

20. *Andreev V.P., Kim V.L.* Razrabotka funktsional'nykh uzlov geterogenного modul'nogo mobil'nogo robota [Development of functional nodes of a heterogeneous modular mobile robot], *Ekstremal'naya robototekhnika (ER-2017): Tr. mezhdunarodnoy nauchno-tekhnicheskoй konferentsii* [Extreme robotics (ER-2017): Proceedings of the international scientific and technical conference]. Saint Peetersburg: ООО «AP4Print», 2016, pp. 359-369.
21. *Yurevich E.I.* Osnovy robototekhniki: uchebnyk dlya vtuzov [Fundamentals of robotics: a textbook for higher education institutions]. Leningrad: Izd-vo Mashinostroenie, Leningr. otd-nie, 1985, 271 p.
22. *Siciliano B. Sciacivco L. [et al.]*. Robotics Modelling, Planning and Control. Springer-Verlag London Limited, 2009, 632 p.
23. *Siegwart R., Nourbakhsh I., Scaramuzza D.* Introduction to Autonomous Mobile Robots. Cambridge: The MIT Press, 2004, 472 p.
24. Tekhnicheskie kharakteristiki servisnogo robota Care-O-bot 4 [Technical characteristics of the Care-O-bot 4 service robot]. Available at: <https://www.care-o-bot.de/de/care-o-bot-4/technical-data.html/> (accessed 26 February 20).
25. Kiva Systems: Three Engineers, Hundreds of Robots, One Warehouse. Available at: <https://spectrum.ieee.org/robotics/robotics-software/three-engineers-hundreds-of-robots-one-warehouse> (accessed 26 February 20).
26. Servisnyy gumanoidnyy robot Rollin' Justin [Service humanoid robot Rollin' Justin]. Available at: <https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-11427/#gallery/29202> (accessed 26 February 20).

Статью рекомендовал к опубликованию д.т.н. Ю.В. Подураев.

Андреев Виктор Павлович – Московский государственный технологический университет «СТАНКИН» (МГТУ «СТАНКИН»); e-mail: andreevvipa@yandex.ru; 127055, Москва, Вадковский пер., 1; тел.: +79652107951; д.т.н.; к.ф.-м.н.; профессор; кафедра сенсорных и управляющих систем.

Ким Валерий Леонидович – e-mail: top7733@gmail.com; тел.: +79153041436; кафедра робототехники и мехатроники; аспирант; ассистент.

Эприков Станислав Робертович – Институт прикладной математики им М.В. Келдыша Российской академии наук; e-mail: stanislav.eprikov@gmail.com; 125047, г. Москва, Миусская пл., 4; тел.: +79175254747; аспирант.

Andreev Victor Pavlovich – Moscow State University of Technology «STANKIN» (MSTU «STANKIN»); e-mail: andreevvipa@yandex.ru; 1, Vadkovsky per., Moscow, 127055, Russia; phone: +79652107951; the department of sensory and drive systems; dr. of eng. sc.; professor.

Kim Valerii Leonidovich – e-mail: top7733@gmail.com; phone: +79153041436; the department of robotics and mechatronics; postgraduate student; assistant.

Eprikov Stanislav Robertovich – The Keldysh Institute of Applied Mathematics; e-mail: stanislav.eprikov@gmail.com; 4, Miusskya pl., Moscow, 125047, Russia; phone: +79175254747; postgraduate student.

УДК 004.932.2

DOI 10.18522/2311-3103-2020-1-218-232

**В.А. Бондаренко, Д.К. Ельцова, А.И. Лизин, В.А. Павлова, М.В. Созинова,
В.А. Тупиков**

МНОГОАГЕНТНЫЙ АЛГОРИТМ АВТОМАТИЧЕСКОГО ОБНАРУЖЕНИЯ И СОПРОВОЖДЕНИЯ НЕДЕТЕРМИНИРОВАННЫХ ОБЪЕКТОВ

В целях разработки робастного алгоритма автоматического обнаружения и сопровождения недетерминированных объектов для встраиваемых вычислительных систем с оптико-электронными устройствами. В рамках данной работы произведено исследование и анализ имеющегося мирового научно-технического опыта в области алгоритмов авто-

