ISSN 1561-8323 (Print)
ISSN 2524-2431 (Online)

ИНФОРМАТИКА

INFORMATICS

УДК 004
https://doi.org/10.29235/1561-8323-2023-67-1-20-26

Поступило в редакцию 03.06.2022
Received 03.06.2022

Б. А. Залесский

Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Республика Беларусь

АЛГОРИТМ ОБНАРУЖЕНИЯ ДВИЖУЩИХСЯ ОБЪЕКТОВ, НАБЛЮДАЕМЫХ ВИДЕОКАМЕРОЙ

(Представлено членом-корреспондентом А. В. Тузиковым)

Аннотация. Представлен алгоритм обнаружения движущихся объектов, наблюдаемых видеокамерой. Алгоритм основан на обнаружении движения на кадрах видеопотока, полученного в общем случае движущейся видеокамерой, а также на построении и анализе траекторий движущихся объектов. Особенность алгоритма заключается в обнаружении на кадрах связных областей (кластеров) возможного движения, которые с большой вероятностью принадлежат изображениям движущихся объектов, а затем – нахождении на обнаруженных кластерах точек возможного движения и построении с помощью оптического потока траекторий движения найденных точек. Для обнаружения движущихся объектов используются только гладкие траектории. Остальные траектории удаляются из рассмотрения. Объект считается движущимся на текущем кадре, если в него попадает достаточное число траекторий движущихся точек, найденных на предыдущих кадрах. Представленный алгоритм имеет малую вычислительную сложность, что позволяет использовать его в режиме реального или близкого к реальному времени на малых вычислителях, имеющих только несколько процессоров архитектуры ARM без мощных средств параллельных вычислений типа GPU или нейросетевых процессоров NPU.

Ключевые слова: видеопоток, движущиеся объекты, обнаружение объектов

Для цитирования: Залесский, Б. А. Алгоритм обнаружения движущихся объектов, наблюдаемых видеокамерой / Б. А. Залесский // Докл. Нац. акад. наук Беларуси. – 2023. – Т. 67, № 1. – С. 20–26. https://doi.org/10.29235/1561-8323-2023-67-1-20-26

Boris A. Zalesky

United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Republic of Belarus

ALGORITHM FOR DETECTION OF MOVING OBJECTS OBSERVED BY A VIDEO CAMERA

(Communicated by Corresponding Member Alexander V. Tuzikov)

Abstract. An algorithm to detect moving objects captured by a moving video camera is presented. The algorithm is based on detection of motion on video frames taken by a moving video camera, as well as on finding and analyzing the trajectories of moving objects. A feature of the algorithm is detection on frames of connected areas (clusters) of possible object motion. Then moving points on the detected clusters are found, and those points trajectories are built with help of the optical flow. The trajectories are used as features of moving objects. Only smooth trajectories are exploited for detection of moving objects, and the remaining ones are removed from consideration. An object is considered as moving on the current frame if it contains ends of a sufficient number of trajectories of moving points found on previous frames. The presented algorithm has a low computational complexity, which allows it to be used in real or near real time on small computers that have only a few processors of the ARM architecture without powerful parallel computing tools such as GPUs or neural network processors NPU.

Keywords: video stream, moving objects, object detection

For citation. Zalesky B. A. Algorithm for detection of moving objects observed by a video camera. *Doklady Natsional'noi akademii nauk Belarusi = Doklady of the National Academy of Sciences of Belarus*, 2023, vol. 67, no. 1, pp. 20–26 (in Russian). https://doi.org/10.29235/1561-8323-2023-67-1-20-26

Введение. Рассматривается задача обнаружения движущихся объектов на кадрах видеопотока, получаемого в общем случае движущейся камерой.

В последние годы значительно возрос интерес к алгоритмам обнаружения объектов на кадрах видеопотока. Это объясняется, с одной стороны, расширением сферы применения средств видеонаблюдения и навигации с использованием видеокамер различного типа, а с другой – появлением большого числа новых подходов к решению задачи, позволивших довести надежность и точность обнаружения до уровня, достаточного для практического использования результатов.

Основные подходы к решению поставленной задачи и обзор последних результатов по обнаружению движущихся объектов представлены в [1–3].

