

## Распределённый алгоритм сопровождения людей в видео

Купляков Д.А.<sup>1</sup>, Шальнов Е.В.<sup>2</sup>, Конушин В.С.<sup>3</sup>, Конушин А.С.<sup>4</sup>

denis.kuplyakov@graphics.cs.msu.ru | eshalnov@graphics.cs.msu.ru | vadim@tevian.ru | ktosh@graphics.cs.msu.ru

<sup>1,2,4</sup> Факультет вычислительной математики и кибернетики,

Московский государственный университет им. М.В. Ломоносова, Москва, Россия;

<sup>1,3</sup> ООО «Технологии видеоанализа», Москва, Россия

*В работе рассматривается задача сопровождения людей в видеопотоке с целью их подсчёта. Современные системы видеонаблюдения такие, как система видеонаблюдения Москвы, содержат сотни тысяч камер. Использование современных методов, разрабатываемых для работы на одном компьютере с дорогим графическим ускорителем, не является экономически целесообразным для систем подобного масштаба. В работе предлагается распределённый алгоритм сопровождения. Он позволяет за счёт детекции на разреженном множестве кадров сократить число необходимых вычислительных ресурсов. Детекция производится на серверах в центре обработки данных, а видеопоток обрабатывается локальными для камер узлами. Представлена методика оценки качества подсчёта людей. Экспериментальная оценка показала, что предложенный алгоритм позволяет достичь приемлемого качества при частоте детекции 4/3 Гц.*

**Ключевые слова:** компьютерное зрение, сопровождение в видео, подсчёт людей.

## Distributed pedestrian video tracking algorithm

Kuplyakov D.<sup>1</sup>, Shalnov E.<sup>2</sup>, Konushin V.<sup>3</sup>, Konushin A.<sup>4</sup>

denis.kuplyakov@graphics.cs.msu.ru | eshalnov@graphics.cs.msu.ru | vadim@tevian.ru | ktosh@graphics.cs.msu.ru

<sup>1,2,4</sup> Department of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Moscow, Russia;

<sup>1,3</sup> «Video Analysis Technologies» LLC, Moscow, Russia

*Problem of tracking people on video with the purpose of their counting is considered. Modern video surveillance systems, like Moscow's video surveillance system, consist of hundreds of thousands of cameras. The use of modern tracking methods, developed to work on a single computer with expensive graphics accelerator, is not economically feasible for systems of such scale. The paper proposes a distributed tracking algorithm. It allows to reduce amount of necessary computational resources by detection on rare frames. Detection is performed on servers in a data center, and the video stream is processed by local to cameras nodes. A methodology for assessing the quality of people counting is presented. The experimental evaluation shows that the proposed algorithm allows to reach an acceptable quality with a detection frequency 4/3 Hz.*

**Keywords:** computer vision, video tracking, people counting.

### 1. ВВЕДЕНИЕ

Практически важной задачей является подсчёт числа людей, прошедших через определённые зоны общественной инфраструктуры, такие как пешеходные переходы, тротуары, площади и т.д. Одним из возможных путей её решения является сопровождение людей в видеопотоке, то есть построение траекторий их движения. Траектория однозначно соответствует человеку и содержит его положение на всех кадрах, где он представлен. Для подсчёта людей обычно на кадре фиксируется *сигнальный отрезок*. При пересечении траекторией человека данного отрезка фиксируется один проход, что позволяет подсчитывать людей.

В работе рассматривается задача полностью автоматического подсчёта людей в видео, снятого неподвижной камерой. На вход алгоритма поступает видеопоток  $\{F_i\}_{i=1}$  кадров, полученных с неподвижной камеры и сигнальный отрезок, заданный упорядоченной парой точек  $(L_a, L_b)$  в кадре. На выходе — поток событий пересечения сигнального отрезка  $\{E_i\}_{i=1}$ . Событие  $E_i = (k_i, r_i, d_i)$  описывается номером кадра  $k_i$ , на котором произошло пересечение сигнального отрезка, прямоугольником  $d_i$ , ограничивающим изображе-

ние человека на кадре в момент пересечения, и направлением пересечения сигнального отрезка  $r_i$ . Данная задача рассматривалась также в работе [16].