матического сопровождения общего назначения. Наиболее успешные из исследованных алгоритмов, подходящие для долговременного устойчивого автоматического сопровождения объектов (без априорного знания о типе объекта слежения) на сегодняшний день уже вышли за рамки решения задачи исключительно сопровождения, и включают в себя синергетическое сочетание нескольких разнородных алгоритмов сопровождения, а также как минимум один алгоритм автоматического обнаружения и/или классификации. В статье показано что наиболее устойчивые современные алгоритмы автоматического сопровождения представляют собой многоагентную систему, принимающую решение о текущем положении, размерах и других параметрах сопровождаемого образа на основе интеллектуального голосования составляющих систему модулей, осуществляющих самостоятельное слежение за объектом и формирование его модели. Индивидуальные модели каждого из модулей уточняются по результатам принятия коллективного решения. Авторами исследования выделены наиболее эффективные из применяемых базовых алгоритмов, подходящие для применения во встраиваемых вычислительных системах робототехнических комплексов, и разработан новый многоагентный алгоритм автоматического обнаружения и сопровождения недетерминированных объектов. Представленный многоагентный алгоритм включает в себя модуль выделения и сопоставления ключевых точек на изображениях, модуль кластеризации и фильтрации ключевых точек с применением алгоритма DBSCAN, модуль сопровождения на основе алгоритма вычисления оптического потока и модуль классификации ключевых точек. Проведено полунатурное тестирование разработанного алгоритма и оценена его эффективность в решении задач не только автоматического сопровождения объектов, но и задач автоматического обнаружения объектов по нескольким эталонным образам. В заключении представлены предложения по дальнейшему повышению точности разработанного алгоритма и по его оптимизации и внедрению в состав специального программного обеспечения бортовых вычислительных систем летательных аппаратов.

Многоагентный алгоритм; автоматическое обнаружение; автосопровождение; ключевые точки; оптический поток; встраиваемые системы.

**V.A. Bondarenko, D.K. Eltsova, A.I. Lizin, V.A. Pavlova, M.V. Sozinova,
V.A. Tupikov**

MULTI-AGENT ALGORITHM FOR AUTOMATIC DETECTION AND TRACKING OF NON-DETERMINISTIC OBJECTS

In order to develop a robust algorithm for the automatic detection and tracking of non-deterministic objects for embedded computing systems, in this work, a study and analysis in the field of state-of-the-art general-purpose automatic tracking algorithms is performed. The most successful of those algorithms suitable for long-term stable automatic tracking of objects (without a priori knowledge of the type of object being tracked) have already gone beyond solving exclusively tracking problems, and include a synergistic combination of several heterogeneous tracking algorithms, as well as at least one automatic detection and / or classification algorithm. Thus, the authors of the article conclude that the most stable modern automatic tracking algorithms are a multi-agent system that makes a decision about the current position, size and other parameters of the tracked object image based on intelligent voting of system's submodules that independently monitor the object and form its model. Individual models of each of the submodules are updated based on the results of a collective decision. The authors of the study identified the most effective of the applied basic algorithms suitable for use in embedded computing systems of robotic systems, and developed a new multi-agent algorithm for the automatic detection and tracking of non-deterministic objects. The presented multi-agent algorithm includes a submodule for extracting and matching key points in images, a clustering and filtering submodule for key points using the DBSCAN algorithm, a tracking submodule based on the optical flow calculation algorithm, and a key point classification submodule. A semi-natural testing of the developed algorithm was carried out and its effectiveness in solving tasks not only of automatic tracking of objects, but also in tasks of automatic objects detection using several reference images were evaluated. In conclusion, the authors present steps for further improving the accuracy and performance of the developed algorithm for its forthcoming implementation for on-board computing systems of aerial vehicles.

Multi-agent algorithm; automatic detection; auto tracking; key points; optical flow; embedded systems.

Введение. В современных системах технического зрения всё ещё остро стоит проблема автоматического сопровождения недетерминированных объектов в сложных условиях наблюдения. Ни один из вариантов решения данной задачи, представленных в общедоступных научных трудах, не гарантирует уверенного долговременного сопровождения, однако анализ тенденций в разработке новых алгоритмов позволяет сделать вывод, что они чаще всего представляют собой группу разнородных связанных алгоритмов обработки изображений, осуществляющих параллельную, согласованную во времени обработку изображения, его анализ, и дальнейшее принятие совместного решения о положении, размерах и иных параметрах сопровождаемого объекта. В ходе обзора современных алгоритмов выделены три наиболее эффективных на данный момент многоагентных алгоритма: TLD [1], CMT [2] и STRUCK [3]. Эти алгоритмы показывают высокую устойчивость к частичным или полным перекрытиям объекта слежения, а также к различным постепенным изменениям масштаба, разворота и ракурса наблюдения объекта.

