

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ, РЕЧИ, ТЕКСТА И РАСПОЗНАВАНИЕ ОБРАЗОВ

SIGNAL, IMAGE, SPEECH, TEXT PROCESSING AND PATTERN RECOGNITION



УДК 004.932
<https://doi.org/10.37661/1816-0301-2021-18-2-83-97>

Оригинальная статья
Original Paper

Автоматическое обнаружение и отслеживание движущихся объектов, наблюдаемых видеокамерой беспилотного летательного аппарата

Р. С. Жук

*Объединенный институт проблем информатики
Национальной академии наук Беларуси,
ул. Сурганова, 6, Минск, 220012, Беларусь
E-mail: ramanzhuck@gmail.com*

Аннотация. Предлагается алгоритм автоматического обнаружения и отслеживания движущихся объектов, предназначенный для использования на борту беспилотного летательного аппарата. Разработанный алгоритм основан на отслеживании на протяжении определенного времени выбранных точек изображения. Отслеживаемые точки выбираются из областей на текущем кадре, в которых интенсивность пикселей отличается от интенсивностей тех же пикселей в предыдущих кадрах, совмещенных с текущим кадром при помощи проективного преобразования. Если на нескольких соседних кадрах не фиксируется смещение отслеживаемых точек, они удаляются и на их место добавляются новые точки из областей, предположительно принадлежащих движущимся объектам на текущем кадре. На каждом кадре близкие по расположению и форме траекторий движения точки объединяются в группы, которые соответствуют движущимся объектам. Отслеживание объектов осуществляется путем сопоставления групп движущихся точек соседних кадров. Группы движущихся точек соседних кадров сопоставляются, когда они содержат большое число общих отслеживаемых точек. Алгоритм позволяет одновременно отслеживать более 20 объектов в реальном времени. Индикация объекта как движущегося происходит только в том случае, если за время его сопровождения он сместился на значительное расстояние. Рассматриваемый алгоритм имеет низкий процент ложных обнаружений объектов, хорошо обнаруживает объекты малого размера и надежно сопровождает объекты.

Ключевые слова: обнаружение движения, сопровождение объектов, беспилотные летательные аппараты, движущаяся камера, оптический поток

Для цитирования. Жук, Р. С. Автоматическое обнаружение и отслеживание движущихся объектов, наблюдаемых видеокамерой беспилотного летательного аппарата / Р. С. Жук // Информатика. – 2021. – Т. 18, № 2. – С. 83–97. <https://doi.org/10.37661/1816-0301-2021-18-2-83-97>

Конфликт интересов. Автор заявляет об отсутствии конфликта интересов.

Поступила в редакцию | Received 03.06.2021
Подписана в печать | Accepted 07.06.2021
Опубликована | Published 26.06.2021

Automatic detection and tracking the moving objects observed by an unmanned aerial vehicles video camera

Raman S. Zhuk

*The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus
E-mail: ramanzhuck@gmail.com*

Abstract. An algorithm of automatic detection and tracking the moving objects for the use in equipment on board of unmanned aerial vehicles is considered. The developed algorithm is based on a tracking specially selected points for a certain period. Tracked points are selected from the areas on the current frame, where the pixel intensity differs from the intensities of the same pixels in previous frames, aligned with the current frame using projective transformation. If the displacement of the tracked points is not fixed on several adjacent frames, they are being deleted, and new points from the areas presumably belonging to moving objects in the current frame are added instead. On each frame the points similar by the location and shape of trajectories of movement are combined into groups that presumably correspond to moving objects. Objects are tracked by comparing the groups of moving points with the points of neighboring frames. Groups of moving points from neighboring frames are matched if they contain a large number of common tracked points. The algorithm allows simultaneous tracking of more than 20 objects in real time. The indication of objects as moving occurs only if during the time of its tracking it has shifted a considerable distance. The algorithm has a low percentage of false detections of moving objects, it detects well small objects and is capable reliably to accompany moving objects.

Keywords: motion detection, objects tracking, unmanned aerial vehicles, moving camera, optical flow

For citation. Zhuk R. S. Automatic detection and tracking the moving objects observed by an unmanned aerial vehicles video camera. *Informatics*, 2021, vol. 18, no. 2, pp. 83–97 (In Russ.). <https://doi.org/10.37661/1816-0301-2021-18-2-83-97>

Conflict of interest. The author declare of no conflict of interest.

Введение. В настоящее время в связи со стремительным развитием и широкой востребованностью систем беспилотных летательных аппаратов (БПЛА), одним из основных применений которых является визуальное наблюдение, актуальны задачи обнаружения и отслеживания объектов, наблюдаемых бортовой видеокамерой. Важным классом отслеживаемых объектов являются движущиеся объекты. В последние годы опубликовано большое число работ, посвященных алгоритмам отслеживания объектов данного класса. Очевидно, что для решения задачи отслеживания движущихся объектов, наблюдаемых видеокамерой, требуется сначала автоматически их обнаружить. Поэтому задача обнаружения движущихся объектов может рассматриваться как часть задачи отслеживания. Следовательно, задача обнаружения и отслеживания движущихся объектов, наблюдаемых бортовой видеокамерой, может быть сформулирована таким образом: имеется видеопоследовательность, получаемая бортовой камерой малого БПЛА; на кадрах этой видеопоследовательности требуется обнаружить все вновь появившиеся движущиеся объекты и отследить все движущиеся объекты, обнаруженные ранее. Решение этой задачи требует разработки новых алгоритмов, учитывающих ее специфику, которая заключается в следующем: задача должна решаться на борту малых БПЛА, оборудованных маломощными вычислителями с ARM-архитектурой (такими, как Raspberry Pi), что делает невозможным использование мощных нейросетевых подходов; малые БПЛА часто оборудуются панхроматическими камерами (без цветовых признаков); камеры перемещаются относительно земной поверхности; объекты, присутствующие на кадрах, часто бывают малоразмерными.

Описание основных подходов и обзор последних результатов обнаружения движущихся объектов представлены в работе [1]. Можно выделить три основных метода решения задачи обнаружения движения: вычитание фона, локальную разность кадров после компенсации смещения камеры (для случая движущейся камеры) и оптический поток. Недостатки этих подхо-

дов не позволяют применять их без существенных модификаций на борту БПЛА. Так, метод вычитания фона требует предварительного долгого наблюдения за одной и той же сценой перед его применением для обнаружения движения в этой сцене. Он хорошо подходит для систем наблюдения с неподвижными (либо закрепленными на неподвижных объектах) камерами. Для случая с подвижной камерой требуется построение панорамного снимка, что представляет собой трудоемкую задачу. Поскольку БПЛА может следовать на довольно высокой скорости по незнакомому маршруту, метод вычитания фона на его борту практически неприменим. Метод компенсации движения основан на локальной разности интенсивностей пикселей текущего кадра и недалеких от него по времени предыдущих кадров, предварительно преобразованных так, чтобы скомпенсировать движение точек фона. Недостатком этого метода является большое число ошибок, связанных с неточностями при компенсации движения точек фона. Чаще всего для решения задачи компенсации движения фона используется проективное преобразование (гомография), которое совмещает точки кадров в предположении, что сцена является плоской. Такой подход хорошо подходит для применения на борту БПЛА, особенно летящего на большой высоте, однако ошибки компенсации движения фона все равно будут присутствовать. Метод, основанный на оптическом потоке, находит смещение всех (либо расположенных на равномерной решетке) точек кадра при помощи одного из алгоритмов вычисления оптического потока, после чего найденное поле смещений анализируется и выделяются области, принадлежащие движущимся объектам. Движущиеся объекты на кадрах видеопоследовательностей, снятых с БПЛА, могут быть небольшого размера, поэтому необходимо искать оптический поток для большого числа точек, что требует слишком значительных вычислительных затрат.