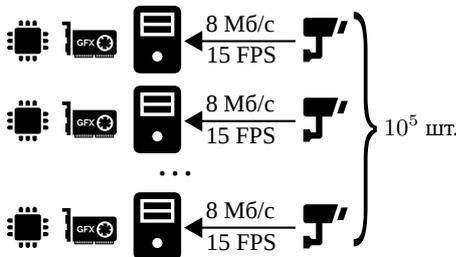
Современные методы сопровождения людей используют подход к сопровождению через обнаружение. Будем рассматривать только алгоритмы способные обрабатывать видеопоток. Их можно классифицировать на 2 группы по временному контексту, используемому для построения результатов на определённом кадре. Первые используют только предыдущие кадры [2], [12], [14], [15], вторые учитывают информацию со следующих кадров, используя метод скользящего окна [1], [3], [6], [11]. Последние позволяют добиться лучшего качества, но результат генерируется с некоторой задержкой.

Вторым ключевым аспектом является детектируемый объект. Наиболее популярным является обнаружение всего человека в кадре [2], [3], [14], [15]. В этом случае перекрытия могут существенно снизить качество результата. Обнаружение голов людей [1], [6], [11] является более стойким к перекрытиям, но детекторы голов обычно производят много ложных обнаружений. Третья группа методов использует детекторы на осно-

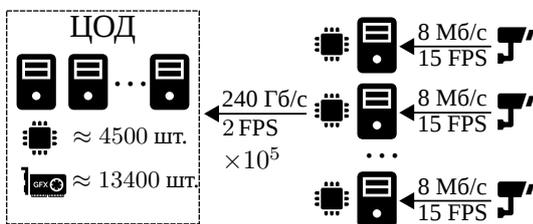
ве деформируемых моделей частей [12] и одновременно учитывает информацию о их положении.

Третьим аспектом является выбор алгоритма объединения обнаружений в траектории. Данную задачу сводят к минимизации функции энергии модели объединения обнаружений. Алгоритмы, обрабатывающие видеопоток, обычно учитывают в функции энергии связи только между соседними обнаружениями траектории. Задачу тогда можно свести к наращиванию траекторий с помощью их сопоставления с обнаружениями со следующего кадра Венгерским алгоритмом [5] (работы [2], [14], [15]), либо приближённо жадными алгоритмами [12]. Для обработки с помощью скользящего окна возможно использование алгоритма на основе модифицированного поиска потока минимальной стоимости [7]. Модель может быть усложнена добавлением типа траектории для выявления ложных обнаружений. В этом случае оптимизация выполняется с помощью метода Монте-Карло [1], [6], [11].

Члены функции энергии могут использовать различные признаки, извлекаемые из видео: нейросетевая реидентификация [14], [15], визуальное сопровождение [1], [6], [11], оптический поток [3]. Некоторые алгоритмы [1], [3], [6], [11] строят *треклеты*, части траектории в коротком отрезке видео, что позволяет строить более точные члены функции энергии для модели. В некоторых работах используется калибровка камеры, что позволяет находить координаты людей в пространстве. В статье [10] представлен метод автоматического получения матрицы калибровки по обнаружениям. Калибровка может быть использована для повышения качества детекции [17].



(а) Обработка каждого видеопотока своим компьютером.



(б) Обработка видеопотоков локально с размещением детекторов в ЦОД.

Рис. 1. Оценочное сравнение конфигураций систем для подсчёта людей.

Современные алгоритмы сопровождения разработаны для работы на одном компьютере с мощным гра-

фическим ускорителем. Он требуется для осуществления детекции с помощью нейронных сетей, что позволяет добиться высокого качества обнаружений. Так как обнаружение необходимо производить для всех кадров видео, а качественные алгоритмы детекции вычислительно ёмкие, потребуется число графических ускорителей равное числу обрабатываемых видеопотоков. Данный подход делает применение системы экономически необоснованным (рис. 1, а)

Учитывая перечисленные выше недостатки, в данной работе предлагается распределённый алгоритм сопровождения для системы, состоящей из подключённых к камерам видеонаблюдения локальных вычислительных узлов и ЦОД с серверами для обнаружения людей (рис. 1, б). На ресурсах ЦОДа выполняется обнаружение людей, но лишь на разреженном множестве кадров. Это позволит уменьшить число используемых графических ускорителей и снизить требование на пропускную способность канала. Чтобы компенсировать отсутствие информации от детектора на остальных кадрах видео, выполняющаяся на локальных узлах часть алгоритма сопровождения будет производить дополнительный анализ видеопотока. Эта часть алгоритма сопровождения должна быть вычислительно простой, чтобы снизить стоимость локальных узлов.

