

РАСПРЕДЕЛЕННЫЙ АЛГОРИТМ СОПРОВОЖДЕНИЯ
ДЛЯ ПОДСЧЕТА ЛЮДЕЙ В ВИДЕО© 2019 г. Д. А. Купляков^{a,b,*}, Е. В. Шальнов^{a,**}, В. С. Конушин^{b,***}, А. С. Конушин^{a,****}^a Московский государственный университет им. М.В. Ломоносова

Факультет вычислительной математики и кибернетики

119991 Москва, ГСП-1, Ленинские горы, д. 1, стр. 52, Россия

^b ООО «Технологии видеоанализа»

119634 Москва, ул. Скульптора Мухомовой, д. 7, этаж 1, пом. II, ком. 2В, Россия

*e-mail: denis.kuplyakov@graphics.cs.msu.ru

**e-mail: eshalnov@graphics.cs.msu.ru

***e-mail: vadim@tevisan.ru

****e-mail: ktosh@graphics.cs.msu.ru

Поступила в редакцию 07.02.2019 г.

После доработки 18.03.2019 г.

Принята к публикации 18.03.2019 г.

В работе рассматривается задача сопровождения людей в видеопотоке с целью их подсчета. 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 распределенный алгоритм сопровождения. Он позволяет за счет 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 подсчета людей. Экспериментальная оценка показала, что предложенный алгоритм позволяет достичь приемлемого качества при частоте детекции 4/3 Гц.

DOI: 10.1134/S0132347419040046

1. ВВЕДЕНИЕ

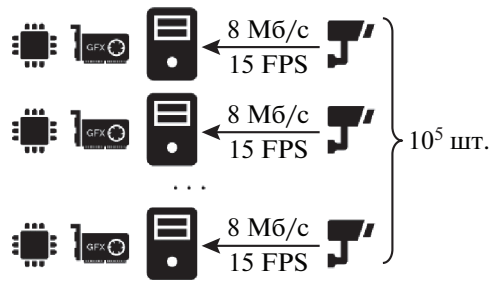
Практически важной задачей является подсчет числа людей, прошедших через определенные зоны общественной инфраструктуры, такие как пешеходные переходы, тротуары, площади и т.д. Одним из возможных путей ее решения является сопровождение людей в видеопотоке, то есть построение траекторий их движения. *Траектория* однозначно соответствует человеку и содержит его положение на всех кадрах, где он представлен. Для подсчета людей обычно на кадре фиксируется *сигнальный отрезок* (рис. 1). При пересечении траекторией человека данного отрезка фиксируется один проход, что позволяет подсчитывать людей.

В работе рассматривается задача полностью автоматического подсчета людей в видео, снятом 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 сигналь-

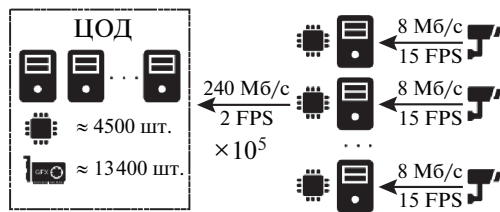
ного отрезка $\{E_i\}_{i=1}$. Событие $E_i = (k_i, r_i, d_i)$ описывается номером кадра k_i , на котором произошло пересечение сигнального отрезка, прямоугольником d_i , ограничивающим изображение человека на кадре в момент пересечения, и направлением



Рис. 1. Пример сигнального отрезка для оценки загрузки пешеходного перехода.



(а) Обработка каждого видеопотока своим компьютером



(б) Обработка видеопотоков локально с размещением детекторов в ЦОД

Рис. 2. Оценочное сравнение конфигураций систем для подсчёта людей.

пересечения сигнального отрезка r_i . Данная задача рассматривалась также в работе [1].