Кроме упомянутых классических методов, в настоящее время в компьютерном зрении широко используются подходы, основанные на применении сверточных нейронных сетей (англ. convolutional neural network, CNN). Алгоритмы, основанные на CNN, показывают наилучшие результаты в задачах обнаружения и сопровождения объектов. Недостатком применения CNN является их высокая вычислительная сложность, требующая использования дорогостоящих мощных вычислителей с графическими процессорами, поддерживающими технологию параллельного программирования CUDA.

Таким образом, из всех рассмотренных методов решения поставленной задачи наиболее приемлемыми (ввиду их высокой производительности) являются подходы, основанные на компенсации движения. Вместе с тем у них есть серьезный недостаток – большое количество ложных обнаружений.

Отслеживание объектов на кадрах видеопоследовательности является еще одной хорошо изученной задачей машинного зрения. Классификация современных алгоритмов отслеживания объектов приведена в работе [2]. Обычно алгоритмы отслеживания объектов разрабатываются с целью сопровождения одного объекта и имеют высокую вычислительную сложность.

Исследование проводилось с целью разработки нового алгоритма, способного в реальном времени обнаруживать и отслеживать движущиеся объекты небольшого размера, видимые на движущейся камере, с низким процентом ложных обнаружений.

Одновременное обнаружение и отслеживание движущихся объектов. Предлагаемый в настоящей статье подход к обнаружению движения использует основные идеи, характерные для методов, основанных на компенсации движения и оптическом потоке. Вначале при помощи метода компенсации движения находятся области, которые, возможно, принадлежат движущимся объектам. После этого в найденных областях выбираются точки, которые в дальнейшем отслеживаются при помощи оптического потока. Похожий подход применялся в работе [3], где области возможного движения сначала оцениваются при помощи CNN, а затем при помощи оптического потока анализируется величина их смещения. Предлагаемый алгоритм не анализирует области возможного движения при помощи нейронной сети, поскольку вид этих областей может зависеть от внешнего вида как движущихся, так и неподвижных объектов, скорости и вида их движения, скорости и вида движения камеры, из-за чего обучение сети для универсального применения представляется большой проблемой. Вместо этого осуществляется отслеживание выбранных точек на протяжении долгого времени. Если на нескольких соседних кадрах не фиксируется смещение отслеживаемых точек, они удаляются. На каждом кадре близ-

кие по расположению и форме траекторий движения точки объединяются в группы, которые предположительно принадлежат движущимся объектам. Группы движущихся точек с соседних кадров сопоставляются в том случае, если они содержат большое число общих отслеживаемых точек, благодаря чему осуществляется отслеживание объектов, которые предположительно являются движущимися. Итоговым критерием того, что отслеживаемый объект движущийся, будет его смещение относительно фона за время отслеживания на заданную длину в пикселах, которая является параметром алгоритма. В результате такого подхода удалось получить низкий процент ложных обнаружений движущихся объектов (ниже, чем представленный в статье [2]).

Таким образом, разработанный алгоритм решает сразу несколько сложных задач компьютерного зрения. Он был протестирован на видеопоследовательностях, снятых с различных БПЛА, и показал свою применимость для решения задачи обнаружения движущихся объектов и их сопровождения.

Алгоритм обнаружения и отслеживания движущихся объектов. На кадрах видеопоследовательности, снимаемой бортовой камерой БПЛА, требуется в автоматическом режиме найти движущиеся объекты в режиме реального времени или близком к нему. В случае остановки обнаруженного движущегося объекта его отслеживание прекращается до того момента, пока он снова не начнет движение. Каждый движущийся объект выделяется ограничивающим прямоугольником и соответствующим ему номером, который свидетельствует о том, что сопровождается тот же самый объект.

В основе представленного алгоритма лежит идея отслеживания некоторых специально выбираемых точек, количество которых n строго фиксировано. На текущем кадре точки выбираются из областей, в которых интенсивность пикселей отличается от интенсивностей тех же пикселей на предыдущих кадрах, совмещенных с текущим кадром при помощи проективных преобразований. Далее выбранные точки отслеживаются на следующих кадрах с помощью алгоритма нахождения разреженного оптического потока Лукаса – Канаде [4] (известного метода компьютерного зрения, позволяющего найти соответствия между точками на двух последовательных кадрах видеопоследовательности).

Пусть $\mathbf{I} = \{I_1, \dots, I_k, \dots\}$ – последовательность кадров, на которых необходимо обнаруживать и отслеживать движущиеся объекты; $\mathbf{A} = \{P_1, \dots, P_n\}$ – массивы положений отслеживаемых точек на кадрах из \mathbf{I} , т. е. $P_i = \{p_i^{k-q_i}, \dots, p_i^{k-1}, p_i^k\}$ на кадре I_k представляет собой массив из положений отслеживаемой точки на всех кадрах с начала ее отслеживания и до текущего момента. Обозначим $\mathbf{P}^k = \{p_1^k, \dots, p_n^k\}$ положения отслеживаемых точек на кадре I_k , а $\Theta^k = \{S_1^k, \dots, S_n^k\}$ – массив траекторий отслеживаемых точек на кадре I_k , где каждая траектория $S_i^k = \{s_i^{k-q_i, k}, \dots, s_i^{k-1, k}, s_i^{k, k}\}$ отслеживаемой точки представляет собой массив точек на кадре I_k , которые соответствуют положениям P_i отслеживаемой точки на предыдущих кадрах. В S_i^k точка $s_i^{k, k} = p_i^k$, а каждая точка $s_i^{k-j, k}$ находится по формуле $s_i^{k-j, k} = H_{k-1}^k \circ H_{k-2}^{k-1} \circ \dots \circ H_{k-j}^{k-j+1}(p_i^{k-j})$, где H_{q-1}^q – проективное преобразование, которое переводит точки кадра I_{q-1} на кадр I_q в предположении, что они не движутся. На практике точка $s_i^{k, k-j} \in S_i^k$ кадра I_k всегда находится на основании точки $s_i^{k-j, k-1} \in S_i^{k-1}$ кадра I_{k-1} по формуле $s_i^{k, k-j} = H_{k-1}^k(s_i^{k-j, k-1})$. Пусть далее $\Theta^k = \{S_1^k, \dots, S_n^k\}$ – массив траекторий отслеживаемых точек на кадре I_k . На рис. 1 показаны траектории отслеживаемых точек на одном из кадров.

На каждом кадре близкие по расположению и форме траекторий отслеживаемые точки объединяются в группы, которые предположительно соответствуют движущимся объектам. Обозначим все найденные в ходе работы алгоритма движущиеся объекты как $\mathbf{O} = \{O_1, \dots, O_l, \dots\}$. Если движущийся объект O_l присутствует на текущем кадре I_k , то он характеризуется цен-

тральной точкой C_i^k , вектором скорости V_i^k , траекторией $T_i^k = \{C_i^{k-m,k}, \dots, C_i^{k-1,k}, C_i^{k,k}\}$, ограничивающим прямоугольником R_i^k , принадлежащими ему положениями отслеживаемых точек $Q_i^k \subset P^k$, а также нормированной гистограммой цветов $h_i^k = \{h_1^k, \dots, h_{64}^k\}$ точек из Q_i^k .