Предположения, используемые для сравнения на рис. 1: один графический ускоритель производит детекцию с частотой 15 Гц; предложенный алгоритм для варианта (б) будет производить детекцию с частотой 2 Гц; объём информации одного пересылаемого детектору кадра — 150 КБайт; один центральный процессор может обеспечить работу детекторов на трёх графических ускорителях.

## 2. ПРЕДЛОЖЕННЫЙ АЛГОРИТМ

Предложенный алгоритм представляет собой модификацию алгоритма [2]. Он состоит из 4-х этапов: 1) детекция людей на ключевых кадрах; 2) построение матрицы стоимостей (энергий) связывания текущего набора траекторий с новыми обнаружениями; 3) поиск оптимального связывания: каждое обнаружение присоединяется к текущей траектории, либо даёт начало новой траектории; 4) поиск пересечений сигнального отрезка. Ниже подробнее описаны этапы, претерпевшие модификации.

### 2.1. Детекция

Детекция производится на *ключевых кадрах* видеопотока:  $F_1, F_{1+step}, F_{1+2step}, \dots$ . Результатом детекции является набор ограничивающих изображения людей в кадре прямоугольников. Для решения этой подзадачи используется детектор на основе подхода FasterRCNN [9] с использованием свёрточных нейронных сетей. Данный этап является единственным, выполняющимся на удалённом сервере.

## 2.2. Построение матрицы стоимостей

Построение матрицы стоимостей в базовом алгоритме производится на основе предсказания положения человека фильтром Калмана и вычислением IOU для предсказанного положения и нового обнаружения. Пары траектории и обнаружения, для которых получившееся значение ниже заданного порога не могут быть сопоставлены. IOU (intersection over union) — функция, вычисляющая отношение площади пересечения прямоугольников к площади их объединения.

В данной работе предлагается внести описанные в следующих подразделах модификации в процедуру вычисления стоимости связывания.

### 2.2.1. Оценка скорости с помощью визуального сопровождения

Для уменьшения требуемого объема вычислительных ресурсов необходимо уменьшать частоту детекции. При больших значениях  $step$  качество работы базового алгоритма сильно ухудшается (см. рис. 3). Это объясняется тем, что адекватную оценку скорости базовый алгоритм получает из фильтра Калмана только при накоплении достаточной статистики. Несколько первых обнаружений траектории должны сопоставиться друг с другом по значению IOU. Увеличение  $step$  приводит к тому, что изображение человека в кадре перемещается между ключевыми кадрами сильно. Следовательно, значение IOU становится околонулевым и сопоставления не происходит.

Предлагается решить данную проблему, оценив скорость человека  $v_x, v_y$  и включив её в видимое состояние фильтра Калмана. Для измерения скорости воспользуемся алгоритмом визуального сопровождения. Он позволит найти изображение человека с кадра  $1 + k \text{ step}$  на кадре  $1 + k \text{ step} + \Delta$ , где  $\Delta \in \mathbb{Z}_+$  — параметр алгоритма. Пусть положение человека, заданное координатами левого верхнего угла и размерами ограничивающего прямоугольника, на первом кадре  $(x, y, w, h)$ , а на втором  $(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{h})$ . Оценим скорости через перемещение центра нижней границы ограничивающего прямоугольника:

$$\begin{aligned} v_x &= \left( (\tilde{x} + \frac{\tilde{w}}{2}) - (x + \frac{w}{2}) \right) \frac{step}{\Delta}, \\ v_y &= \left( (\tilde{y} + \tilde{h}) - (y + h) \right) \frac{step}{\Delta}. \end{aligned} \quad (1)$$

Среди современных методов визуального сопровождения, проанализированных по результатам конкурса VOT2016 [4] и имеющих открытую реализацию, был выбран метод ASMS [13]. Он обладает небольшой вычислительной сложностью, что позволяет осуществлять визуальное сопровождение в режиме реального времени.