Современные методы сопровождения людей используют подход к сопровождению через обнаружение. Будем рассматривать только алгоритмы способные обрабатывать видеопоток. Их можно классифицировать на две группы по временному контексту, используемому для построения результатов на определенном кадре. Первые используют только предыдущие кадры [2–5], вторые учитывают информацию со следующих кадров, используя метод скользящего окна [6–9]. 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 [2–4, 9]. В этом случае перекрытия могут существенно снизить качество результата. Обнаружение голов людей [6–8] является более стойким к перекрытиям, но детекторы голов обычно производят много ложных обнаружений. Третья группа методов использует детекторы на основе деформируемых моделей частей [5] 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 [10] (работы [2–4]), либо приближенно жадными алгоритмами [5]. Для обработки с помощью скользящего окна возможно использование алгоритма на основе модифицированного поиска потока минимальной стоимости [11]. Модель может быть усложнена добавлением типа траектории для выявления ложных обнаружений. В этом случае оптимизация выполняется с помощью метода Монте-Карло [6–8].

Члены функции энергии могут использовать различные признаки, извлекаемые из видео: нейросетевая реидентификация [3, 4], визуальное сопровождение [6–8], оптический поток [9]. Некоторые алгоритмы [6–9] 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 [12] представлен метод автоматического получения матрицы калибровки по обнаружениям. Калибровка может быть использована для повышения качества детекции [13].

Современные алгоритмы сопровождения разработаны для работы на одном компьютере с мощным графическим 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 емкие, потребуется число графических ускорителей равное числу обрабатываемых видеопотоков. Данный подход делает применение системы экономически необоснованным (рис. 2, а).

Учитывая перечисленные выше недостатки, в данной работе предлагается распределенный алгоритм сопровождения для системы, состоящей из подключенных к камерам видеонаблюдения локальных вычислительных узлов и ЦОД с серверами для обнаружения людей (рис. 2, 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.

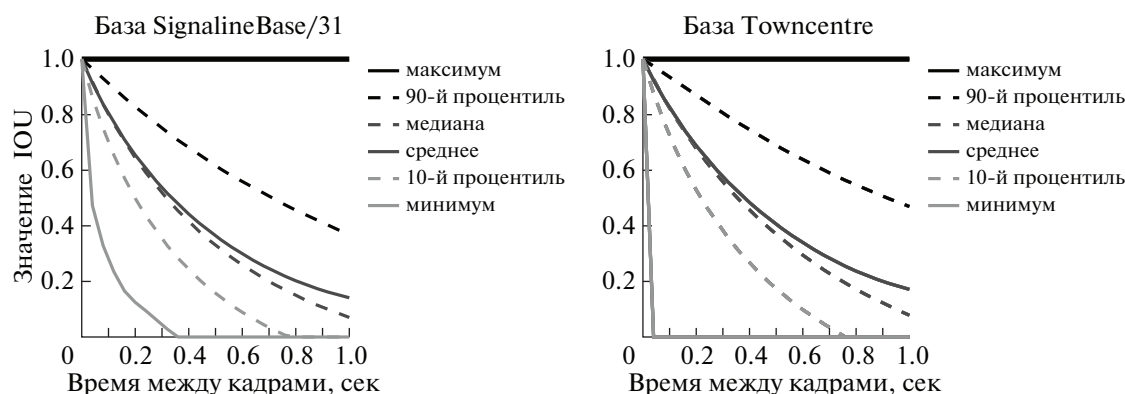


Рис. 3. Статистика значения IOU для двух обнаружений одной траектории из ручной разметки в зависимости от времени между кадрами.

