с прошлого обновления больше, чем заранее заданное значение, и новый пример значительно отличается. При обновлении параметров фильтров необходимо решить систему линейных уравнений. Для этого применяется метод сопряженных градиентов.

3.4.4. CSR-DCF

Первым отличием алгоритма CSR-DCF от ECOhs является использование карты пространственной надежности (spatial reliability map) для обучения модели. Карта пространственной надежности представляет из себя матрицу, которая показывает принадлежность определенной точки объекту. Это позволяет игнорировать заведомо известные шумы при обучении фильтров. Также этот подход решает проблему описывающего прямоугольника, когда у объекта необычная форма, например тор.

Вторым отличием является использование весовых коэффициентов надежности (reliability weights). Они используются при подсчете результирующего отклика корреляции для улучшения точности.

3.4.5. Выбор алгоритма

Алгоритмы MOSSE и KCF показали неудовлетворительную точность работы. Главным отличием алгоритма CSR-DCF от ESOhs является поддержка особых форм объекта, что не особо важно в контексте данной задачи.

Для реализации был выбран алгоритм ESO. Однако он использует сверточные нейронные сети, поэтому более подходящей под задачу оказалась его измененная версия ESOhs.

4. Архитектура и реализация системы

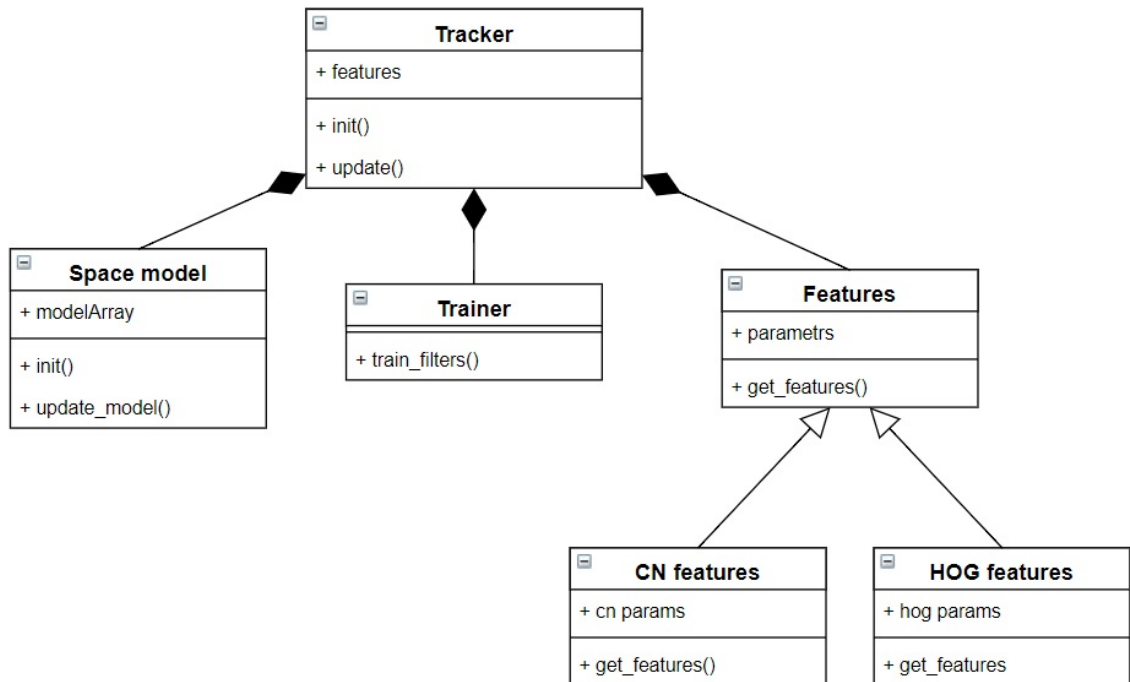


Рис. 5: Диаграмма классов

На рис.5 представлена диаграмма классов реализации алгоритма ESOhs. Далее будет рассмотрен каждый класс в отдельности.

Класс Tracker отвечает за инициализацию начальных параметров, нахождение и обработку особенностей входного изображения, а также за расчет результирующих координат целевого объекта. Класс хранит в себе список текущих фильтров и модель объекта.