Выбранный алгоритм визуального сопровождения локализует изображение человека в кадре не так, как это делает детектор. Это приводит к наличию постоянной ошибки в определении скорости. Предлагается решить данную проблему, применив визуальное

сопровождение к паре кадров с номерами  $1 + k \text{ step}$  и  $1 + k \text{ step} + \varepsilon$ , где  $\varepsilon \in \mathbb{Z}$ ,  $\varepsilon < \Delta$  — параметр алгоритма. Соответствующее предсказание обозначим через  $(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{h})$ . Тогда оценим скорость через:

$$\begin{aligned} v_x &= \left( (\tilde{x} + \frac{\tilde{w}}{2}) - (\bar{x} + \frac{\bar{w}}{2}) \right) \frac{step}{\Delta - \varepsilon}, \\ v_y &= \left( (\tilde{y} + \tilde{h}) - (\bar{y} + \bar{h}) \right) \frac{step}{\Delta - \varepsilon}. \end{aligned} \quad (2)$$

Оба предсказания  $(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{h})$  и  $(\bar{x}, \bar{y}, \bar{w}, \bar{h})$  получены одним алгоритмом визуального сопровождения. Возникающая из-за различий в локализации ошибка таким образом будет устранена.

### 2.2.2. Использование оценки скорости для новых обнаружений

Заметим, что при оценке скорости оценка между кадрами с разницей в номерах  $\Delta - \varepsilon$  экстраполируется на разницу  $step$ , исходя из модели равномерного прямолинейного движения. Траектории людей в кадре могут быть с достаточной точностью описаны данной моделью только на коротких интервалах времени. Также увеличение  $\Delta$  приводит к ошибкам в визуальном сопровождении. Это приводит к тому, что последующее увеличение параметра  $step$  ухудшает качество алгоритма (см. рис. 3).

Данную проблему возможно частично устранить. Для этого будем предсказывать положение человека не на  $step$  кадров вперед, а на  $step/2$ . Оценки скоростей через визуальное сопровождение (2) изменятся на:

$$\begin{aligned} v_x &= \left( (\tilde{x} + \frac{\tilde{w}}{2}) - (\bar{x} + \frac{\bar{w}}{2}) \right) \frac{step/2}{\Delta - \varepsilon}, \\ v_y &= \left( (\tilde{y} + \tilde{h}) - (\bar{y} + \bar{h}) \right) \frac{step/2}{\Delta - \varepsilon}. \end{aligned} \quad (3)$$

Для новых обнаружений на очередной итерации алгоритма будем предсказывать положение на  $step/2$  кадров назад. Для этого достаточно применить визуальное сопровождение к одному из предыдущих кадров. Будем вычислять новую функцию стоимости как IOU между предсказанным фильтром Калмана на  $step/2$  кадров вперед положением траектории и предсказанным на  $step/2$  кадров назад положением обнаружения.

Данный подход позволяет сократить в два раза интервал, на который экстраполируется оценка скорости. Однако теперь алгоритм визуального сопровождения необходимо запускать четыре раза для каждого обнаружения, вместо двух. Число запусков можно сократить до трёх. Для этого нужно использовать 0 в качестве значения параметра  $\varepsilon$ .

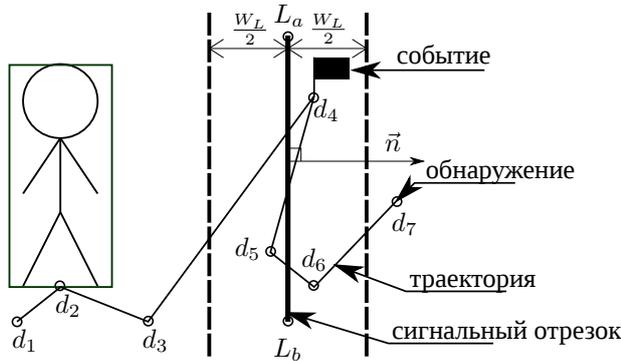
## 2.3. Поиск событий

Данный этап не присутствует в базовом алгоритме, так как он решает задачу сопровождения, а не подсчёта людей. Чтобы получить искомые события, необходимо для каждой траектории найти её пересечения с сигнальным отрезком. Для этого рассматривается ломаная линия, соответствующая нижней середине обнаружений (рис. 2).