Алгоритм TLD основан на использовании подхода Сопровождение-Обучение-Обнаружение (Tracking-Learning-Detection). Помимо краткосрочного сопровождения объекта от кадра к кадру, производится обнаружение всех локальных появлений объекта и на их основе строится модель объекта слежения. Модуль обучения призван снизить ошибки работы алгоритма, путем коррекции модели по результатам сопровождения. Однако, согласно данному алгоритму, обнаружение производится по всему кадру изображения с окнами разных масштабов, а для обучения используются вейвлеты Хаара. Это существенно увеличивает вычислительную сложность алгоритма и не позволяет использовать его в системах реального времени.

Алгоритм CMT (Consensus-based Matching and Tracking) основан на сопровождении объекта по ключевым точкам. Обнаружение производится посредством голосования каждой отобранной точки за центр объекта. За счет использования быстрых детекторов и бинарных дескрипторов, алгоритм способен работать в реальном масштабе времени, а отказ от обновления информации об объекте слежения (сравнение осуществляется с первоначально составленным эталонным описанием) снижает риск возникновения ошибок при сопровождении.

Однако, отказ от обновления информации об объекте является основным недостатком этого алгоритма, поскольку в реальных условиях наблюдения взятые на сопровождение объекты могут претерпевать сильные масштабные, поворотные и перспективные изменения.

Алгоритм Struck (Structured Output Tracking with Kernels) использует для классификации метод опорных векторов (SVM), который обучается в процессе работы для обеспечения адаптивного сопровождения. Алгоритм предполагает механизм бюджетирования, который предотвращает неограниченный рост числа применяемых для обучения векторов признаков сопровождаемого объекта.

Основные недостатки – это достаточно высокая вычислительная сложность, а также относительная низкая устойчивость к шуму, так как у алгоритма отсутствуют методы отбора применяемых для обучения признаков.

На основе анализа сильных сторон описанных алгоритмов авторами был разработан новый многоагентный алгоритм автоматического обнаружения и сопровождения объектов, позволяющий осуществлять устойчивое автоматическое сопровождение объектов в различных условиях наблюдения, сохраняя при этом возможность применения во встраиваемых системах в режиме реального времени.

Применяемый подход. На вход реализованного алгоритма подаются видеопоследовательность I, \dots, I_n и описывающий прямоугольник, задающий первоначальное положение объекта для сопровождения (захват на сопровождение). Основная задача алгоритма сопровождения – автоматическое нахождения положения

объект интереса на дальнейшей последовательности изображений, если объект не найден на протяжении указанного времени, сообщить оператору об этом и/или запустить алгоритм обнаружения объекта. На рис. 1 представлена общая блок-схема разработанного многоагентного алгоритма автоматического сопровождения.