Нахождение особенностей выполняется в классе Features. Экземпляр класса производит поиск векторов особенностей на входящих изображениях. Расчет цветового дескриптора производится в экземпляре класса CNfeatures, а гистограммы направленных градиентов в экземпляре класса HOGfeatures.

Класс Space model отвечает за создание модели объекта, и обеспечивает ее корректное обновление по необходимости.

В классе Trainer реализован алгоритм обновления фильтров с помощью метода сопряженных градиентов. Результат передается в класс

Tracker для нахождения результирующих координат.

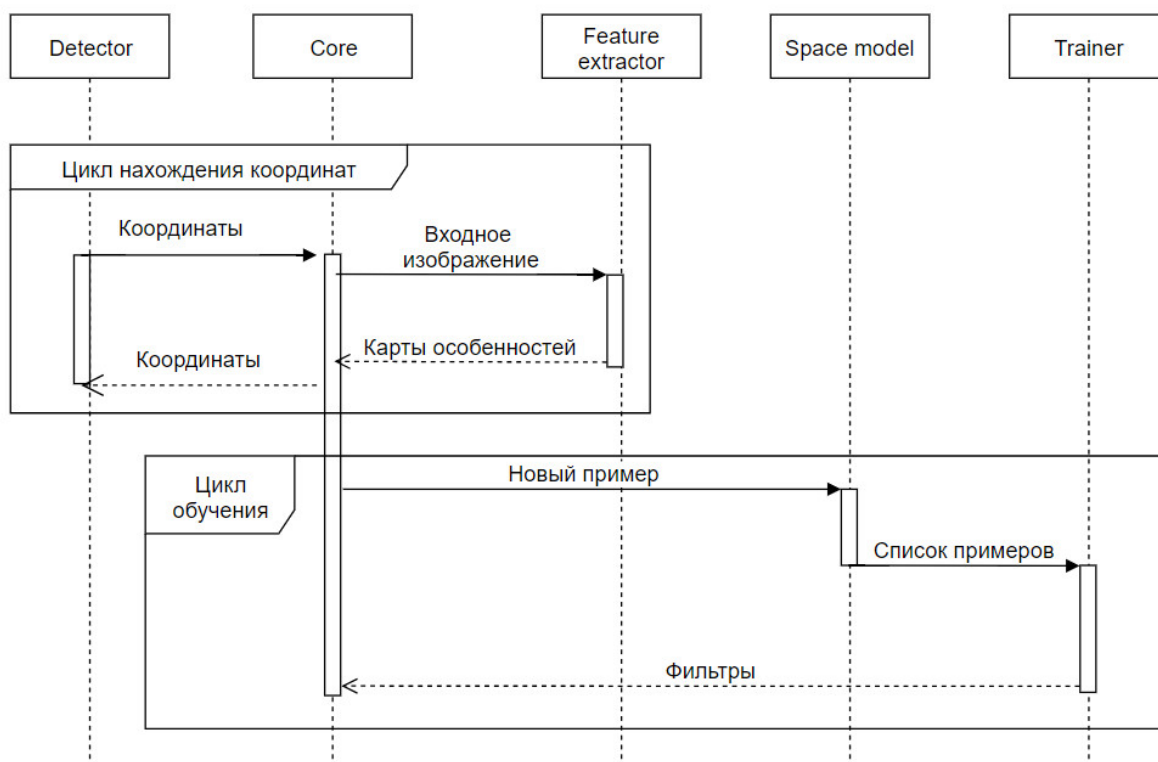


Рис. 6: Диаграмма последовательностей

Обработка одного кадра состоит из двух независимых частей – цикл нахождения координат и цикл обучения, что показывает диаграмма последовательностей (рис.6).

В цикле нахождения координат взаимодействуют подсистемы Detector, Core и Feature extractor. В цикле обучения происходит обновление модели объекта и фильтров с помощью работы подсистем Core, Space model и Trainer.

5. Апробация прототипа

В данной главе будет описано сравнение прототипа с другими алгоритмами и проверка на соответствие требованиям. Эксперименты проводились на компьютере со следующими характеристиками: процессор – Intel Core i7-4710HQ 2.50GHz, 8 гигабайт оперативной памяти DDR3, тип жесткого диска – HDD.