Пусть I_k – текущий кадр видеопоследовательности \mathbf{I} . Задачей разработанного алгоритма является нахождение на нем движущихся объектов. Кроме этого, данный алгоритм устанавливает соответствие между движущимися объектами, обнаруженными на текущем кадре I_k , и объектами, которые ранее были обнаружены на предыдущих кадрах видеопотока, благодаря чему отслеживаются движущиеся объекты на кадрах видеопоследовательности.

Ниже описываются 11 шагов разработанного алгоритма, с помощью которых отслеживаются движущиеся объекты на всех кадрах видеопоследовательности, кроме двух стартовых. После этого приводятся действия, которые осуществляются на стартовых кадрах видеопоследовательности.

Первым шагом алгоритма является нахождение проективного преобразования H_{k-1}^k между I_{k-1} и I_k , которое переводит точки кадра I_{k-1} на кадр I_k в предположении, что они не движутся. Для нахождения H_{k-1}^k необходимо сначала установить соответствие между некоторыми точками (желательно, не принадлежащими движущимся объектам) кадров I_{k-1} и I_k . Для этого берутся точки кадра I_{k-1} , расположенные на равномерной сетке с шагом 50 пикселей, и ищутся точки, соответствующие им на кадре I_k , с помощью алгоритма вычисления разреженного оптического потока.

На *втором шаге* определяются \mathbf{P}^k и Θ^k – массивы положений и траекторий отслеживаемых точек на k -м кадре. Точки \mathbf{P}^k ищутся при помощи вычисления оптического потока между кадрами I_{k-1} и I_k для точек \mathbf{P}^{k-1} . После этого из массивов Λ , \mathbf{P}^k , \mathbf{P}^{k-1} и Θ^{k-1} удаляются некоторые элементы, не удовлетворяющие нескольким перечисленным ниже условиям. Если точке p_i^k выходит за пределы кадра I_k либо длина вектора смещения точки $ds_i^k = \|p_i^k - H^k(p_i^{k-1})\|$ больше заданного параметра алгоритма d_{\max} , то из массивов Λ , \mathbf{P}^k , \mathbf{P}^{k-1} и Θ^{k-1} удаляются P_i , p_i^k , p_i^{k-1} и S_i^{k-1} соответственно. Далее все неудаленные точки из \mathbf{P}^k добавляются в массивы из Λ , т. е. каждая точка $p_i^k \in \mathbf{P}^k$ добавляется в соответствующий ей массив $P_i \in \Lambda$. Чтобы найти траектории Θ^k , нужно сначала перевести все точки траекторий Θ^{k-1} на кадр I_k при помощи гомографии H_{k-1}^k и добавить к ним найденные при помощи оптического потока точки \mathbf{P}^k .

На *третьем шаге* алгоритма в Θ^k ищутся траектории, которые соответствуют движущимся на кадре I_k точкам. Вначале для каждой траектории $S_i^k \in \Theta^k$, количество точек в которой m_i , вычисляются три параметра: L_i – суммарная длина траектории S_i^k ; d_i – медиана среди 10 последних (либо всех, если $m_i < 10$) смещений точек в траектории S_i^k ; η_i – условный коэффициент, который принимает значение 1, если траектория S_i^k не содержит резких (более 90°) изменений направления, и 0 – в противном случае. После этого траектория S_i^k считается соответствующей движущейся точке при выполнении условий

$$\begin{cases} d_i \geq d_{\min}, \\ L_i > d_{\min} \cdot m_i, \\ \eta_i = 1, \\ m_i > \tau_{\min}, \end{cases}$$

где d_{\min} и τ_{\min} – параметры алгоритма, которые задают минимальное смещение и минимальное время отслеживания точек соответственно. На рис. 1 положения и траектории движущихся точек показаны голубым цветом, а неподвижных – черным.



Рис. 1. Положения и траектории движущихся и неподвижных отслеживаемых точек для одного из кадров



Рис. 2. Группы траекторий движущихся точек для одного из кадров

На четвертом шаге производится объединение положений и траекторий движущихся точек в группы. Пусть $\hat{\mathbf{P}}^k \subset \mathbf{P}^k$ и $\hat{\Theta}^k \subset \Theta^k$ – массивы положений и траекторий движущихся точек. Вначале для каждой движущейся точки с положением $p_i^k \in \hat{\mathbf{P}}^k$ и траекторией $S_i^k \in \hat{\Theta}^k$ на кадре I_k определяются два параметра: вектор скорости движущейся точки $V_i^k = (s_i^{k,k} - s_i^{k-m_i+1,k}) / m_i$, где m_i равно либо количеству точек в траектории S_i^k , если оно меньше 10, либо 10; оценка расстояния до ближайших точек группы $\rho_i^k = 4 + 2 \cdot \log_2 r_i^{\min}$, где r_i^{\min} – расстояние от точки p_i^k до ближайшей точки из $\hat{\mathbf{P}}^k$. Затем находится первая группа движущихся точек с положениями $\hat{\mathbf{Q}}_1^k \subset \hat{\mathbf{P}}^k$ и траекториями $\hat{\Omega}_1^k \subset \hat{\Theta}^k$. Для этого положение $p_i^k \in \hat{\mathbf{P}}^k$ и траектория $S_i^k \in \hat{\Theta}^k$ любой движущейся точки добавляются в пустые массивы $\hat{\mathbf{Q}}_1^k$ и $\hat{\Omega}_1^k$ и затем находятся положения и траектории движущихся точек с близкими к p_i^k и V_i^k положениями и скоростями, т. е. движущиеся точки с положениями $p_j^k \in \hat{\mathbf{P}}^k$ и траекториями $S_j^k \in \hat{\Theta}^k$, для которых выполняются условия

$$\begin{cases} r_{i,j}^k \leq \rho_i^k + \rho_j^k, \\ \|V_i^k\| < 2 \cdot \|V_j^k\|, \\ \|V_j^k\| < 2 \cdot \|V_i^k\|, \\ \alpha_{i,j}^k < \pi/3, \end{cases}$$

где $r_{i,j}^k$ – расстояние между точками p_i^k и p_j^k ; $\alpha_{i,j}^k$ – угол между векторами V_i^k и V_j^k . Далее точка с положением $p_i^k \in \hat{\mathbf{Q}}_1^k$ и траектория $S_i^k \in \hat{\Omega}_1^k$ помечаются как просмотренные, просматривается следующая движущаяся точка с положением $p_j^k \in \hat{\mathbf{Q}}_1^k$ и траекторией $S_j^k \in \hat{\Omega}_1^k$, а в массивы $\hat{\mathbf{Q}}_1^k$ и $\hat{\Omega}_1^k$ добавляются положения и траектории близких к ней движущихся точек в том случае, если они не были добавлены ранее. Процесс добавления положений и траекторий новых точек и траекторий продолжается, пока не будут просмотрены все точки и траектории в массивах $\hat{\mathbf{Q}}_1^k$ и $\hat{\Omega}_1^k$. Аналогично находится вторая группа движущихся точек с положениями $\hat{\mathbf{Q}}_2^k \subset \hat{\mathbf{P}}^k \setminus \hat{\mathbf{Q}}_1^k$ и траекториями $\hat{\Omega}_2^k \subset \hat{\Theta}^k \setminus \hat{\Omega}_1^k$, и так продолжается до тех пор, пока все движущиеся точки с положениями и траекториями из $\hat{\mathbf{P}}^k$ и $\hat{\Theta}^k$ не будут добавлены в группы.