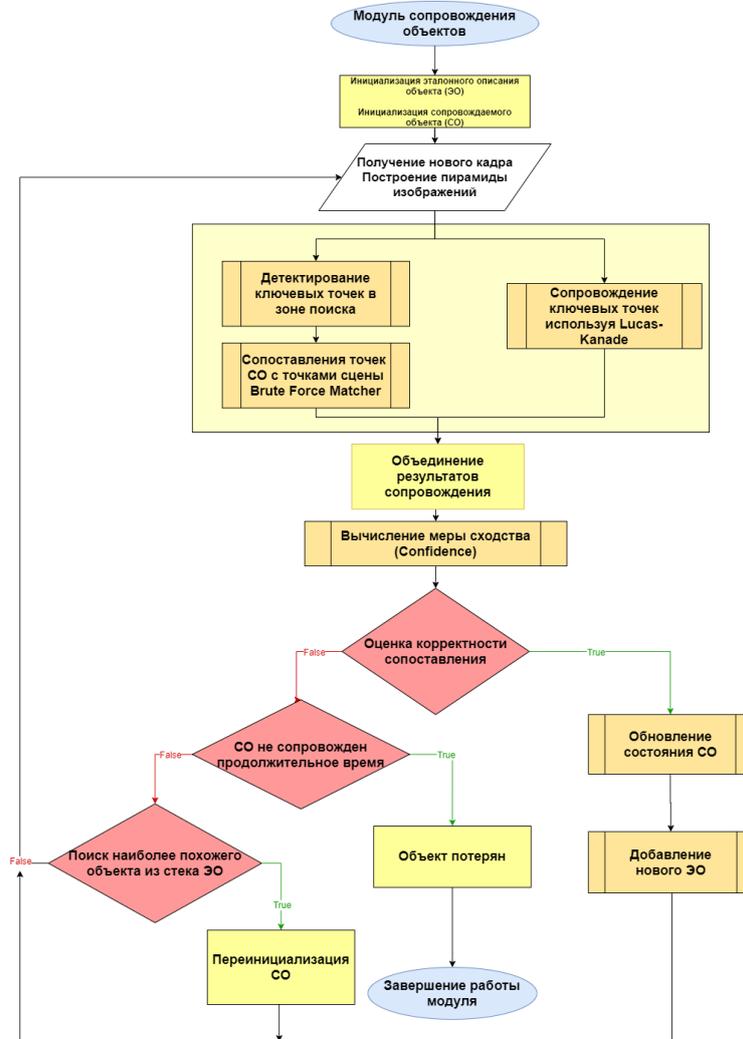


Рис. 1. Общая блок-схема разработанного алгоритма сопровождения

Рассмотрим каждый шаг алгоритма подробнее.

Пирамида изображений. Пирамидой изображений называется сгруппированный набор кратномасштабных изображений сцены. Использование пирамидальной структуры для выделения и сопоставления ключевых точек имеет следующие полезные свойства:

1. Сокращение времени обработки изображений.
2. Определение более точных начальных приближений для обработки нижних уровней по результатам обработки верхних уровней.
3. Масштабная инвариантность получаемого описания.
4. Иерархическая структура объекта.

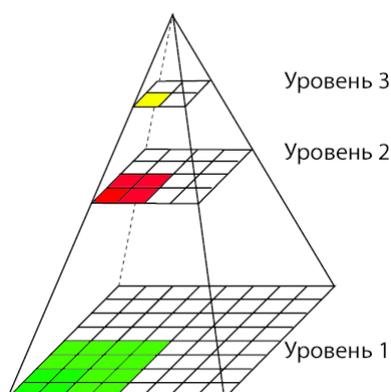


Рис. 2. Схематическое представление пирамиды изображений

Принцип построения пирамиды изображения представлен на рис. 2. Такая пирамида является последовательностью из N изображений. Каждый последующий уровень пирамиды представляет собой сжатое в K раз изображение предыдущего уровня (в применённом подходе $K = [1.2-1.5]$, $N = [8-10]$). Представленная пирамида используется для детекции ключевых точек и вычисления оптического потока, о котором будет рассказано далее.

Получив пирамиду изображений, появляется возможность воспользоваться ее свойствами для получения ключевых точек и их описания. В ходе анализа алгоритмов детектирования ключевых точек были рассмотрены [4] следующие алгоритмы:

- ◆ SIFT;
- ◆ ORB;
- ◆ Harris;
- ◆ Fast.

Детектор ORB оказался лучшим, среди рассмотренных, решением по отношению скорости работы к точности.

Детектор и дескриптор ORB. Основной используемой группой алгоритмов стали *детекторы ключевых точек*. Алгоритмы этой группы находят ключевые точки на изображении и описывают их так называемыми *дескрипторами*, позволяя отличить одну точку от другой. Главным преимуществом таких алгоритмов является существенное снижение обрабатываемой информации. В работах [5, 6] сформулированы основные требования к ключевым точкам. Наилучшие результаты данные алгоритмы демонстрируют на пирамиде изображений.

ORB (Oriented FAST and Rotated BRIEF) является развитием алгоритмов Fast и BRIEF [7]. Основные этапы алгоритма можно описать следующим образом:

1. Особые точки обнаруживаются на исходном изображении и на нескольких изображениях из пирамиды уменьшенных изображений при помощи быстрого древовидного алгоритма FAST.
2. Для обнаруженных точек вычисляется мера Харриса, кандидаты с низким значением меры Харриса отбрасываются.
3. Вычисляется угол ориентации особой точки.
4. Последовательность точек для бинарных сравнений в дескрипторе BRIEF поворачивается в соответствии с вычисленным углом.
5. По полученным точкам вычисляется бинарный дескриптор BRIEF.

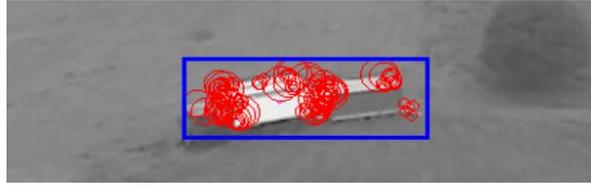


Рис. 3. Пример выделения ключевых точек

Как было представлено на блок-схеме, на первом этапе сопровождения на вход подаются координаты описывающего прямоугольника объекта. В данной области, используя детектор ORB, производится обнаружение ключевых точек (рис. 3). Полученные ключевые точки являются эталонным описанием объекта и будут использованы при дальнейшем сопровождении.