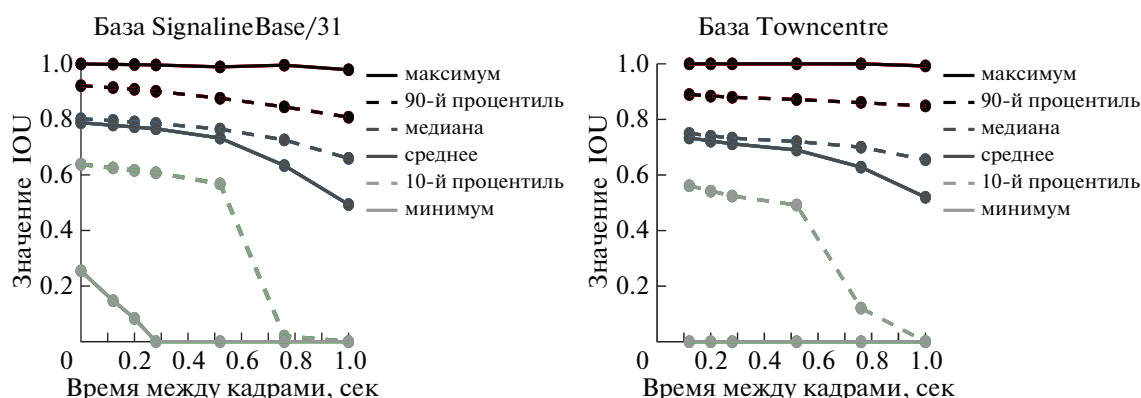


Рис. 4. Статистика значения IOU для предсказанного с помощью алгоритма визуального сопровождения ASMS и истинного обнаружений. В качестве исходных брались обнаружения из ручной разметки.

Предположения, используемые для сравнения на рис. 2: один графический ускоритель производит детекцию с частотой 15 Гц; предложенный алгоритм для варианта (б) будет производить детекцию с частотой 2 Гц; объем информации одного пересылаемого детектору кадра — 150 КБайт; один центральный процессор может обеспечить работу детекторов на трех 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 дает 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, ограничивающий изображения людей в кадре прямоугольниками. Для решения этой подзадачи используется детектор на основе подхода FasterRCNN [14] с использованием сверточных нейронных сетей. Данный этап является единственным, выполняющимся на удаленном 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 полученное значение ниже заданного порога не могут быть сопоставлены. *IOU* (intersection over union) — функция, вычисляющая отношение площади пересечения прямоугольников к площади их объединения.



Рис. 5. Пример, демонстрирующий разницу в локализации человека в кадре детектором и алгоритмом визуального сопровождения. Черным цветом нарисовано обнаружение детектора, белым — ограничивающий прямоугольник, полученный алгоритмом визуального сопровождения для того же обнаружения на том же кадре.

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. 7). 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 (рис. 3.). Следовательно, значение *IOU* становится околонулевым и сопоставления не происходит.

Предлагается решить данную проблему, оценив скорость человека v_x, v_y и включив ее 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:

$$\begin{aligned} v_x &= \left(\left(\tilde{x} + \frac{\tilde{w}}{2} \right) - \left(x + \frac{w}{2} \right) \right) \frac{\text{step}}{\Delta}, \\ v_y &= \left((\tilde{y} + \tilde{h}) - (y + h) \right) \frac{\text{step}}{\Delta}. \end{aligned} \quad (2.1)$$

Среди современных методов визуального сопровождения, проанализированных по результатам конкурса VOT2016 [15] и имеющих открытую реализацию, был выбран метод ASMS [16]. Он обладает небольшой вычислительной сложностью, что позволяет осуществлять визуальное сопровождение в режиме реального времени.

Следует отметить, что значение Δ не следует выбирать очень большим, так как алгоритмы визуального сопровождения рассчитаны на работу с соседними кадрами видеопотока. При увеличении разницы во времени между кадрами ошибка решения задачи визуального сопровождения возрастает (рис. 4).

Выбранный алгоритм визуального сопровождения локализует изображение человека в кадре не так, как это делает детектор (рис. 5). 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 $1 + k \text{ step} + \epsilon$, где $\epsilon \in \mathbb{Z}$, $\epsilon < \Delta$ — параметр алгоритма. Соответствующее предсказание обозначим через $(\bar{x}, \bar{y}, \bar{w}, \bar{h})$. Тогда оценим скорость через:

$$\begin{aligned} v_x &= \left(\left(\tilde{x} + \frac{\tilde{w}}{2} \right) - \left(\bar{x} + \frac{\bar{w}}{2} \right) \right) \frac{\text{step}}{\Delta - \epsilon}, \\ v_y &= \left((\tilde{y} + \tilde{h}) - (\bar{y} + \bar{h}) \right) \frac{\text{step}}{\Delta - \epsilon}. \end{aligned} \quad (2.2)$$

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 $\Delta - \epsilon$ 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 Δ приводит к ошибкам в визуальном сопровождении. Это приводит к тому, что последующее увеличение параметра *step* ухудшает качество алгоритма (см. рис. 7).