Пусть далее $\hat{\Sigma}^k = \{\hat{Q}_1^k, \dots, \hat{Q}_j^k, \dots\}$ и $\hat{\Psi}^k = \{\hat{\Omega}_1^k, \dots, \hat{\Omega}_j^k, \dots\}$ – множества таких найденных массивов положений и траекторий движущихся точек, в которых количество положений и траекторий превосходит заданный параметр v . На рис. 2 разными цветами показаны найденные массивы траекторий движущихся точек из $\hat{\Psi}^k$.

На *пятом шаге* алгоритма определяется, соответствуют ли найденные группы точек из $\hat{\Sigma}^k$ движущимся объектам, обнаруженным на предыдущем I_{k-1} кадре видеопоследовательности. Пусть объект $O_l \in \mathbf{O}$ был обнаружен на кадре I_{k-1} , тогда ему соответствует некоторая группа точек $Q_l^{k-1} \subset \mathbf{P}^{k-1}$. Пусть $\mathbf{O}^{k-1} \subset \mathbf{O}$ – массив объектов, обнаруженных на предыдущем кадре I_{k-1} . Будем считать, что движущаяся точка с положениями $P_i \in \Lambda$ на кадре I_{k-1} принадлежит объекту O_l , если $p_i^{k-1} \in Q_l^{k-1}$. Также будем считать, что движущаяся точка с положениями $P_i \in \Lambda$ принадлежит группе $\hat{Q}_j^k \in \hat{\Sigma}^k$ на кадре I_k , если $p_i^k \in \hat{Q}_j^k$. Тогда каждому объекту $O_l \in \mathbf{O}^{k-1}$, обнаруженному на кадре I_{k-1} , ставится в соответствие группа из $\hat{\Sigma}^k$, которой принадлежит на кадре I_k максимальное количество принадлежащих объекту O_l на кадре I_{k-1} движущихся точек. Если нет ни одной движущейся точки, одновременно принадлежащей объекту O_l на кадре I_{k-1} и какой-то группе из $\hat{\Sigma}^k$ на кадре I_k , то будем считать объект O_l потерянным на кадре I_k .

На *шестом шаге* определяются характеристики движущихся объектов, которым на предыдущем шаге удалось поставить в соответствие группы точек из $\hat{\Sigma}^k$. Пусть объекту $O_l \in \mathbf{O}^{k-1}$ соответствует группа $\hat{Q}_j^k \in \hat{\Sigma}^k$. Тогда группа точек Q_l^k либо равна \hat{Q}_j^k в том случае, если не было установлено соответствие между \hat{Q}_j^k и другими объектами из \mathbf{O}^{k-1} , либо состоит только из тех точек, принадлежащих \hat{Q}_j^k , которые являются положениями на кадре I_k движущихся точек, принадлежащих объекту O_l на кадре I_{k-1} . Далее находятся центр объекта C_l^k как среднее значение координат точек из группы Q_l^k , нормированная гистограмма цветов h_l^k точек из Q_l^k и скорость объекта V_l^k как среднее значение скоростей движущихся точек, принадлежащих Q_l^k на кадре I_k . Для нахождения траектории объекта T_l^k нужно сначала перевести все точки траекторий T_l^{k-1} на кадр I_k при помощи гомографии H_{k-1}^k и добавить к ним C_l^k .

На *седьмом шаге* алгоритма устанавливаются соответствия между группами точек из $\hat{\Sigma}^k$, которым не удалось поставить в соответствие движущиеся объекты, обнаруженные на предыдущем кадре I_{k-1} видеопоследовательности, и потерянными объектами. Пусть далее $\tilde{\mathbf{O}} \subset \mathbf{O}$ – массив потерянных объектов и для каждого объекта $O_w \in \tilde{\mathbf{O}}$ известен номер последнего кадра t_w , на котором объект O_w был обнаружен. Тогда вначале из $\hat{\Sigma}^k$ удаляются группы точек, соответствующие объектам из \mathbf{O}^{k-1} , а из массивов $\tilde{\mathbf{O}}$ и \mathbf{O} удаляются такие объекты $O_w \in \tilde{\mathbf{O}}$, для которых $t_w < k - \xi$, где ξ – заданный параметр. Пусть для каждого потерянного объекта $O_w \in \tilde{\mathbf{O}}$ известны следующие показатели: $C_w^{t_w, k-1}$ – найденное на кадре I_{k-1} положение, которое объект занимал в момент времени t_w ; $T_w^{t_w, k-1}$ – траектория объекта до момента времени t_w , найденная на кадре I_{k-1} ; $V_w^{t_w, k-1}$ – вектор скорости объекта в момент времени t_w , пересчитанный для кадра I_{k-1} . Тогда для каждого потерянного объекта $O_w \in \tilde{\mathbf{O}}$ находятся значения $C_w^{t_w, k} = H_{k-1}^k(C_w^{t_w, k-1})$ и $V_w^{t_w, k}(k) = H_{k-1}^k(C_w^{t_w, k-1} + V_w^{t_w, k-1}) - H_{k-1}^k(C_w^{t_w, k-1})$. Также для каждой группы точек $\hat{Q}_j^k \in \hat{\Sigma}^k$ вычисляются среднее значение координат точек \hat{C}_j^k , средняя скорость точек \hat{V}_j^k

и нормированная гистограмма цветов точек $\hat{\mathbf{h}}_j^k$. Затем рассчитываются расстояния от каждого потерянного объекта из $\tilde{\mathbf{O}}$ до каждой группы из $\hat{\Sigma}^k$. Расстояние $d_{w,j}$ от $O_w \in \tilde{\mathbf{O}}$ до $\hat{Q}_j^k \in \hat{\Sigma}^k$ находится по формулам

$$\begin{aligned} Z_{w,j} &= \hat{C}_j^k - C_w^{t_w \cdot k}, \\ d_{w,j}^p &= 1/(1 + \rho_{w,j}), \\ d_{w,j}^c &= \|Z_{w,j} - (k - t_w) \cdot V_w^{t_w \cdot k}\| / \|(k - t_w) \cdot V_w^{t_w \cdot k}\|, \\ d_{w,j}^a &= 2 - \cos(\alpha_{w,j}) - \cos(\beta_{w,j}), \\ d_{w,j}^h &= \|\hat{\mathbf{h}}_j^k - \mathbf{h}_w^{t_w}\|, \\ d_{w,j} &= 0,3 \cdot d_{w,j}^p + 0,1 \cdot d_{w,j}^c + 0,4 \cdot d_{w,j}^a + 0,3 \cdot d_{w,j}^h, \end{aligned}$$

где $\rho_{w,j}$ – количество движущихся точек, одновременно принадлежащих группе \hat{Q}_j^k на кадре I_k и объекту O_w на кадре I_{t_w} ; $\alpha_{w,j}$ – угол между векторами $Z_{w,j}$ и \hat{V}_j^k ; $\beta_{w,j}$ – угол между векторами $Z_{w,j}$ и $V_w^{t_w \cdot k}$. Далее последовательно устанавливаются соответствия между элементами массивов $\tilde{\mathbf{O}}$ и $\hat{\Sigma}^k$. Для этого в массивах $\tilde{\mathbf{O}}$ и $\hat{\Sigma}^k$ находятся элементы, расстояние между которыми минимально, и если это расстояние меньше единицы, то между ними устанавливается соответствие, в противном случае процесс поиска соответствий прекращается. Если процесс поиска не прекратился, то среди других элементов массивов $\tilde{\mathbf{O}}$ и $\hat{\Sigma}^k$ ищется новая пара с минимальным расстоянием и так продолжается до тех пор, пока процесс не остановится. После этого происходит вычисление характеристик объектов из $\tilde{\mathbf{O}}$, которым удалось поставить в соответствие группы из $\hat{\Sigma}^k$. Пусть объекту $O_w \in \tilde{\mathbf{O}}$ соответствует группа $\hat{Q}_j^k \in \hat{\Sigma}^k$, тогда $\mathbf{Q}_w^k = \hat{Q}_j^k$, $C_w^k = \hat{C}_j^k$, $V_w^k = \hat{V}_j^k$, $\mathbf{h}_w^k = \hat{\mathbf{h}}_j^k$. Чтобы найти траектории объекта T_w^k , нужно сначала перевести все точки траекторий $T_w^{t_w \cdot k-1}$ на кадр I_k при помощи гомографии H_{k-1}^k и добавить к ним C_w^k .