Поскольку было получено описание объекта и его положение на кадре, можно предположить, что на следующем кадре видеопотока сопровождаемый объект все еще находится в некоторой окрестности от предыдущего положения. Поэтому при последующем сопровождении, поиск объекта будет производиться не по всему кадру, а только в некоторой зоне (размер которой может корректироваться в процессе сопровождения на основе анализа скорости и направления движения объекта), что существенно уменьшает время работы детектора ключевых точек, а также уменьшает накладные расходы для построения пирамиды изображений.

Для уменьшения количества ключевых точек, не относящихся к объекту, был применен алгоритм кластеризации DBSCAN, позволяющий значительно снизить количество ключевых точек, не принадлежащих к объекту, что в свою очередь ускоряет сопоставление ключевых точек и повышает его точность.

Алгоритм DBSCAN. Данный подход эффективно разбивает наборы точек на группы по принципу минимального количества соседей и плотности распределения [8]. Большим отличительным плюсом данного алгоритма является отсутствие необходимости задания количества кластеров заранее.

Опишем подход более формально:

Пусть задано множество объектов X , для которых определена некоторая функция расстояния $\rho(x,y)$ и константы ϵ – максимальное расстояние между соседними объектами и m – количество соседей. Тогда:

1. Существует область $E(x)$, для которой $\forall u \in X : \rho(x,u) \leq \epsilon$. Такая область называется ϵ -окрестность объекта x .
2. Корневым объектом называется объект $p \in X$, если его ϵ -окрестность содержит не менее m объектов: $|E(p)| \geq m$.
3. Объект $q \in X$ непосредственно плотно-достижим из объекта p , если $q \in E(p)$ и p – корневой объект.
4. Объект $q \in X$ называется достижимым из объекта p , если существует путь p_1, p_2, \dots, p_n , где $p_1 = p, p_n = q$, такой, что $\forall i \in 1 \dots n - 1: p_{i+1}$ непосредственно плотно достижим из p_i .

Алгоритм выбирает необработанный корневой объект p из стека точек, помечает его и помещает всех его непосредственно плотно-достижимых соседей в список обхода. Далее для каждого объекта q из списка: помечается эта точка, и, если она тоже корневая, все её соседи добавляются в список обхода. Легко заметить, что кластеры помеченных точек, сформированные в ходе работы этого алгоритма, максимальны (т.е. их нельзя расширить еще одной точкой, чтобы удовлетворялись условия) и связаны в смысле плотно-достижимости. Отсюда следует, что, если не все точки были пройдены, можно перезапустить обход из какого-нибудь другого корневого объекта, и новый кластер не поглотит предыдущий. На рисунке 4 приведен пример визуализации результатов работы алгоритма кластеризации DBSCAN.

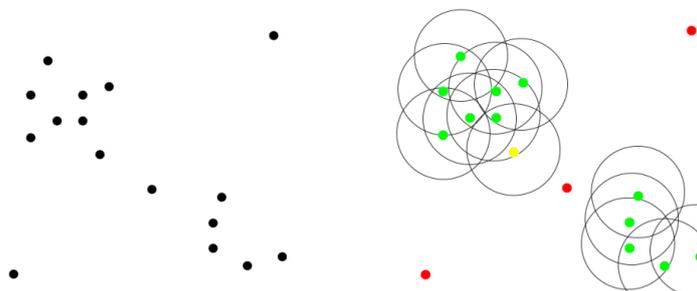


Рис. 4. Визуализация работы алгоритма кластеризации DBSCAN

Точки, не принадлежащие ни одному кластеру после работы алгоритма (помеченные особым образом) удаляются из списка, что при составлении эталонного описания сопровождаемого объекта позволяет существенно сократить количество обрабатываемых точек впоследствии.

Взятое эталонное описание сопровождаемого объекта берётся на сопровождение, по результатам которого эталонное описание обновляется. Сопровождение осуществляется с применением комбинации из трёх алгоритмов:

1. Brute Force Matching – сопоставление ключевых точек объекта и наблюдаемой сцены.
2. Lucas-Kanade – вычисление оптического потока для точек сопровождаемого объекта.
3. DBSCAN – кластеризация результатов голосования за центр сопровождаемого объекта точек, прослеживаемых методами 1 и 2.

Также применяется некоторая дополнительная обработка результатов данных алгоритмов.