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 (2.2) изменятся на:

$$\begin{aligned} v_x &= \left(\left(\tilde{x} + \frac{\tilde{w}}{2} \right) - \left(\bar{x} + \frac{\bar{w}}{2} \right) \right) \frac{\text{step}/2}{\Delta - \epsilon}, \\ v_y &= \left((\tilde{y} + \tilde{h}) - (\bar{y} + \bar{h}) \right) \frac{\text{step}/2}{\Delta - \epsilon}. \end{aligned} \quad (2.3)$$

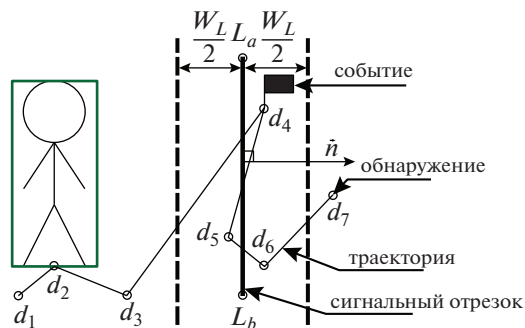


Рис. 6. Пример результатов работы алгоритма поиска событий. Использование значения ширины сигнального отрезка позволило избежать ложных событий для обнаружений d_5 и d_6 .

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 трех. Для этого нужно использовать 0 в качестве значения параметра ϵ . Как видно на рис. 5, алгоритм визуального сопровождения производит локализацию переданного ему ограничивающего прямоугольника, даже если в качестве второго кадра подать исходный. Использование нулевого значения ϵ также должно сделать оценку скорости более точной, так как для оценки будет использоваться кадр, содержащий исходное обнаружение.

2.3. Поиск событий

Данный этап не присутствует в базовом алгоритме, так как он решает задачу сопровождения, а не подсчета людей. Чтобы получить искомые события, необходимо для каждой траектории найти ее пересечения с сигнальным отрезком. Для этого рассматривается ломанная линия, соответствующая нижней середине обнаружений (рис. 6).

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 мо-

Ошибка подсчета E , %

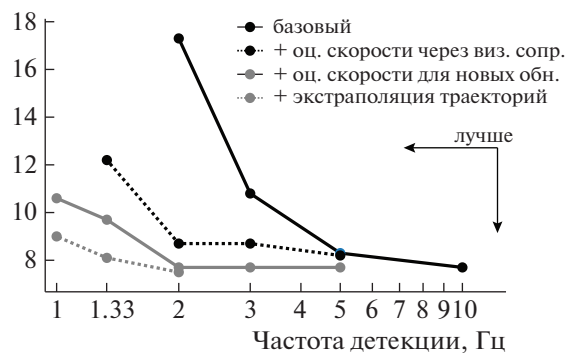


Рис. 7. Зависимость средней относительной ошибки подсчета от частоты детекции.

мента предыдущего события для этой траектории (либо с начала траектории, если не было событий) существует обнаружение, находящееся на расстоянии 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 ее 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 Δ_{ex} . Аналогично поступим для последних двух обнаружений траектории, добавив новое обнаружение в конец. Если траектория началась не позднее, чем Δ_{ex} кадров с момента пересечения сигнального отрезка человеком, и он двигался с постоянной скоростью, то пропущенное событие будет восстановлено.

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 видео последовательности общей продолжительностью 6 часов из коллекции конкурса MOTChallenge [17], собранной ООО “Технологии

Таблица 1. Экспериментальное сравнение базового алгоритма и его модификаций. В герцах указана частота ключевых кадров. Обозначения: $P(\uparrow)$ — *Precision*, точность; $R(\uparrow)$ — *Recall*, полнота; $E(\downarrow)$ — средняя относительная ошибка подсчета.