Следует отметить, что большая часть новых алгоритмов обнаружения объектов, в том числе движущихся, основана на использовании глубинного обучения и сверточных нейронных сетей (СНС). Однако существует потребность в менее вычислительно трудных алгоритмах, в которых не используются нейронные сети. Например, при эксплуатации малых беспилотных летательных аппаратов (БЛА) или недорогих систем видеонаблюдения, оборудованных маломощными вычислителями с ARM-архитектурой без средств параллельных вычислений типа GPU или нейросетевых процессоров NPU, невозможно применять программные реализации нейросетевых алгоритмов.

Далее представлен алгоритм CAMOD (Cluster Algorithm Moving Objects Detection) обнаружения движущихся объектов на кадрах видеопотока без использования СНС. Представленный алгоритм имеет относительно малую вычислительную сложность, что позволяет использовать его в режиме реального или близкого к реальному времени на малых вычислителях, имеющих только несколько процессоров архитектуры ARM.

Тестирование алгоритма проводилось на видеопоследовательностях, снятых различными видеокамерами, а также на общеизвестном размеченном наборе данных UAVDT.

Алгоритм CAMOD предназначен для обнаружения движущихся объектов на кадрах видеопотока, снятых в том числе движущейся камерой. Это делает возможным его применение на БЛА и других движущихся средствах видеонаблюдения, в том числе оборудованных маломощными вычислителями с архитектурой ARM.

Алгоритм CAMOD ориентирован на работу с полутоновым видеопотоком, кадры которого обозначим через $\mathbf{I}_t$. Множество пикселов кадра $\mathbf{I}_t$ размера $w \times h$ обозначим через $S = \{(x, y)\}$, $x = 0, ..., w - 1$, $y = 0, ..., h - 1$, а значения яркостей в пикселах $\mathbf{p} \in S$ через $I_t(\mathbf{p})$.

В случае цветного видеопотока алгоритм может применяться отдельно для каждого канала цветного видеопотока, однако это приводит к увеличению вычислительных затрат, но не улучшает его характеристики, поэтому лучше преобразовать кадры в полутоновые изображения.

Во многих алгоритмах-детекторах движения первичное обнаружение движущихся объектов производится с помощью маски движения [2; 3]. Для построения маски движения рассматривается текущий $\mathbf{I}_t$ кадр и один из его близких предшественников, например, $\mathbf{I}_{t-1}$. Предыдущий кадр $\mathbf{I}_{t-1}$ совмещается с текущим $\mathbf{I}_t$ с помощью проективного преобразования $P_{t-1,t}$ так, чтобы видимые на обоих кадрах неподвижные части сцены совпадали попиксельно (подробно процесс выравнивания кадров описан в [3]). Один из возможных способов построения маски движения для текущего кадра $\mathbf{I}_t$ заключается в бинаризации модуля разности $\left|P_{t-1,t}\mathbf{I}_{t-1} - \mathbf{I}_t\right|$, т. е. вычислении для наперед заданного порога τ, $0 < \tau \le 255$, бинарного изображения $\mathbf{D}(t)$ с яркостями

$$D(t, \mathbf{p}) = \begin{cases} 1, & \text{если } \left|P_{t-1,t} I_{t-1}(\mathbf{p}) - I_t(\mathbf{p})\right| \ge \tau \\ 0, & \text{иначе} \end{cases}.$$

Такой способ построения бинарной маски движения имеет существенный недостаток – из-за погрешности совмещения кадров $\mathbf{I}_t$ и $\mathbf{I}_{t-1}$, являющихся 2D-изображениями 3D-сцены, с помощью проективного преобразования $P_{t-1,t}$ на маске появляются области ложного движения (данный эффект в оптике называется параллаксом. В меньшей степени он проявляется в случае, когда расстояние до камеры значительно больше глубины сцены). Для уменьшения влияния параллакса в алгоритме CAMOD предложен более сложный подход к построению маски движения. Вначале

строятся две предварительные бинарные маски движения $\mathbf{D}_{\text{low}}(t)$ и $\mathbf{D}_{\text{high}}(t)$ для низкого r_{low} и высокого r_{high} порогов. Затем эти бинарные маски кластеризуются с помощью алгоритма наращивания областей [4]. Напомним, что кластером называется связное в четырехточечной (восьмиточечной или иной) системе окрестностей множество C пикселов $\mathbf{p}$ таких, что $D(t, \mathbf{p}) = 1$ для каждого пиксела $\mathbf{p} \in C$, и для любого соседнего с $\mathbf{p}$ пиксела $\mathbf{q} \notin C$ значение маски $D(t, \mathbf{q}) = 0$.