Чтобы подавить ложные события вызванные незначительными колебаниями обнаружений траектории, для сигнального отрезка вводится ширина  $W_L$ . Пересечение траекторией сигнального отрезка приводит к генерации события, только если с момента предыдущего события для этой траектории (либо с начала траектории, если не было событий) существует обнаружение, находящиеся на отдалении не менее  $W_L/2$  с соответствующей стороны от прямой, проходящей через сигнальный отрезок.

**2.3.1. Экстраполяция траекторий**

Во время работы алгоритма из-за перекрытий, неточностей детекции или каких-то других ошибок человек может начать сопровождаться после момента пересечения сигнального отрезка. Если траектория началась около сигнального отрезка и продолжилась в сторону от него, разумно предположить, что ранее произошло пересечение сигнального отрезка. Чтобы учесть описанную ситуацию, предлагается произвести предобработку каждой траектории путем её экстраполяции на одно обнаружение в обе стороны. Оценив скорость от второго к первому обнаружению траектории, добавим в начало траектории обнаружение полученное сдвигом первого обнаружения на вектор, соответствующий перемещению с полученной скоростью за число кадров  $\Delta_{ex}$ . Аналогично поступим для последних двух обнаружений траектории, добавив новое обнаружение в конец. Если траектория началась не позднее, чем  $\Delta_{ex}$  кадров с момента пересечения сигнального отрезка человеком, и он двигался с постоянной скоростью, то пропущенное событие будет восстановлено.



**Рис. 2.** Пример результатов работы алгоритма поиска событий. Использование значения ширины сигнального отрезка позволило избежать ложных событий для обнаружений  $d_5$  и  $d_6$ .

**3. ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА**

**3.1. Методика исследования качества**

**3.1.1. Наборы данных**

Для тестирования базового алгоритма и его модификации были использованы наборы с ручной разметкой траекторий людей для задачи сопровождения.

Всего использовалось 22 видеопоследовательности общей продолжительностью 6 часов из коллекции конкурса MOTChallenge [8], собранной ООО «Технологии видеоанализа» коллекции и набора Towncentre [1]. Для каждого набора вручную было установлено положение сигнального отрезка. Далее с помощью алгоритма поиска событий (разд. 2.3) была получена эталонная разметка  $\hat{E}$  для задачи подсчёта людей. Получившееся число эталонных событий во всех видеопоследовательностях — 2657.

**3.1.2. Сопоставление событий**

Среди результирующих событий могут быть верные и ложные. Ложные события не имеют соответствия в эталонной разметке. Будем считать, что событие из выходных данных  $E_i$  соответствует событию в эталонной разметке  $\hat{E}_j$ , если они соответствуют переходу через сигнальный отрезок одного и того же человека в один момент времени. Заметим, что эти события могут быть на разных кадрах из-за разницы локализации изображения человека ограничивающим прямоугольником и из-за того, что предложенный алгоритм может выдать событие только для ключевого кадра.

Для сопоставления событий из выходных данных и эталонной разметки предлагается воспользоваться жадным алгоритмом с отсечением сопоставлений по двум порогам:  $\Delta_f$  и  $IOU_e$ . Первый задаёт допустимую разницу во времени между событиями, второй задаёт допустимую погрешность в определении позиции человека в кадре. Жадный алгоритм также получает на вход разметку для задачи сопровождения с информацией о сопоставлении эталонных событий и траекторий, т.к. событие  $E_i$  по положению необходимо сопоставлять не с положением  $\hat{E}_j$ , а с положением соответствующей  $\hat{E}_j$  траектории в момент времени события  $E_i$ .

**3.1.3. Оценка качества по числу событий**

На практике результаты подсчёта числа людей часто рассматривают в виде числа проходов за определённый отрезок времени, не учитывая пространственное расположение событий в кадре. Предлагается ввести ещё одну характеристику качества, соответствующую такому подходу.