На *восьмом шаге* алгоритма в массив \mathbf{O} добавляются новые движущиеся объекты, соответствующие некоторым группам точек из $\hat{\Sigma}^k$. Пусть $\hat{Q}_j^k \in \hat{\Sigma}^k$ не соответствует движущимся объектам из $\tilde{\mathbf{O}}$ и количество точек в самой длинной траектории из \hat{Q}_j^k больше заданного параметра τ , тогда в массив \mathbf{O} длины q добавляется новый объект O_{q+1} с характеристиками $\mathbf{Q}_{q+1}^k = \hat{Q}_j^k$, $C_{q+1}^k = \hat{C}_j^k$, $T_{q+1}^k = \{C_{q+1}^k\}$, $V_{q+1}^k = \hat{V}_j^k$, $\mathbf{h}_{q+1}^k = \hat{\mathbf{h}}_j^k$.

На *девятом шаге* алгоритма производится индикация движущихся объектов. Если длина траектории T_l^k объекта $O_l \in \mathbf{O}^k$ больше заданного параметра алгоритма λ , то для данного объекта вычисляется ограничивающий прямоугольник R_l^k точек \mathbf{Q}_l^k , который рисуется на изображении. Если объекту O_l на предыдущих кадрах не был присвоен уникальный идентификатор, то он присваивается на текущем кадре. На рис. 3 показан итоговый вид индикации движущихся объектов для одного из кадров.

Хотя движущиеся объекты были найдены на предыдущем шаге, для того чтобы алгоритм мог и дальше находить движущиеся объекты на последующих кадрах, нужно совершить еще несколько дополнительных шагов.

На *десятом шаге* удаляются некоторые положения и траектории отслеживаемых точек из массивов Λ , \mathbf{P}^k и Θ^k . Если количество точек в траектории S_i^k больше τ_{\min} и либо $d_i < d_{\min}$, либо $\eta_i = 0$, то из Λ , \mathbf{P}^k и Θ^k удаляются P_i , p_i^k и S_i^k соответственно.



Рис. 3. Индикация движущихся объектов для одного из кадров



Рис. 4. Итоговый вид изображения D_k для одного из кадров

Одиннадцатым шагом алгоритма является добавление новых отслеживаемых точек. Пусть количество точек в массиве \mathbf{P}^k на данный момент равно q , где $q < n$. Обозначим число точек, которые необходимо добавить к \mathbf{P}^k , как $\mu = n - q$. Далее строится изображение D_k с такой же размерностью, как у кадра I_k , интенсивность каждого пиксела p которого равна $D_k(p) = \min(|I_k(p) - I_{k-1}(H_k^{k-3}(p))|, |I_k(p) - I_{k-1}(H_k^{k-6}(p))|)$ в том случае, если номер кадра $k > 6$, либо $D_k(p) = |I_k(p) - I_{k-1}(H_k^{k-1}(p))|$, если $k \leq 6$. В представленных формулах H_k^{k-j} – проективное преобразование, которое переводит точки кадра I_k на кадр I_{k-j} . (H_k^{k-j} можно найти как композицию преобразований между соседними кадрами, т. е. $H_k^{k-j} = H_{k-j+1}^{k-j} \circ \dots \circ H_{k-1}^{k-2} \circ H_k^{k-1}$, где, например, H_k^{k-1} – отображение, обратное к H_{k-1}^{k-1} .) Далее в D_k обнуляются значения тех пикселей, интенсивность которых меньше заданного параметра γ , и пикселей, соответствующих уже отслеживаемым точкам \mathbf{P}^k . На рис. 4 показан итоговый вид изображения D_k для одного из кадров. После нахождения изображения D_k на нем ищутся связанные области пикселей с ненулевой интенсивностью. После этого из $q/3$ самых больших найденных областей по очереди выбираются новые точки p_{q+j}^k , где $j \leq \mu$, и добавляются в \mathbf{P}^k . В массивы Λ и Θ^k также добавляются новые элементы $P_{q+j}^k = \{p_{q+j}^k\}$ и $S_{q+j}^k = \{p_{q+j}^k\}$.

Как было сказано выше, действия, которые осуществляются на стартовых кадрах I_1 и I_2 видеопотока, отличаются от описанных выше. Кадр I_1 просто сохраняется. Для кадра I_2 сначала необходимо найти проективное преобразование H_1^2 между I_1 и I_2 , как на первом шаге алгоритма, после чего добавить положения и траектории n отслеживаемых точек в пустые массивы Λ , \mathbf{P}^2 и Θ^2 , как на одиннадцатом шаге алгоритма.

Результаты тестирования алгоритма. Работа представленного алгоритма была проверена на видеопоследовательностях двух широко известных баз данных (UAV123 [5] и VisDrone [6]), снятых бортовой видеокамерой БПЛА. Благодаря появившейся недавно базе данных PESMOD [7], содержащей размеченные видеопоследовательности с БПЛА, удалось численно сравнить результаты работы представленного алгоритма с тремя известными алгоритмами, основанными на компенсации движения: MCD [8], SCBU [9] и разработанным создателями базы данных PESMOD.

Для тестирования алгоритма из баз данных UAV123 и VisDrone были выбраны видеопоследовательности, полученные съемкой с достаточно большой высоты. Угол наклона камеры на видеопоследовательностях варьируется от 0 до 80° к вертикали. Выбранные видеопоследовательности были разбиты на две группы. В первую группу были отобраны 6 видеопоследовательностей, снятых с высоты выше 100 м, а во вторую – 16 видеопоследовательностей, снятых с высоты около 50 м. На тестовых видеопоследовательностях присутствуют движущиеся

объекты разного размера (от 40 до 10 000 пикселей), снятые при разном освещении. Количество одновременно движущихся объектов на некоторых видеопоследовательностях достигает 20. Размер кадров равен 1280×720 пикселей. Работа алгоритма тестировалась на персональном компьютере с процессором Intel(R) Core(TM) i7-4770 с тактовой частотой 3,5 ГГц. Быстродействие алгоритма на всех видеопоследовательностях составляет 25–30 кадров в секунду. На рис. 5 показаны результаты применения алгоритма на кадрах из видеопоследовательностей первой (сверху) и второй (снизу) групп соответственно.