Brute Force Matcher. Brute Force Matcher [9] (BFM) является популярным способом сопоставления ключевых точек между собой. Алгоритм путем простого перебора всех точек находит наиболее похожий дескриптор. Такой подход никак не учитывает положение ключевой точки на сцене, поэтому довольно часто дает ложные совпадения. Предлагается несколько модификаций данного алгоритма:

- ◆ Введение штрафной функции. Чем дальше находится точка от ожидаемых координат, тем сильнее штраф на ее меру схожести.
- ◆ Голосование за центр объекта. Все сопоставленные точки, принадлежащие объекту, должны проголосовать за центр объекта (относительная дистанция до центра объекта хранится в описании сопровождаемой ключевой точки). Используя алгоритм DBSCAN, требуется отсеять некорректные голоса.

Обе модификации в ходе тестирования показали высокую степень корректного сопоставления точек объекта и наблюдаемой сцены, однако второй подход показал себя более удачным при сопровождении объектов, движущихся с непостоянной скоростью.

Lucas-Kanade. Алгоритм Lucas-Kanade (LK) - широко распространённый алгоритм, позволяющий вычислить оптический поток [10]. Оптический поток – определение вектора перемещения указанной области изображения относительно её предыдущего положения, представляющее собой сдвиг каждой точки между двумя изображениями. Метод основан на предположении, что в локальной окрестности каждого пикселя значение оптического потока одинаково или изменяется мало, таким образом можно записать основное уравнение (1) оптического потока для всех пикселей окрестности и решить полученную систему (2) уравнений методом наименьших квадратов.

$$M\vec{u} = \vec{b} \quad (1)$$

или

$$\begin{pmatrix} I_x(x_1, y_1) & I_y(x_1, y_1) \\ * & * \\ * & * \\ * & * \\ I_x(x_n, y_n) & I_y(x_n, y_n) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -I_t(x_1, y_1) \\ * \\ * \\ * \\ -I_t(x_1, y_1) \end{pmatrix}. \quad (2)$$

Сопровождение ключевых точек объекта методами Brute Force Matcher и Lucas-Kanade на данном кадре осуществляется независимо друг от друга, после результаты работы алгоритмов объединяются с применением процедуры голосования за центр цели. Из двух результатов выбирается наиболее точный результат, если дистанция между голосами $d \leq D_n$, в противном случае предпочтение отдаётся BFM.

Объединение результатов работы (процедура Consensus) разнородных алгоритмов, позволяет воспользоваться их преимуществами (высокая точность сопровождения LK на малых временных отрезках) и нивелировать их недостатки («уплывание» точек, сопровождаемых LK, корректируется BFM). Благодаря комбинации трех подходов во время сопровождения имеется возможность отсекаать точки, не принадлежащие объекту.

Мера схожести. Получив новые координаты ключевых точек сопровождаемого объекта, необходимо оценить корректность полученного результата и оценить общую степень уверенности системы в том, что объект удалось сопроводить. Предлагается вычислять меру уверенности (Confidence), показывающую на сколько удачно удалось сопроводить объект. Предложенная мера состоит из 6 параметров и соответствующей весовой функции:

1. Средняя схожесть сопоставленных точек (при сравнении дескрипторов в BFM).
2. Количество успешно сопровождаемых точек (по отношению к общему количеству сопровождаемых точек).
3. Оценка меры сходства инвариантных *Hu* моментов вычисляемых на наборе успешно сопоставленных ключевых точек (по сравнению с предыдущим значением).
4. Вероятность того, что точки принадлежат сопровождаемому объекту, полученная от Байесовского классификатора (обучаемого в процессе сопровождения по бинарным дескрипторам ключевых точек).
5. Схожесть площади выпуклого многоугольника, описывающего успешно сопровождаемые точки.
6. Схожесть периметра выпуклого многоугольника, описывающего успешно сопровождаемые точки.

Оцениваемые меры взвешиваются весовой функцией и суммируются, в сумме данные параметры должны составлять единицу для полного совпадения. Некоторые из оцениваемых параметров (*Hu* моменты, геометрические соотношения прямоугольников) являются менее точными при малых размерах сопровождаемого объекта и становятся значительно точнее при увеличении его размеров.

Применение весовой функции позволяет динамически изменять вклад оцениваемых параметров в определение меры сходства, что позволяет отдавать больший вклад нужным параметрам при изменении размеров объекта и в целом повышает устойчивость получаемой оценки меры схожести.