На кластерных представлениях масок $\mathbf{CD}_{\text{low}}(t)$ и $\mathbf{CD}_{\text{high}}(t)$ отбрасываются маленькие кластеры, а для оставшихся кластеров вычисляются ограничивающие их прямоугольники. На окончательной маске движения $\mathbf{D}(t)$ оставляются только те кластеры маски $\mathbf{CD}_{\text{low}}(t)$, ограничивающие прямоугольники которых имеют значительное пересечение (более 95 %) с хотя бы одним прямоугольником, ограничивающим кластер на $\mathbf{CD}_{\text{high}}(t)$. На рисунке *a* приведена обычная маска движения и маска движения (рисунок *b*), построенная предложенным способом.

Кластеры, оставленные на маске движения $\mathbf{D}(t)$, используются для обнаружения и сопровождения движущихся объектов. Для оценки движения объектов на каждом впервые появившемся на маске $\mathbf{D}(t)$ кластере выбирается заданное количество пикселов, траектории движения которых отслеживаются с помощью оптического потока. Если на одной из предыдущих масок $\mathbf{D}_{t-\Delta}$ кластер $C_j(t-\Delta)$ появился впервые, в момент его появления формируется множество, со-

a

b

Негативное изображение масок. Обычная маска движения $\mathbf{D}_{\text{low}}(t)$ содержит большие области ложного движения, возникающие из-за параллакса (*a*); маска движения $\mathbf{D}(t)$, построенная предложенным способом (*b*)

Negative image of masks. Usual motion mask $\mathbf{D}_{\text{low}}(t)$ contains large areas of false motion resulting from parallax (*a*); motion mask $\mathbf{D}(t)$ built by the proposed technique (*b*)

стоящее из заданного числа $\Bbbk$ равномерно распределенных на нем пикселов $\mathbf{p}_{j,m}(t-\Delta) \in C_j(t-\Delta)$, $0 \le m < \Bbbk$. Каждый пиксел $\mathbf{p}_{j,m}(t-\Delta)$ рассматривается как начальная точка $\mathbf{d}_{j,m}(t-\Delta) = \mathbf{p}_{j,m}(t-\Delta)$, порождающая новую траекторию. В момент времени $t-\Delta+1$ точка $\mathbf{d}_{j,m}(t-\Delta)$ переводится оптическим потоком в точку $\mathbf{d}_{j,m}(t-\Delta+1)$ на следующем кадре $\mathbf{I}_{t-\Delta+1}$ и так далее до точки $\mathbf{d}_{j,m}(t)$ текущего кадра (в общем случае, точки $\mathbf{d}_{j,m}(t)$ имеют рациональные координаты, в отличие от пикселов, имеющих целочисленные). Каждому кластеру $C_j(t-\Delta)$, впервые появившемуся в момент времени $t-\Delta$, соответствует множество наборов точек

$$\{\mathbf{d}_{j,m}(t-\Delta),\ \mathbf{d}_{j,m}(t-\Delta+1), \ldots, \mathbf{d}_{j,m}(t)\},\ 0 \le m < \Bbbk, \tag{1}$$

полученных с помощью перевода оптическим потоком образа начальной точки $\mathbf{d}_{j,m}(t-\Delta)$ с кадра на кадр. Данные наборы точек нельзя назвать траекториями, так как они вычислены без учета смещения изображений снимаемой сцены на кадрах относительно друг друга, вызванного движением камеры. Для построения траектории движения точек (1) на текущем кадре $\mathbf{I}_t$ нужно найти проекции точек $\mathbf{d}_{j,m}(\ell)$, $t-\Delta \le \ell \le t$, на текущий кадр с помощью проективных преобразований $P_{t-\Delta,t-\Delta+1}, P_{t-\Delta+1,t-\Delta+2}, \ldots, P_{t-1,t}$. Для этого используется суперпозиция проективных преобразований

$$P_{t-\ell,t} = P_{t-1,t} \cdot \ldots \cdot P_{t-i,t-i+1} \cdot \ldots \cdot P_{t-\ell,t-\ell+1}$$

(очевидно, что $P_{t,t}$ равно единичному оператору E). Траектории

$$T_{j,m}(t-\Delta,t) = \{\mathbf{q}_{j,m}(t-\Delta),\ \mathbf{q}_{j,m}(t-\Delta+1), \ldots, \mathbf{q}_{j,m}(t)\},\ 0 \le m < \Bbbk,$$