	Все кадры			5 Гц			3 Гц		
	$P, \%$	$R, \%$	$E, \%$	$P, \%$	$R, \%$	$E, \%$	$P, \%$	$R, \%$	$E, \%$
базовый	93	91	7.7	94	90	8.3	94	81	10.8
+ оц. скорости через виз. сопр.	—	—	—	91	91	8.2	93	90	8.7
+ оц. скорости для новых обн.	—	—	—	93	90	7.7	93	90	7.7
+ экстраполяция траекторий	—	—	—	—	—	—	—	—	—
	2 Гц			4/3 Гц			1 Гц		
	$P, \%$	$R, \%$	$E, \%$	$P, \%$	$R, \%$	$E, \%$	$P, \%$	$R, \%$	$E, \%$
базовый	90	76	17.3	—	—	—	—	—	—
+ оц. скорости через виз. сопр.	93	90	8.7	93	85	12.2	—	—	—
+ оц. скорости для новых обн.	93	90	7.7	93	87	9.7	92	85	10.6
+ экстраполяция траекторий	91	90	7.5	90	89	8.1	90	88	9

видеоанализа” коллекции и набора Towncentre [6]. 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 подсчета людей. Получившееся число эталонных событий во всех видеопоследовательностях — 2657.

3.1.2. Сопоставление событий. Среди результатов 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 (алг. 1) с отсечением сопоставлений по двум порогам: Δ_f и IOU_e . Первый задает допустимую разницу во времени между событиями, второй задает 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 положе-

нием соответствующей $\hat{E}_j$ траектории в момент времени события E_i .

Получив сопоставление событий, возможно вычислить следующие характеристики качества решения задачи подсчета числа людей:

- $FP(\downarrow)$ — число несопоставленных событий из результатов E ;
- $FN(\downarrow)$ — число несопоставленных событий из эталонной разметки $\hat{E}$;

$Precision(\uparrow) = \frac{|E| - FP}{|E|} \in [0; 1]$ — точность определения событий;

• $Recall(\uparrow) = \frac{|E| - FN}{|\hat{E}|} \in [0; 1]$ — полнота определения событий.

$\uparrow$ означает, что чем выше соответствующая характеристика, тем лучше. $\downarrow$ — наоборот, чем меньше, тем лучше.

3.1.3. Оценка качества по числу событий. На практике результаты подсчета числа людей часто рассматривают в виде числа проходов за определенный отрезок времени, не учитывая пространственное расположение событий в кадре. Предлагается ввести еще 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 подсчета на отрезках:

$$\mathbb{E}_i = \frac{|FP_i - FN_i|}{GT_i}. \quad (3.1)$$

Введем характеристику качества $\mathbb{E}$, как среднюю относительную ошибку на всех рассматриваемых отрезках.

Алгоритм 1: Алгоритм сопоставления событий

Вход:

- События $E = \{(k_i, r_i, d_i)\}_{i=1}^{|E|}$, заданные номером кадра, направлением пересечения сигнального отрезка и прямоугольником обнаружения;
- Эталонные траектории $\hat{T} = \{\hat{T}_i\}_{i=1}^{|\hat{T}|}$, $\hat{T}_i = \{(f_{i,j}, \hat{d}_{i,j})\}_{j=1}^{|\hat{T}_i|}$, заданные как множества пар из номера кадра и обнаружения;
- Эталонные связанные с траекториями события $\hat{E} = \{(\hat{k}_i, \hat{r}_i, \hat{d}_i)\}_{i=1}^{|\hat{E}|}$, где $\hat{d}_i$ — номер соответствующей траектории из $\hat{T}$;
- Порог на разницу между номерами кадров Δ_f ;
- Порог по значению IOU IOU_e .

Выход: Набор пар $\{(p_i, q_i)\}_{i=1}^M$, т.ч. событие E_{p_i} соответствует событию $\hat{E}_{q_i}$.

```

1  $R \leftarrow \emptyset$ 
2 foreach  $E_i \in E$  do
3    $found \leftarrow 0$ 
4   for  $f \leftarrow k_i - \Delta_f$  to  $k_i + \Delta_f$  do
5     foreach  $\hat{E}_j$ , такого что  $\hat{k}_j = f_i$  and  $\hat{r}_j = r$  do
6        $c \leftarrow \operatorname{argmin}_u |\hat{d}_{\hat{n}_j, u} - k_i|$ 
7       if  $IOU(d_i, \hat{d}_{\hat{n}_j, c}) \geq IOU_e$  then
8          $found \leftarrow 1$ 
9          $R \leftarrow R \cup \{(i, j)\}$ 
10        break
11      end
12    end
13    if  $found = 1$  then
14      break
15    end
16  end
17 end
18 return  $R$ 

```

3.2. Качество подсчета людей

Для сопоставления событий (разд. 3.1.2) использовалось соответствующее двум секундам видео значение Δ_f и $IOU_e = 0,4$. Для оценки качества по числу событий (разд. 3.1.3) $P = 10$. Значения *Precision*, *Recall* вычислялись после сум-

мирования значений FP и FN для всех наборов. Относительная ошибка подсчета $\mathbb{E}$ 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 в таблице 1 и на рис. 7.