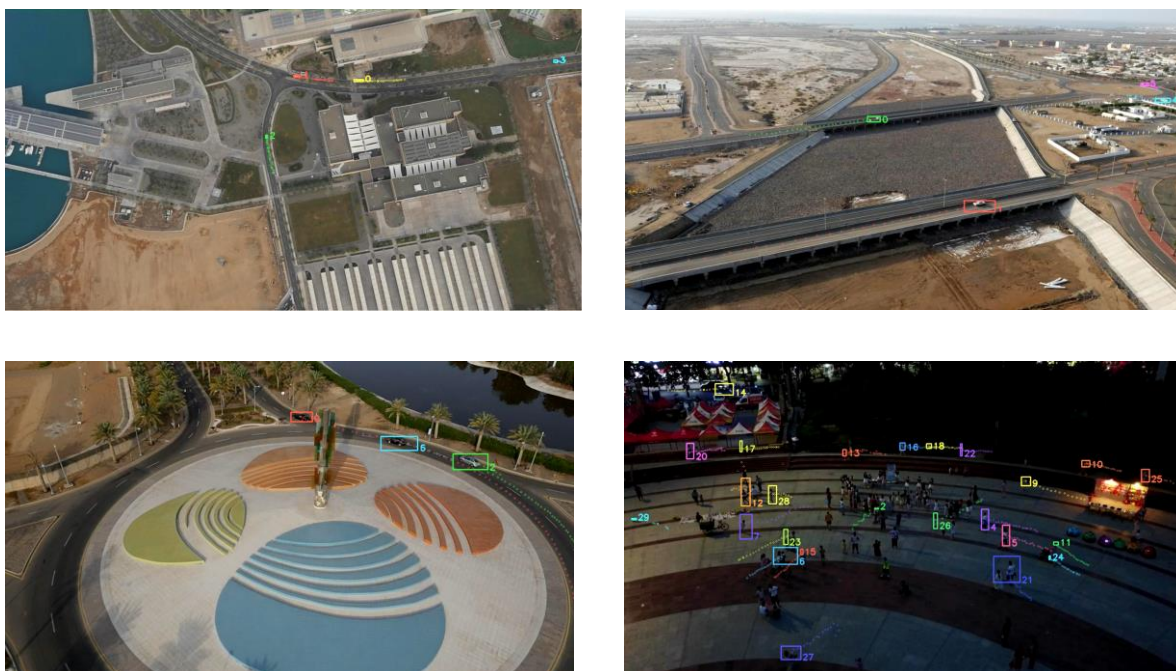


Рис. 5. Примеры работы алгоритма на кадрах видеопоследовательностей первой (сверху) и второй (снизу) групп

При проведении анализа работы алгоритма для каждого движущегося объекта из видеопоследовательностей были записаны интервалы его движения, а также интервалы верной индикации движения, для каждого интервала индикации объекта – номер, которым идентифицировался объект. Отдельно были записаны номера и интервалы ошибочной индикации движущихся объектов. В том случае, если несколько объектов идентифицировались одним номером, считалось, что в данный момент идентифицировался только один объект (рис. 6). Если один объект идентифицировался несколькими номерами, считалось, что все индикации, кроме одной, ошибочны (рис. 7). Благодаря такому ручному описанию движений объектов и результатов работы алгоритма на видеопоследовательностях были вычислены некоторые характеристики качества работы алгоритма.



Рис. 6. Пример идентификации двух движущихся объектов одним индексом



Рис. 7. Пример идентификации одного движущегося объекта двумя индексами

Далее вначале рассмотрим характеристики качества индикации движущихся объектов, а затем характеристики качества сопровождения движущихся объектов.

С целью оценки качества индикации движущихся объектов вычислялись следующие характеристики: количество движущихся объектов на всех видеопоследовательностях, n ; количество обнаруженных хотя бы один раз движущихся объектов на всех кадрах всех видеопоследовательностей, m ; количество изображений движущихся объектов на всех кадрах всех видеопоследовательностей, N ; количество верных индикаций движущихся объектов на всех кадрах всех видеопоследовательностей, TP ; количество ошибочных индикаций движущихся объектов на всех кадрах всех видеопоследовательностей, FP ; точность (*precision*) индикации, $Pr = TP / (TP + FP)$; полнота (*recall*) индикации, $R = TP / N$.

Ниже представлены результаты тестирования для двух групп видеопоследовательностей. Видеопоследовательности первой группы, снятые с большой высоты, содержат движущиеся объекты небольшого размера (около 100 пикселей), смещение которых на соседних кадрах невелико, поэтому алгоритм запускался с $n = 2000$, $d_{\max} = 100$, $d_{\min} = 0,25$, $\tau_{\min} = 5$, $\tau = 20$, $\nu = 3$, $\xi = 100$, $\gamma = 20$ и $\lambda = 10$. Видеопоследовательности второй группы, снятые с небольшой высоты, содержат объекты разного размера и типа (в основном люди и автомобили), величина смещения объектов между кадрами сильно зависит от типа объекта и высоты съемки. На второй группе видеопоследовательностей алгоритм запускался с $n = 2000$, $d_{\max} = 100$, $d_{\min} = 0,5$, $\tau_{\min} = 5$, $\tau = 20$, $\nu = 5$, $\xi = 100$, $\gamma = 20$ и $\lambda = 40$. Оценки качества индикации движущихся объектов на кадрах видеопоследовательностей первой и второй групп приведены в табл. 1.

Таблица 1

Оценки качества индикации движущихся объектов

	Группа	
	Первая	Вторая
Количество движущихся объектов, n	48	229
Количество обнаруженных движущихся объектов, m	40	192
Количество изображений движущихся объектов на кадрах, N	10 869	59 405
Количество верных индикаций движущихся объектов, TP	5826	31 791
Количество ошибочных индикаций движущихся объектов, FP	168	1736
Точность индикации движущихся объектов, Pr	0,972	0,948
Полнота индикации движущихся объектов, R	0,536	0,535

Относительно невысокие значения полноты индикации R объясняются тем, что алгоритм перед индикацией объекта некоторое время отслеживает его до того момента, пока не будет зафиксировано значительное смещение объекта. Благодаря этому удалось получить высокую точность индикации Pr .

Чтобы увидеть зависимость вероятности верной индикации движущихся объектов от времени их движения, для каждого изображения движущегося объекта на кадрах видеопоследовательности было вычислено время движения – количество кадров, прошедшее с момента начала его движения. Оценки зависимости вероятности верной индикации движущегося объекта от времени движения для видеопоследовательностей первой группы приведены на рис. 8, а и 9, а, второй группы – на рис. 8, б и 9, б. На рис. 8 синим цветом показаны графики зависимости количества изображений движущихся объектов на кадрах видеопоследовательностей от времени движения, желтым – графики зависимости количества верных индикаций изображений движущихся объектов от времени движения. На рис. 9 изображены графики зависимости вероятности верной индикации движущихся объектов на кадрах видеопоследовательностей от времени движения.