движения начальных точек $\mathbf{d}_{j,m}(t-\Delta)$ на текущем кадре имеют вид

$$T_{j,m}(t-\Delta,t) = \{P_{t-\Delta,t}\mathbf{d}_{j,m}(t-\Delta),\ P_{t-\Delta+1,t}\mathbf{d}_{j,m}(t-\Delta+1), \ldots, P_{t,t}\mathbf{d}_{j,m}(t)\},\ 0 \le m < \Bbbk,$$

т. е. точки $\mathbf{q}_{j,m}(\ell)$, $t-\Delta \le \ell \le t$, траектории $T_{j,m}(t-\Delta,t)$ вычисляются по формуле $\mathbf{q}_{j,m}(\ell) = P_{\ell,t}\mathbf{d}_{j,m}(\ell)$. Совокупность траекторий, первоначально порожденных кластером $C_j(t-\Delta)$ обозначим через $\mathbf{T}_j(t-\Delta,t)$.

Далее будем называть кластер $C_j(t)$ на маске $\mathbf{D}(t)$ впервые появившимся, если ни одна из существующих в данный момент траекторий не попадает в прямоугольник, ограничивающий этот кластер. Это не самое точное определение новизны кластера, однако его проверка занимает допустимое время.

До настоящего момента упоминались совокупности траекторий $\mathbf{T}_j(t-\Delta,t)$, порождаемые на впервые появившихся кластерах. Простые реализации оптического потока (например, Лукаса–Канаде), используемые для обеспечения режима реального времени работы алгоритма, в том числе на малых вычислителях, обладают ощутимой погрешностью, с которой приходится бороться. Для этого после перевода точек траекторий $T_{j,m}(t-\Delta,t-1)$, $0 \le m < \Bbbk$, из набора $\mathbf{T}_j(t-\Delta,t)$, порожденных отдельным кластером $C_j(t-\Delta)$ на текущий кадр $\mathbf{I}_t$ и их дополнения точками $\mathbf{d}_{j,m}(t)$ до $T_{j,m}(t-\Delta,t)$, вычисляется центр масс этих точек

$$\mathbf{m}_j(t) = \frac{1}{\Bbbk}\sum_{m=0}^{\Bbbk-1}\mathbf{d}_{j,m}(t),$$

а затем проверяется условие попадания последних точек траекторий $\mathbf{d}_{j,m}(t)$ в круг наперед заданного радиуса R_0, т. е. условие

$$\left\|\mathbf{m}_j(t) - \mathbf{d}_{j,m}(t)\right\| \le R_0,\ 0 \le m < \Bbbk. \tag{2}$$

Траектории $T_{j,m}(t-\Delta,t)$, для которых условие (2) не выполняется, удаляются из $\mathbf{T}_j(t-\Delta,t)$.

Удаляются также негладкие траектории, а также траектории с точками, средняя скорость которых слишком мала (такие траектории с большой вероятностью соответствуют неподвижным или останавливающимся объектам). Для удаления траекторий использован алгоритм, подробно описанный в [3].

Если число оставшихся после удаления траекторий в наборе $\mathbf{T}_j(t-\Delta, t)$ не превосходит двух, этот набор удаляется.

Если число оставшихся после удаления траекторий в наборе $\mathbf{T}_j(t-\Delta, t)$ больше двух, и оставшиеся траектории попадают в какой-либо кластер движения $C_i(t)$, вместо удаленных траекторий порождается такое же количество новых траекторий $T_{j,m}(t, t)$, состоящих в момент времени t из одной точки $\mathbf{d}_{j,m}(t)$. Начальные точки $\mathbf{d}_{j,m}(t)$ новых траекторий выбираются из кластера $C_i(t)$ на маске $\mathbf{D}(t)$, в который попали оставшиеся траектории из набора $\mathbf{T}_j(t-\Delta, t)$. Таким образом, после дополнения в наборе $\mathbf{T}_j(t-\Delta, t)$ вновь оказывается $\mathbb{k}$ траекторий $T_{j,m}(t_m, t)$, но возможно, порожденных на разных кадрах.