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. 2, а системы (15 Гц) в 11.25 раза.

4. ЗАКЛЮЧЕНИЕ

Предложен алгоритм, являющийся модификацией работы [2], способный работать в распределенной системе с детекцией на разреженном видеопотоке и локальной обработкой, заключающейся в визуальном сопровождении найденных обнаружений. Метод позволяет значительно снизить необходимый объем вычислительных ресурсов для задачи подсчета 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”.

СПИСОК ЛИТЕРАТУРЫ

1. Подсчет количества людей в видеопоследовательности на основе детектора головы человека / Филипов И.В. [и др.] // Программные продукты и системы. 2015. № 1. С. 121–126.
2. *Bewley A., Ge Z., Ott L., Ramos F., Upcroft B.* Simple Online and Realtime Tracking // Computer Vision and Pattern Recognition 2016. arXiv:1602.00763 [cs.CV].
3. *Fengwei Y., Wenbo L., Quanquan L., Yu L., Xiaohua S., Junjie Y.* Poi: Multiple object tracking with high performance detection and appearance feature // European Conference on Computer Vision. Springer. 2016. P. 36–42.
4. *Wojke N., Bewley A., Paulus D.* Simple Online and Realtime Tracking with a Deep Association Metric // Computer Vision and Pattern Recognition. 2017. arXiv:1703.07402 [cs.CV].
5. Part-based multiple-person tracking with partial occlusion handling / G. Shu [и др.] // IEEE Conference on Computer Vision and Pattern Recognition. 2012. P. 1815–1821.
6. *Benfold B., Reid I.* Stable Multi-Target Tracking in Real-Time Surveillance Video // IEEE Conference on Computer Vision and Pattern Recognition. 2011. P. 3457–3464.
7. *Shalnov E., Konushin V., Konushin A.* An improvement on an MCMC-based video tracking algorithm // Pattern Recognition and Image Analysis. 2015. V. 25. P. 532–540.
8. *Kuplyakov D., Shalnov E., Konushin A.* Markov chain Monte Carlo based video tracking algorithm // Programming and Computer Software. United States. 2017. V. 43. № 4. P. 224–229.
9. *Choi W.* Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor // Computer Vision and Pattern Recognition. 2015. arXiv:1504.02340 [cs.CV].
10. *Kuhn H. W.* The Hungarian method for the assignment problem // Naval Research Logistics (NRL). 1955. V. 2. № 1/2. P. 83–97.
11. *Lenz P., Geiger A., Urtasun R.* FollowMe: Efficient Online Min-Cost Flow Tracking With Bounded Memory and Computation // The IEEE International Conference on Computer Vision (ICCV). December, 2015.
12. *Shalnov E.V., Konushin A.S., Konushin V.S.* Convolutional neural network for camera pose estimation from object detections // ISPRS – International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2017. V. 42. P. 1–6.
13. *Шальнов Е.В., Конушин А.С.* Использование геометрии сцены для увеличения точности детекторов // Программные продукты и системы. 2017. Т. 30. № 1. С. 106–111.
14. Faster r-cnn: Towards real-time object detection with region proposal networks / S. Ren [и др.] // Advances in neural information processing systems. 2015. P. 91–99.
15. The Visual Object Tracking VOT2016 challenge results / M. Kristan [и др.]. December, 2016. <http://www.springer.com/gp/book/9783319488806>. Springer.
16. *Vojir T., Noskova J., Matas J.* Robust scaleadaptive mean-shift for tracking // Scandinavian Conference on Image Analysis. Springer. 2013. P. 652–663.
17. MOT16: A Benchmark for Multi- Object Tracking / A. Milan [и др.] // arXiv:1603.00831 [cs]. March, 2016. <http://arxiv.org/abs/1603.00831>.