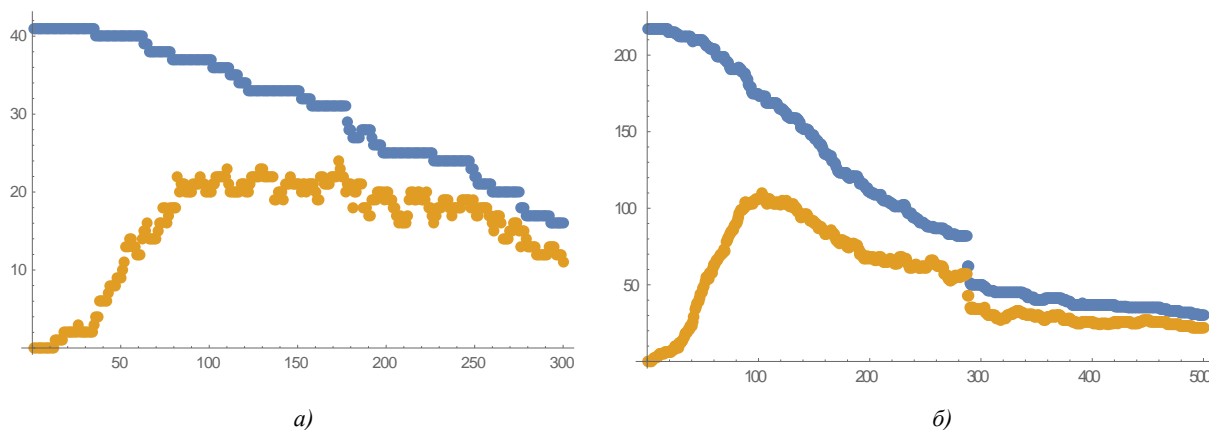


Рис. 8. Графики зависимостей количества изображений движущихся объектов и их индикаций от времени движения на кадрах видеопоследовательностей: а) первой группы; б) второй группы

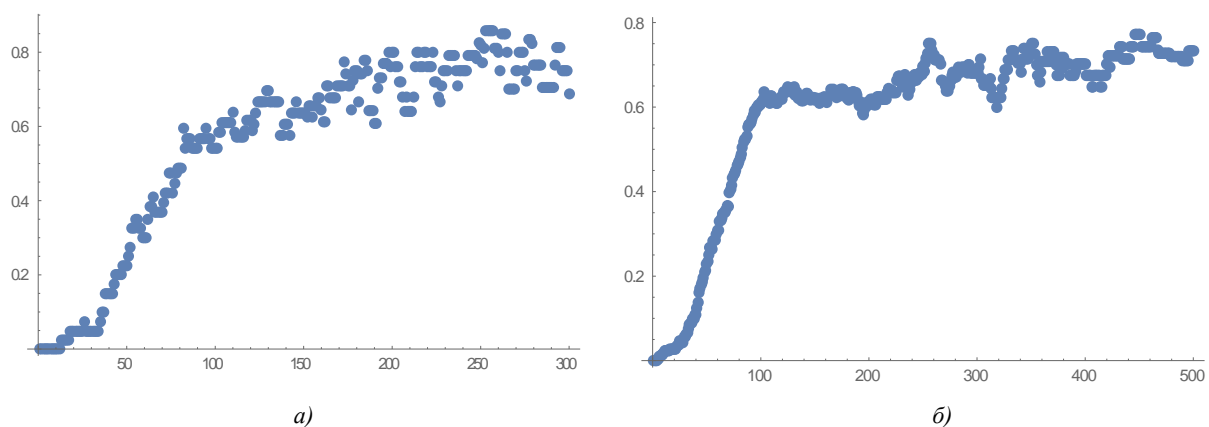


Рис. 9. Графики зависимости вероятности верной индикации движущихся объектов от времени движения для видеопоследовательностей: а) первой группы; б) второй группы

На представленных графиках видно, что алгоритм плохо обнаруживает движущиеся объекты в начале их движения, однако со временем вероятность индикации линейно растет примерно до 100-го кадра, после 100-го кадра вероятность индикации составляет примерно 0,6–0,8.

С целью оценки качества отслеживания движущихся объектов вычислялись следующие характеристики: количество обнаруженных движущихся объектов на всех видеопоследовательностях, m ; количество верно отслеженных движущихся объектов на всех видеопоследовательностях, t ; вероятность верного отслеживания объекта на видеопоследовательности, $P_i = t / m$. Обнаруженный объект считался верно отслеженным в том случае, если на протяжении всего времени движения он идентифицировался только одним числом и ни один другой объект (либо ложное обнаружение) не идентифицировался этим числом. Оценки качества отслеживания движущихся объектов на кадрах видеопоследовательностей первой и второй групп приведены в табл. 2.

Таблица 2

Оценки качества отслеживания движущихся объектов

	Группа	
	Первая	Вторая
Количество обнаруженных движущихся объектов, m	40	192
Количество верно отслеженных движущихся объектов, t	35	140
Вероятность верного отслеживания объекта, P_i	0,875	0,729

Из представленных вероятностей неясно, как долго алгоритм может отслеживать движущийся объект, поскольку все верные сопровождения могут прийти на объекты, которые двигались непродолжительное время. Для того чтобы увидеть зависимость вероятности верного отслеживания такого объекта от времени движения, для каждого идентифицированного изображения движущегося объекта на кадрах видеопоследовательности было вычислено время отслеживания – количество кадров, прошедшее с момента его первой индикации на предыдущих кадрах. Также было проверено, верно ли отслежен объект, т. е. идентифицируется ли он на всех предыдущих кадрах только одним числом. Результаты анализа зависимости вероятности верного отслеживания движущегося объекта от времени отслеживания для видеопоследовательностей первой группы приведены на рис. 10, *а* и 11, *а*, второй группы – на рис. 10, *б* и 11, *б*. На рис. 10 и 11 желтым цветом показаны графики зависимости количества идентифицированных изображений движущихся объектов на кадрах видеопоследовательностей от времени отслеживания, синим – графики зависимости количества верно отслеженных изображений движущихся объектов от времени отслеживания. На рис. 11 изображены графики зависимости вероятности верного отслеживания движущихся объектов на кадрах видеопоследовательностей от времени отслеживания.

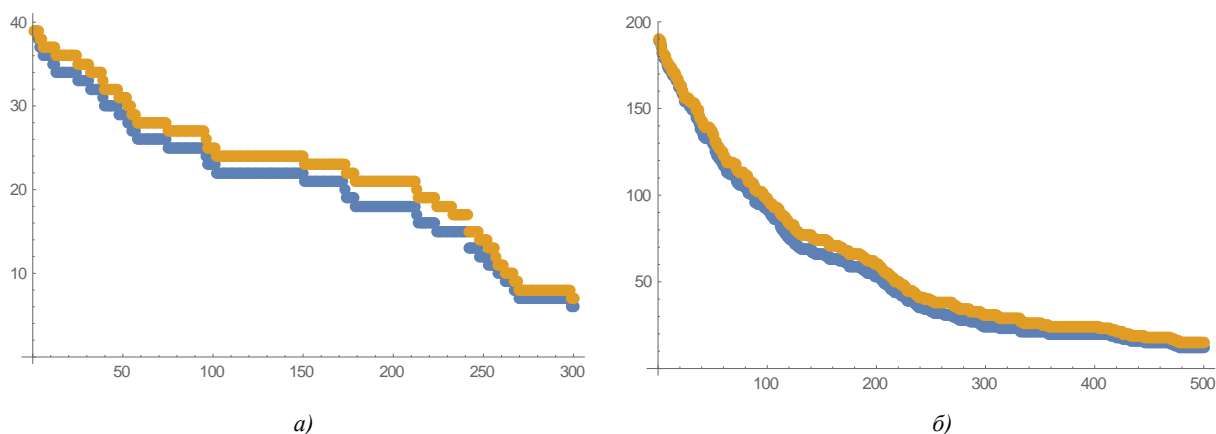


Рис. 10. Графики зависимости количества идентифицированных и верно отслеженных изображений движущихся объектов от времени отслеживания на кадрах видеопоследовательностей: *а*) первой группы; *б*) второй группы