Рассмотрим отрезки видеопоследовательности, содержащие хотя бы  $P$  эталонных событий. Из них рассмотрим те, которые нельзя сузить, не нарушив условие на число эталонных событий. Согласно разд. 3.1.2 сопоставим множества событий  $E$  и  $\hat{E}$ . Для каждого отрезка вычислим: число ложных событий  $FP_i$  как число несопоставленных событий из результата, число пропущенных событий  $FN_i$  как число несопоставленных событий из эталона; общее число эталонных событий на отрезке  $GT_i$ . Вычислим относительную ошибку подсчёта на отрезках:

$$E_i = \frac{|FP_i - FN_i|}{GT_i} \tag{4}$$

Введем характеристику качества  $E$ , как среднюю относительную ошибку на всех рассматриваемых отрезках.

### 3.2. Качество подсчёта людей

Для сопоставления событий (разд. 3.1.2) использовались соответствующее двум секундам видео значение  $\Delta_f$  и  $IOU_e = 0,4$ . Для оценки качества по числу событий (разд. 3.1.3)  $P = 10$ . Относительная ошибка подсчёта  $E$  вычислялась после объединения всех отрезков всех наборов. Модификации добавлялись к базовому алгоритму одна за другой, чтобы оценить их вклад в полученный результат. Полученные результаты приведены на рис. 3.

Можно сделать вывод, что каждая из предложенных модификаций улучшает качество алгоритма. Предложенный алгоритм при частоте детекции 4/3 Гц показывает приемлемое качество и позволяет сократить число используемых графических ускорителей по сравнению с базовым алгоритмом для изображенной на рис. 1, а системы (15 Гц) в 11,25 раза.

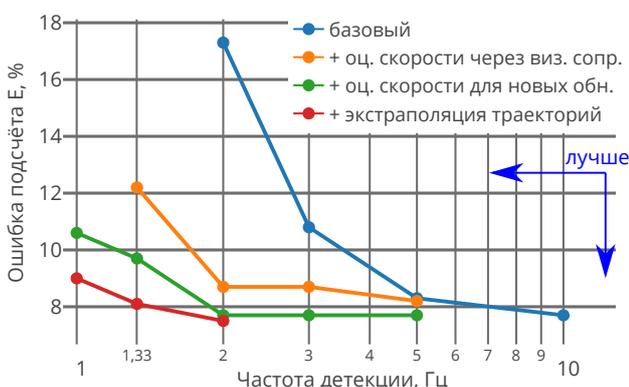


Рис. 3. Результаты экспериментальной оценки.

## 4. ЗАКЛЮЧЕНИЕ

Предложен алгоритм, являющийся модификацией работы [2], способный работать в распределённой системе с детекцией на разреженном видеопотоке и локальной обработкой, заключающейся в визуальном сопровождении найденных обнаружений. Метод позволяет значительно снизить необходимый объём вычислительных ресурсов для задачи подсчёта людей, не уступая в качестве базовому алгоритму.

## 5. БЛАГОДАРНОСТИ

Работа выполнена при поддержке гранта РФФИ 16-29-09612 офи\_м «Исследование и разработка методов биометрической идентификации человека по походке, жестам и комплекции в данных видеонаблюдения».

## 6. ЛИТЕРАТУРА

[1] Ben Benfold и Ian Reid. «Stable Multi-Target Tracking in Real-Time Surveillance Video». В: *CVPR*. Июнь 2011, с. 3457—3464.