Набор траекторий $\mathbf{T}_j(t-\Delta, t)$ начинает рассматриваться как соответствующий движущемуся объекту, если в нем найдется хотя бы одна достаточно длинная траектория $T_{j,m}(t_m, t)$, или более конкретно, при первом выполнении для некоторого натурального числа τ условия

$$\max_{0 \le m < \mathbb{k}} (\dim(T_{j,m}(t_m, t))) \ge \tau. \tag{3}$$

Условие (3) используется для уменьшения вероятности ложного обнаружения движущихся объектов.

В результате применения алгоритма находятся прямоугольники Rec_j, ограничивающие кластеры маски $\mathbf{D}(t)$, в которые попали траектории хотя бы одного набора $\mathbf{T}_j(t-\Delta, t)$, удовлетворяющего условию (3). Находятся также центры масс $\mathbf{m}_j(t)$ конечных точек $\mathbf{d}_{j,m}(t)$ траекторий. С высокой вероятностью значительная часть прямоугольника Rec_j лежит внутри прямоугольной рамки, ограничивающей силуэт движущегося объекта, а центр масс $\mathbf{m}_j(t)$ траекторий $\mathbf{T}_j(t-\Delta, t)$ (удовлетворяющих условию (3)) принадлежит силуэту движущегося объекта или находится на очень близком расстоянии от него. Движущийся объект отмечается центром $\mathbf{m}_j(t)$ набора траекторий $\mathbf{T}_j(t-\Delta, t)$ (удовлетворяющих условию (3)).

Поскольку алгоритм CAMOD предназначен для выполнения на малых вычислителях, не имеющих GPU и NPU, он решает только задачу обнаружения движущихся объектов. Его можно применять в режиме, близком к режиму реального времени для сопровождения лишь нескольких движущихся объектов. Для надежного сопровождения большого числа обнаруженных движущихся объектов в настоящее время используются более вычислительно трудоемкие алгоритмы, основанные в том числе на анализе внешних признаков объектов с помощью СНС, выполняемые на GPU или NPU.

В случае необходимости и возможности решения задачи сопровождения всех движущихся объектов на вычислителе с GPU можно добавить к CAMOD один из лучших известных в настоящее время трекеров движущихся объектов Deep SORT [5].

Результаты и их обсуждение. Для исследования характеристик построенного алгоритма CAMOD было проведено его тестирование на видеопоследовательностях разных размеров, начиная от 320 × 240 до 1920 × 1080 (Full HD), снятых различными камерами.

Ниже приведены результаты тестирования алгоритма на открытом размеченном наборе данных UAVDT, содержащем 20 различных видео, состоящих суммарно из 16592 кадров размера 1024 × 540 пикселов. Размеченный движущийся на тестовом видео объект считался обнаруженным, если центр $\mathbf{m}_j(t)$ какого-либо набора траекторий $T_{j,m}(t_m, t)$, соответствующего этому объекту, находился от него на разрешенном расстоянии, не превышающем наперед заданное r.

При тестировании использовались следующие значения расстояний $r = 0,\ 1,\ \frac{1}{2}R_0,\ R_0$ пикселов от центров $\mathbf{m}_j(t)$ наборов траекторий до истинных (ground truth) прямоугольников, ограничивающих движущиеся объекты. Такой способ тестирования объясняется тем, что маска движения никогда не попадает полностью в силуэт движущегося объекта, и следовательно, точки траекторий $\mathbf{d}_{j,m}(t)$ время от времени оказываются рядом с объектом. При $r = 0$ движущийся объект считается обнаруженным, если центр $\mathbf{m}_j(t)$ набора траекторий попадает в истинный прямоугольник, ограничивающий этот объект.

Ниже приведены результаты тестирования для разных значений r при $R_0 = 10$ пикселов (таблица). При тестировании использованы стандартные характеристики: *точность* (Prec) –

отношение числа верно обнаруженных объектов к числу всех обнаруженных объектов, *полнота* (Rec) – отношение числа верно обнаруженных объектов к числу всех размеченных в датасете объектов и

$$F_1 = \frac{2\ \mathrm{Prec}\ \mathrm{Rec}}{\mathrm{Prec} + \mathrm{Rec}},$$

используемые в распознавании образов.