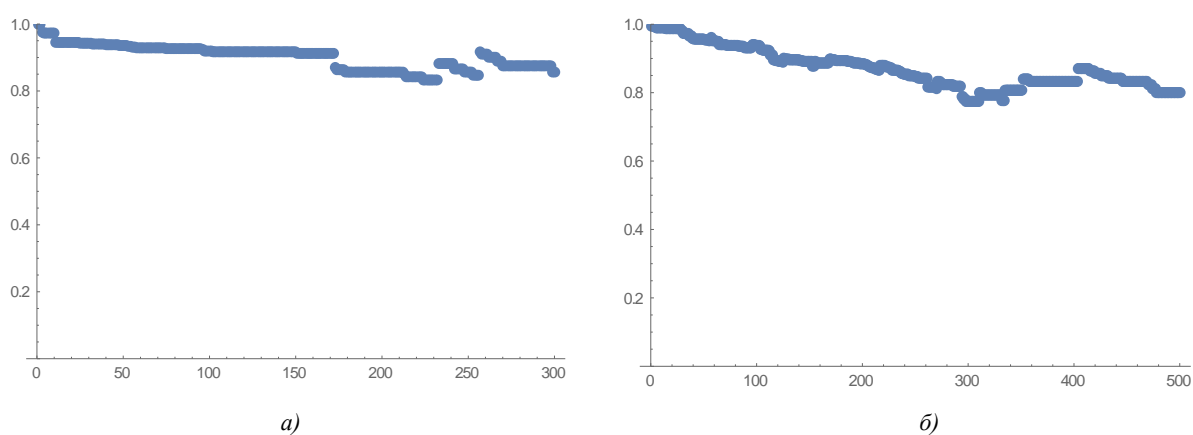


Рис. 11. Графики зависимости вероятности верного отслеживания движущихся объектов от времени отслеживания для видеопоследовательностей: *а*) первой группы; *б*) второй группы

На рис. 10 и 11 видно, что алгоритм надежно сопровождает движущиеся объекты, вероятность верного сопровождения незначительно снижается с течением времени, однако остается высокой даже для объектов, которые на тестовых видеопоследовательностях двигались наиболее продолжительное время.

Кроме представленного выше анализа работы алгоритма, было проведено его сравнение с известными алгоритмами индикации движущихся объектов, основанными на компенсации движения. Сравнение проводилось на базе данных PESMOD, в которой находятся шесть сложных видеопоследовательностей с небольшими движущимися объектами. Размер кадров в PESMOD равен 1920×1080 пикселей. На рис. 16 приведены примеры работы алгоритма на кадрах последовательностей из PESMOD. Зелеными рамками показаны движущиеся объекты, выделенные вручную, синими – найденные алгоритмом.



Рис. 16. Примеры работы алгоритма на кадрах видеопоследовательностей PESMOD

Благодаря программному обеспечению, разработанному создателями базы PESMOD, для каждой видеопоследовательности удалось вычислить следующие критерии качества работы алгоритма индикации: точность (precision) индикации, Pr ; полноту (recall) индикации, R и обобщающий их критерий $F_1 = 2 \cdot Pr \cdot R / (Pr + R)$.

В табл. 3 производится сравнение рассматриваемого в настоящей статье алгоритма с известными алгоритмами MCD, SCBU, а также алгоритмом, разработанным создателями базы данных PESMOD. Данные, содержащиеся в таблице, – это средние значения критериев Pr , R и F_1 , полученные для всех шести последовательностей из PESMOD. Предлагаемый алгоритм тестировался при $n=1000$, $d_{\max}=100$, $d_{\min}=0,5$, $\tau_{\min}=5$, $\tau=5$, $\nu=3$, $\xi=100$, $\gamma=30$, $\lambda=10$.

Таблица 3

Сравнительные характеристики представленного алгоритма и известных алгоритмов

Показатели	MCD	SCBU	PESMOD	Представленный алгоритм
Pr	0,5424	0,4268	0,5452	0,8636
R	0,5607	0,4416	0,5070	0,6338
F_1	0,3828	0,4108	0,4302	0,6718

Из табл. 3 видно, что разработанный алгоритм превосходит рассмотренные известные алгоритмы, основанные на компенсации движения. Наибольшим преимуществом алгоритма является высокий показатель точности индикации Pr .

Проведенный анализ показывает, что предварительное отслеживание движущихся объектов позволяет значительно понизить вероятность ложных обнаружений движущихся объектов.

Заключение. Разработан алгоритм автоматического обнаружения и отслеживания движущихся объектов, предназначенный для использования на борту БПЛА. Эксперименты с видеопоследовательностями, снятыми видеокамерами с различных БПЛА, показали работоспособность предложенного алгоритма. Достоинством алгоритма является то, что он редко обнаруживает ложное движение. Вместе с тем данный алгоритм не сразу обнаруживает движущиеся объекты, а только через некоторое время после начала их движения – это, несомненно, его недостаток.

Численное сравнение работы представленного алгоритма на базе данных PESMOD с тремя известными алгоритмами, основанными на компенсации движения, показало, что он превосходит другие алгоритмы по всем важным показателям качества индикации движущихся объектов, поскольку надежно сопровождает движущиеся объекты и способен в реальном времени решать задачу сопровождения их большого количества. Кроме того, представленный алгоритм способен обнаруживать и отслеживать движущиеся объекты малого размера (до 40 пикселей), смещение которых между соседними кадрами в среднем может составлять 0,25 пикселя. Таким образом, разработанный алгоритм может успешно применяться на борту БПЛА.

References

1. Chapel M.-N., Bouwmans T. *Moving Objects Detection with a Moving Camera: A Comprehensive Review*. Available at: <https://arxiv.org/abs/2001.05238> (accessed 19.04.2021).
2. Fiaz M., Mahmood A., Jung S.-K. *Tracking Noisy Targets: A Review of Recent Object Tracking Approaches*. Available at: <https://arxiv.org/abs/1802.03098> (accessed 19.04.2021).
3. Ye D. H., Li J., Chen Q., Wachs J., Bouman C. Deep learning for moving object detection and tracking from a single camera in unmanned aerial vehicles (UAVs). *IS&T International Symposium on Electronic Imaging 2018*. Burlingame, California, USA, 2018, pp. 4661–4666.
4. Lucas B., Kanade T. An iterative image registration technique with an application to stereo vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*. Vancouver, 1981, vol. 2, pp. 674–679.
5. Mueller M., Smith N., Ghanem B. A benchmark and simulator for uav tracking. *Proceedings of the 14th European Conference on Computer Vision (ECCV '16)*. Amsterdam, 2016, pp. 445–461. https://doi.org/10.1007/978-3-319-46448-0_27
6. Du D., Zhu P., Wen L., Bian X., Ling H., ..., Liu Z. VisDrone-DET2019: the vision meets drone object detection in image challenge results. *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. Seoul, 2019, pp. 213–226. <https://doi.org/10.1109/ICCVW.2019.00030>
7. Delibasoglu I. *UAV Images Dataset for Moving Object Detection from Moving Cameras*. Available at: <https://arxiv.org/abs/2103.11460> (accessed 19.04.2021).
8. Yi K. M., Yun K., Kim S. W., Chang H., Choi J. Detection of moving objects with non-stationary cameras in 5.8ms: bringing motion detection to your mobile device. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*. Portland, OR, USA, 2013, pp. 27–34.
9. Yun K., Lim J., Choi J. Y. Scene conditional background update for moving object detection in a moving camera. *Pattern Recognition Letters*, 2017, vol. 88, pp. 57–63. <https://doi.org/10.1016/j.patrec.2017.01.017>

Информация об авторе

Жук Роман Сергеевич, младший научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Беларусь.
E-mail: ramanzhuck@gmail.com

Information about the author

Raman S. Zhuk, Junior Researcher, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.
E-mail: ramanzhuck@gmail.com