Ну моменты. Инвариантные моменты представляют собой комбинации нормализованных пространственных моментов с порядком до третьего включительно. Они остаются неизменными при перемещении, повороте и масштабировании изображения. В общем случае момент изображения вычисляется как:

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) dx dy ,$$

где $p, q = 0, 1, 2, 3 \dots n$; $f(x, y)$ – функция яркости; x, y – координаты.

Алгоритм Naive Bayes Classifier. Наивный байесовский классификатор [11] – вероятностный классификатор, использующий в своей основе теорему Байеса со строгими (наивными) предположениями о независимости. Классическая область применения данного алгоритма – фильтрация писем от спама.

Основными причинами выбора данного классификатора является малое количество данных необходимых для обучения, оценки параметров, классификации и возможность обучаться во время сопровождения объекта.

В основе классификатора лежит теорема Байеса:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)},$$

где, $P(c|d)$ – вероятность, что d принадлежит классу c , $P(d|c)$ – вероятность встретить d среди всей выборки класса c , $P(c)$ – безусловная вероятность встретить класс c в общей выборке, $P(d)$ – безусловная вероятность d в общей выборке. При классификации основная задача – это определить наиболее вероятный класс объекта (в случае сопровождения – объект/не объект). Байесовский классификатор использует оценку апостериорного максимума для определения наиболее вероятного класса. Опуская вывод итоговой формулы, получаем:

$$c = \operatorname{argmax} c \in C \left[\log \log \frac{D_c}{D} + \sum_{i=1}^n \frac{W_{ic}}{|V| + \sum_{i' \in V} W_{i'c}} \right], \quad (3)$$

где, D_c – количество объектов, принадлежащих классу c , а D – общее количество объектов в обучающей выборке, W_{ic} – количество сколько раз i -ый признак встречается в классе c , V – список всех уникальных объектов.

В простейшем случае выбирается тот класс, который получил наибольшую оценку. Применяя представленный подход к классификации, во время сопровождения будут распознаваться два класса: объект и не объект. Под не объектом будем понимать все сторонние структуры вне прямоугольника, описывающего сопровождаемый объект. Используя формулу (3), появляется возможность на следующем поступившем кадре предположить, какой объект какого класса алгоритм пытается сопроводить.

Алгоритм Convex hull. Выпуклой оболочкой (Convex hull) называют наименьшее выпуклое множество, содержащее множество X . Данный алгоритм позволяет более корректно оценивать геометрические изменения объекта, чем простой описывающий прямоугольник, который часто используется в алгоритмах сопровождения.

Алгоритм состоит из следующих шагов:

1. Определяется точка p_0 множества с самой маленькой y -координатой (если таких несколько – выбирается самая правая из них), добавляется в оболочку.
2. Все оставшиеся точки сортируются по значению полярного угла относительно p_0 .
3. Самая первая из отсортированных точек p_1 добавляется в оболочку.

4. Анализируется следующая по счету точка t . Пока t и две последних точки в текущей оболочке p_i и p_{i-1} образуют неправый поворот точка p_i удаляется из оболочки.

5. Точка t добавляется в оболочку.

6. Пункты 4-5 выполняются до тех пор, пока не закончатся точки.

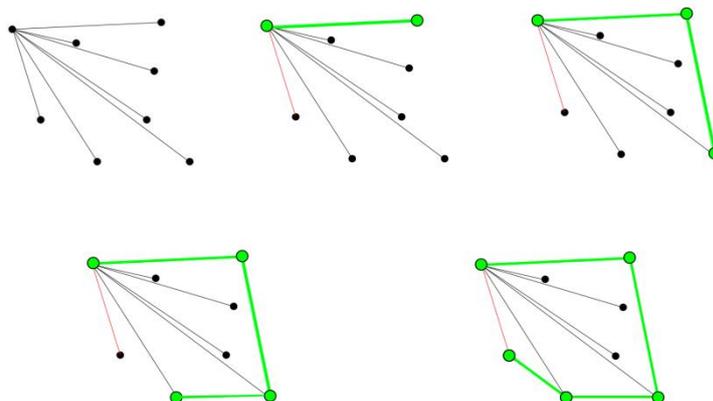


Рис. 5. Пример построения выпуклой оболочки вокруг набора точек

На рис. 5 представлен пример построения выпуклой оболочки вокруг набора точек. Для полученного таким образом многоугольника можно рассчитать его площадь и периметр, что дает большую точность чем обычный описывающий прямоугольник. Вычисленные значения можно использовать при определении меры уверенности алгоритма.