- [2] Alex Bewley, ZongYuan Ge, Lionel Ott и др. «Simple Online and Realtime Tracking.» В: *CoRR abs/1602.00763* (2016).
- [3] Wongun Choi. «Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor.» В: *CoRR abs/1504.02340* (2015).
- [4] Matej Kristan, Aleš Leonardis, Jiri Matas и др. *The Visual Object Tracking VOT2016 challenge results*. Springer. Окт. 2016. URL: <http://www.springer.com/gp/book/9783319488806>.
- [5] Harold W Kuhn. «The Hungarian method for the assignment problem». В: *Naval Research Logistics (NRL) 2.1-2* (1955), с. 83—97.
- [6] D. Kuplyakov, E. Shalnov и A. Konushin. «Markov chain Monte Carlo based video tracking algorithm». Англ. В: *Programming and Computer Software 43.4* (2017), с. 224—229. ISSN: 1608-3261; 0361-7688. DOI: 10.1134/S0361768817040053.
- [7] Philip Lenz, Andreas Geiger и Raquel Urtasun. «FollowMe: Efficient Online Min-Cost Flow Tracking With Bounded Memory and Computation». В: *The IEEE International Conference on Computer Vision (ICCV)*. Дек. 2015.
- [8] A. Milan, L. Leal-Taixé, I. Reid и др. «MOT16: A Benchmark for Multi-Object Tracking». В: *arXiv:1603.00831 [cs]* (март 2016). URL: <http://arxiv.org/abs/1603.00831>.
- [9] Shaoqing Ren, Kaiming He, Ross Girshick и др. «Faster r-cnn: Towards real-time object detection with region proposal networks». В: *Advances in neural information processing systems*. 2015, с. 91—99.
- [10] E. V. Shalnov, A. S. Konushin и V. S. Konushin. «Convolutional neural network for camera pose estimation from object detections». Англ. В: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 42.2-W4* (2017), с. 1—6. DOI: 10.5194/isprs-archives-XLII-2-W4-1-2017.
- [11] Eugeny Shalnov, Vadim Konushin и Anton Konushin. «An improvement on an MCMC-based video tracking algorithm». В: *Pattern Recognition and Image Analysis 25* (2015), с. 532—540.
- [12] Guang Shu, Afshin Dehghan, Omar Oreifej и др. «Part-based multiple-person tracking with partial occlusion handling.» В: *CVPR. IEEE Computer Society, 2012*, с. 1815—1821. ISBN: 978-1-4673-1226-4.
- [13] Tomas Vojir, Jana Noskova и Jiri Matas. «Robust scale-adaptive mean-shift for tracking». В: *Scandinavian Conference on Image Analysis*. Springer. 2013, с. 652—663.
- [14] N. Wojke, A. Bewley и D. Paulus. «Simple Online and Realtime Tracking with a Deep Association Metric». В: *ArXiv e-prints* (март 2017).

- [15] Fengwei Yu, Wenbo Li, Quanquan Li и др. «Poi: Multiple object tracking with high performance detection and appearance feature». В: *European Conference on Computer Vision*. Springer. 2016, с. 36—42.
- [16] И. В. Филиппов, В. А. Кононов, В. С. Конушин и др. «Подсчет количества людей в видеопоследовательности на основе детектора головы человека». В: *Программные продукты и системы* 1 (2015), с. 121—126. ISSN: 0236-235X.
- [17] Е. В. Шальнов и А. С. Конушин. «Использование геометрии сцены для увеличения точности детекторов». В: *Программные продукты и системы* 30.1 (2017), с. 106—111. ISSN: 0236-235X. DOI: 10.15827/0236-235X.030.1.106-111.

## ОБ АВТОРАХ

Купляков Денис Анатольевич, магистр факультета вычислительной математики и кибернетики Московского государственного университета им. М.В. Ломоносова. Разработчик программного обеспечения ООО «Технологии видеоанализа». Его e-mail [denis.kuplyakov@graphics.cs.msu.ru](mailto:denis.kuplyakov@graphics.cs.msu.ru)

Шальнов Евгений Вадимович, к.ф.-м.н., сотрудник факультета вычислительной математики и кибернетики Московского государственного университета им. М.В. Ломоносова. Его e-mail [eshalnov@graphics.cs.msu.ru](mailto:eshalnov@graphics.cs.msu.ru)

Конушин Вадим Сергеевич, генеральный директор ООО «Технологии видеоанализа». Его e-mail [vadim@tevia.ru](mailto:vadim@tevia.ru)

Конушин Антон Сергеевич, к.ф.-м.н., доцент Московского государственного университета им. М.В. Ломоносова и Национального исследовательского университета «Высшая школа экономики». Его e-mail [ktosh@graphics.cs.msu.ru](mailto:ktosh@graphics.cs.msu.ru)