5.1. Критерии тестирования

В качестве параметров сравнения используются параметры точности и отказоустойчивости. Проводимое тестирование является самообновляющимся. Это значит, если пересечение результата тестируемого прототипа системы слежения и размеченных данных меньше заранее заданного значения (было выбрано 30%), то происходит ре-инициализация системы слежения через пять кадров.

Точность прототипа на определенном кадре определяется как отношение пересечения предполагаемого описывающего прямоугольника и реального к их объединению. Средняя точность вычисляется как отношение суммы точности на каждом кадре к числу кадров.

Параметр отказоустойчивости показывает среднее количество обработанных кадров без ре-инициализаций. Из-за свойства самообновления тестирования возможны две ситуации, когда требуется ре-инициализация: алгоритм не смог построить модель объекта за несколько первых кадров (отрицательный случай); точность работы алгоритма упала из-за накопления ошибок (положительный случай). Эти ситуации приводят к мультимодальному распределению параметра отказоустойчивости, поэтому будут рассмотрены отдельно.

5.2. Сравнение с существующими решениями

Для сравнения результатов были использованы 29 видео из дата-сета ОТВ (Object Tracking Benchmark) [20] с суммарным количеством кадров 15870.

В таблицах 1 и 2 приведены получившиеся результаты. Жирным шрифтом выделены результаты работы реализованного прототипа алгоритма ECOhs.

Название алгоритма	Точность		Кадров в секунду		Устойчивость
	Среднее	Дисперсия	Среднее	Дисперсия	
BACF	0.65	0.09	13	5	53.1
CSR-DCF	0.62	0.10	15	20	48.3
KCF	0.51	0.13	331	113	36.5
MOSSE	0.47	0.15	732	204	22.3
ECOhs	0.73	0.08	7.3	4	58.4

Таблица 1: Результаты точности, скорости и средней устойчивости различных алгоритмов

Название алгоритма	Отрицательная устойчивость		Положительная устойчивость	
	Среднее	Дисперсия	Среднее	Дисперсия
BACF	2.47	9.7	74.9	336
CSR-DCF	5.77	14.1	69.4	284
KCF	5.7	18.3	58.2	187
MOSSE	1.6	10.5	50.7	152
ECOhs	3.42	11.7	83.2	341

Таблица 2: Результаты отрицательной и положительной устойчивости различных алгоритмов

Скорости работы алгоритмов KCF и MOSSE удовлетворяет требованию работы в режиме реального времени, но это достигается низкими параметрами точности и отказоустойчивости. Получается, что алгоритм MOSSE требовал ре-инициализации примерно каждые 22 кадра со средним параметром точности в 47%. Алгоритм KCF достиг примерно той же точности, но ре-инициализация происходила каждые 36 кадров.

Скорости работы CSR-DCF и BACF [11] не удовлетворяют требованию, а результаты точности и отказоустойчивости оказались меньше, чем у ECOhs.