Результаты тестирования CAMOD

CAMOD test results

r (пикс)	Precision (%)	Recall (%)	F_1 (%)
0	62,22	55,15	58,47
1	62,63	55,65	58,94
5	63,68	57,44	60,40
10	64,39	60,35	62,31

Для сравнения разных алгоритмов нужно анализировать результаты их работы на одних и тех же размеченных наборах данных, которые не всегда можно получить в открытом доступе, поэтому эффективность CAMOD сравнивалась на использованном датасете UAVDT с эффективностью алгоритма, предложенного Р. С. Жуком в [3]. Этот алгоритм тестировался ранее и сравнивался на датасете PESMOD с другими известными алгоритмами. Он значительно превзошел другие известные алгоритмы, не использующие нейронные сети (результаты тестирования приведены в [3]).

Результаты, полученные алгоритмом [3] на наборе данных UAVDT: $\mathrm{Precision}_{[3]} = 64{,}96$ %, $\mathrm{Recall}_{[3]} = 42{,}35$ % и $F_{1[3]} = 51{,}27$ %. Результаты, полученные CAMOD при самой строгой проверке обнаружения движущегося объекта (при $r = 0$): $\mathrm{Precision}_{\mathrm{CAMOD}} = 62{,}22$ %, $\mathrm{Recall}_{\mathrm{CAMOD}} = 55{,}15$ % и $F_{1\mathrm{CAMOD}} = 58{,}47$ %. Значение комплексного критерия F_1 алгоритма CAMOD (при строгой проверке обнаружения) на 7,2 % больше значения F_1 упомянутого алгоритма [1].

Быстродействие неоптимизированной версии CAMOD при выполнении на PC Intel Core i7-6700 на наборе данных UAVDT равно 49,2 кадрам в секунду.

Заключение. Тестирование построенного алгоритма CAMOD показало возможность его применения для обнаружения движущихся объектов с помощью вычислителей без мощных средств параллельных вычислений, таких как GPU и NPU.

Недостатком алгоритма является его относительно низкая точность (precision) при обнаружении движения на глубоких сценах, например, когда видео получено камерой низколетящего над городом БЛА, из-за проявляющегося в этом случае параллакса.

Список использованных источников

1. Chapel, M.-N. Moving Objects Detection with a Moving Camera: A Comprehensive Review / M.-N. Chapel, T. Bouwmans // Computer Science Review. – 2020. – Vol. 38. – Art. 100310. https://doi.org/10.1016/j.cosrev.2020.100310

2. Motion Detection [Electronic resource]. – Mode of access: https://paperswithcode.com/task/motion-detection. – Date of access: 25.05.2022.

3. 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

4. Gonzales, R. C. Digital Image Processing / R. C. Gonzales, R. E. Woods. – 4 ed. – Pearson/Prentice-Hall, 2018. – 1192 p.

5. Wojke, N. Simple online and realtime tracking with a deep association metric / N. Wojke, A. Bewley, D. Paulus // ICIP'17: Proceedings of 2017 IEEE International Conference on Image Processing. – 2017. – P. 3645–3650. https://doi.org/10.1109/icip.2017.8296962

References

1. Chapel M.-N., Bouwmans T. Moving Objects Detection with a Moving Camera: A Comprehensive Review. *Computer Science Review*, 2020, vol. 38, art. 100310. https://doi.org/10.1016/j.cosrev.2020.100310

2. *Motion Detection.* Available at: https://paperswithcode.com/task/motion-detection (accessed 25 May 2022).

3. 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 Russian). https://doi.org/10.37661/1816-0301-2021-18-2-83-97

4. Gonzales R. C., Woods R. E. *Digital Image Processing. Forth Edition*. Pearson/Prentice-Hall. 2018. 1192 p.

5. Wojke N., Bewley A., Paulus D. Simple online and realtime tracking with a deep association metric. *ICIP'17: Proceedings of 2017 IEEE International Conference on Image Processing.* 2017, pp. 3645–3650. https://doi.org/10.1109/icip.2017.8296962

Информация об авторе

Залесский Борис Андреевич – д-р физ.-мат. наук, заведующий лабораторией. Объединенный институт проблем информатики НАН Беларуси (ул. Сурганова, 6, 220012, Минск, Республика Беларусь). E-mail: zalesky@newman.bas-net.by.

Information about the author

Zalesky Boris A. – D. Sc. (Physics and Mathematics), Head of the Laboratory. United Institute of Informatics Problems of the National Academy of Sciences of Belarus (6, Surganov Str., 220012, Minsk, Republic of Belarus). E-mail: zalesky@newman.bas-net.by.