5.3. Сравнение с требованиям

5.3.1. Точность

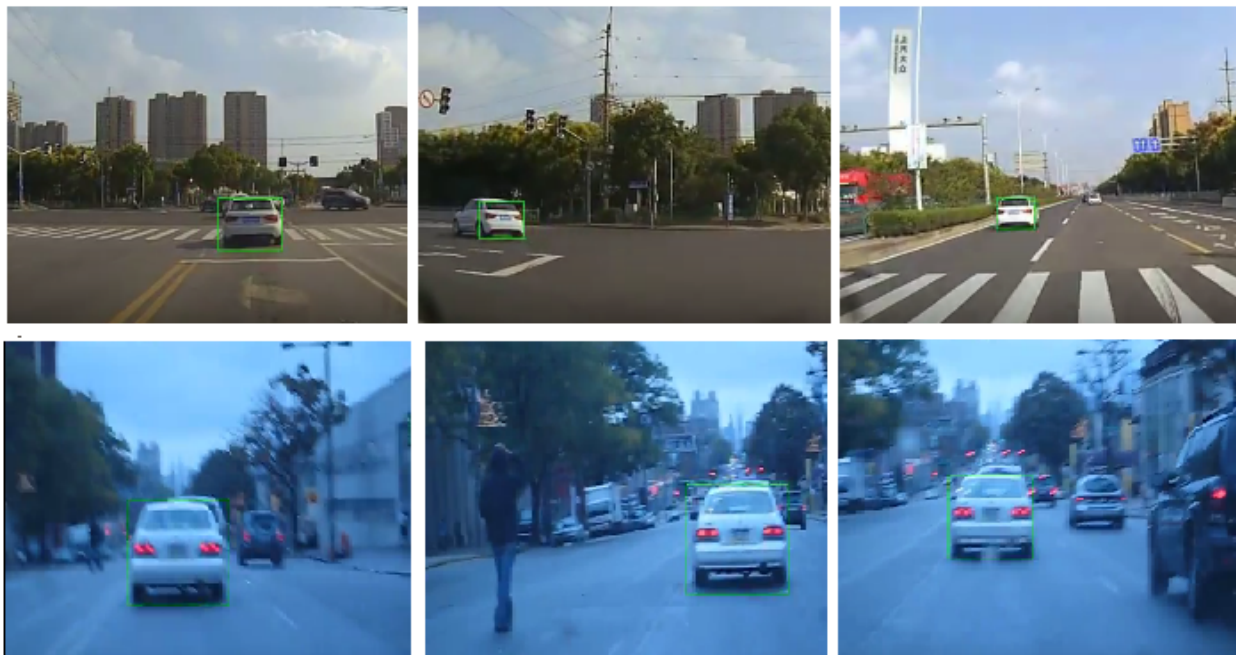


Рис. 7: Примеры: верхняя строка – поворот, нижняя – размытость

Параметр средней точности прототипа получился 0.73, что удовлетворяет заявленному требованию.

Также прототип обрабатывает такие сложные ситуации, как повороты или размытые входные изображения (рис. 7)

5.3.2. Устойчивость



Рис. 8: Пример перекрытия объекта

Параметр средней отказоустойчивости прототипа получился 58.4, что удовлетворяет заявленному требованию.

Прототип производит корректную обработку перекрытия целевого объекта (рис. 8), если выполнены следующие условия: в модели целевого объекта нет места для новых примеров; полное перекрытие продолжается на протяжении не более 30 кадров.

5.3.3. Работа в режиме реального времени

Среднее количество обработанных кадров в секунду сильно зависит от частоты обучения фильтров. Тестирование показало, что оптимальным вариантом является обновление после каждых пяти кадров. При таком подходе алгоритм будет своевременно реагировать на изменения целевого объекта, но и не будет переобучаться. Из-за этого среднее время обработки одного кадра равняется 0.13 секундам.

5.3.4. Объекты слежения



Рис. 9: Пример слежения за мотоциклом

Прототип корректно выполняет слежение за другими видами транспортных средств (рис 9).

5.4. Выводы

Прототип получился достаточно точным и устойчивым, но скорость обработки каждого кадра не позволяет использовать его в системах реального времени.

Дальнейшее развитие прототипа возможно в двух направлениях. Его можно ускорить, используя библиотеку Eigen [10] и многопоточность. Либо использовать для построения проверочных данных, на которых можно будет тестировать другие подходы и алгоритмы.

Заключение

В ходе работы были получены следующие результаты.

- Проанализированы существующие и разработаны дополнительные требования к прототипу, из которых основными являются: требование типа входных данных; точность работы алгоритма; отсутствие графического процессора и язык реализации C++.
- Проведен анализ подходов и технологий систем слежения.
- Разработана архитектура прототипа на основе следующих подходов и технологий: ESOhs, гистограммы направленных градиентов, цветовой дескриптор.
- Реализован прототип системы слежения за автомобилями на малых и средних дистанциях.
- Выполнено сравнение точности и устойчивости прототипа с существующими решениями. Проверено соответствие прототипа и требований. Видео примеров работы прототипа выложены на сайте youtube¹. Код работы выложены на сайте github².

¹<https://www.youtube.com/channel/UCVsPQiOT2FuPV6Q4lk4UUaA>

²<https://github.com/Antropovi/tracker>

Список литературы

- [1] Avidan S. Support Vector Tracking // Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. — 2001. — P. I-184–I-191 vol.1.
- [2] Brodia Ted J, Chellappa Rama. Estimation of object motion parameters from noisy images // IEEE transactions on pattern analysis and machine intelligence. — 1986. — no. 1. — P. 90–99.
- [3] Circulant matrix // Wikipedia. — URL: https://en.wikipedia.org/wiki/Circulant_matrix (online; accessed: 26.05.2018).
- [4] Comaniciu Dorin, Ramesh Visvanathan. Real-time tracking of non-rigid objects using mean shift. — 2003. — US Patent 6,590,999.
- [5] Comaniciu D., Ramesh V., Meer P. Kernel-based object tracking // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2003. — May. — Vol. 25, no. 5. — P. 564–577.
- [6] Dalal N., Triggs B. Histograms of oriented gradients for human detection // 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). — Vol. 1. — 2005. — June. — P. 886–893 vol. 1.
- [7] Declercq Arnaud, Piater Justus H. Online Learning of Gaussian Mixture Models - a Two-Level Approach // VISAPP. — 2008.
- [8] Discriminative Correlation Filter with Channel and Spatial Reliability / Alan Lukezic, Tomás Vojír, Luka Cehovin et al. // CoRR. — 2016. — Vol. abs/1611.08461.
- [9] ECO: Efficient Convolution Operators for Tracking / Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, Michael Felsberg // CVPR. — 2017.

- [10] Eigen benchmark // Eigen home site.— URL: <http://eigen.tuxfamily.org/index.php?title=Benchmark> (online; accessed: 26.05.2018).
- [11] Galoogahi Hamed Kiani, Fagg Ashton, Lucey Simon. Learning Background-Aware Correlation Filters for Visual Tracking // CoRR. — 2017. — Vol. abs/1703.04590.
- [12] Gaussian mixture model // Wikipedia.— URL: https://en.wikipedia.org/wiki/Mixture_model (online; accessed: 26.05.2018).
- [13] High-Speed Tracking with Kernelized Correlation Filters / João F. Henriques, Rui Caseiro, Pedro Martins, Jorge Batista // CoRR. — 2014. — Vol. abs/1404.7584. — 1404.7584.
- [14] High-quality video view interpolation using a layered representation / C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele et al. // ACM transactions on graphics (TOG) / ACM. — Vol. 23. — 2004. — P. 600–608.
- [15] Image inpainting / Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, Coloma Ballester // Proceedings of the 27th annual conference on Computer graphics and interactive techniques / ACM Press/Addison-Wesley Publishing Co. — 2000. — P. 417–424.
- [16] Kalal Z., Mikolajczyk K., Matas J. Tracking-Learning-Detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2012. — July. — Vol. 34, no. 7. — P. 1409–1422.
- [17] Learning color names for real-world applications / Joost Van De Weijer, Cordelia Schmid, Jakob Verbeek, Diane Larlus // IEEE Transactions on Image Processing. — 2009. — Vol. 18, no. 7. — P. 1512–1523.
- [18] Li D., Chen W. Object tracking with convolutional neural networks and kernelized correlation filters // 2017 29th Chinese Control And Decision Conference (CCDC). — 2017. — May. — P. 1039–1044.

- [19] MobileEye technology // Home site MobileEye inc. — URL: <https://www.mobileeye.com/our-technology/> (online; accessed: 26.05.2018).
- [20] Object Tracker Dataset // Computer Vision Lab. — URL: http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html (online; accessed: 26.05.2018).
- [21] Tesla autopilot // Home site Tesla inc. — URL: <https://www.tesla.com/autopilot/> (online; accessed: 26.05.2018).
- [22] Tikhonov regularization // Wikipedia. — URL: https://en.wikipedia.org/wiki/Tikhonov_regularization (online; accessed: 26.05.2018).
- [23] Veenman Cor J, Reinders Marcel JT, Backer Eric. Resolving motion correspondence for densely moving points // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2001. — Vol. 23, no. 1. — P. 54–72.
- [24] Visual object tracking using adaptive correlation filters / David S Bolme, J Ross Beveridge, Bruce A Draper, Yui Man Lui // Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on / IEEE. — 2010. — P. 2544–2550.