Оценка корректности сопоставления. Оценка корректности сопоставления точек одна из важных частей разработанного многоагентного алгоритма. От ее правильности зависят дальнейшие этапы работы алгоритма. Рассчитанная ранее мера уверенности является достаточно точной мерой оценки, но поскольку она рассчитывается на основе сопоставленных точек она может давать некорректные результаты. По этой причине стоит ввести несколько дополнительных условий по которым будет выноситься решение об успешности сопровождения. К таким условиям относятся:

- ◆ Количество сопоставленных точек.
- ◆ Соотношение сопровождаемых точек к сопоставленным.
- ◆ Разница между предыдущей мерой уверенности и текущей.

Принимая во внимание представленные условия, но в большей степени опираясь на меру схожести, можно корректно оценить правильность сопровождения.

Обновление модели сопровождаемого объекта. В процессе обновления модели сопровождаемого объекта у точек, которые удалось сопроводить, обновляются координаты, дескрипторы и параметры движения. Точки, которые не удалось сопроводить перемещаются на новые координаты на среднее смещение объекта (если подобные точки не обновлялись достаточно давно, они удаляются из стека точек). Обновляются также ну моменты набора ключевых точек объекта, классификатору подаются новые данные для обучения.

Также, поскольку наблюдаемый объект со временем видоизменяется, необходима процедура не только обновления текущих точек объекта, но и добавления новых точек со сцены.

Поиск и добавление кандидатов. Данная процедура пытается добавить к сопровождаемому набору точек объекта точки со сцены. На первом этапе требуется найти возможные точки кандидаты для сопровождаемого объекта. В описываемом прямоугольнике сопровождаемого объекта, увеличенного на 10 % от наименьшей стороны, запоминаются точки сцены, не относящиеся к сопровождаемому стеку.

Сохраненные точки спустя определенный отрезок времени сопоставляются с обновленной сценой. Корректно сопоставленные точки добавляются в кандидаты на добавление (в набор точек объекта) и отслеживаются параллельно с сопровождаемым объектом. Сопровождение кандидатов происходит по той же логике, как и сопровождение объекта.

Процедура добавления ключевых точек из кандидатов различна для объектов разного размера. Для больших объектов производится кластеризация точек объекта совместно с точками кандидатами. Из кластера, где соотношение точек кандидатов к точкам объекта является наибольшим, добавляются точки кандидаты, которые сопровождалась дольше чем другие, в стек сопровождаемого объекта. Для малых объектов добавление происходит путем нахождения той части объекта, где множество сопровождаемых точек объекта наиболее разрежено. Таким образом, точки сцены, движущиеся вместе с объектом, могут быть добавлены к сопровождаемому объекту, при условии, что сопровождаемых точек в наборе становится недостаточно, а их текстурные характеристики удовлетворяют критериям классификации Bayes Classifier.

Добавление нового эталона. Использование набора (стека) эталонных описаний объекта позволяет при потере объекта производить его переобнаружение, используя уже описанные выше алгоритмы и меры оценки. Если мера схожести между последним эталоном и текущим состоянием объекта достаточно высока (но при этом в достаточной мере изменился периметр или площадь), имеет смысл добавить данный сопровождаемый объект как новый эталон. Максимальное количество эталонов в стеке ограничивается в соответствии с доступной производительностью применяемой вычислительной системы.

Потеря объекта. Стоит отметить, что при потере объекта трудно спрогнозировать ожидаемое положение объекта. Поэтому с каждым кадром, на котором не удалось найти объект, требуется постепенно увеличивать зону поиска. В ходе разработки алгоритма было принято решение не превышать зону поиска объекта более чем на 80 % от площади кадра, для экономии вычислительных ресурсов.

В данной процедуре с каждой новой итерацией поиска перебирается накопленный список эталонов, которые удалось добавить в стек в ходе работы алгоритма сопровождения, и эталоны по очереди сопоставляются с наблюдаемой сценой до достижения наилучшего соответствия, оцениваемого по мере уверенности, описанной выше. Поиск объекта будет прекращен по истечении N секунд или после корректного повторного обнаружения (с достаточно высокой мерой сходства).

При корректном обнаружении для сопоставленных точек строится описывающий многоугольник, на основании которого со сцены берутся все ключевые точки, которыми заново инициализируется вновь обнаруженный сопровождаемый объект.

На рис. 6 представлены примеры работы разработанного алгоритма сопровождения (красные точки – точки сопровождаемого объекта, жёлтые – точки-кандидаты, голубые – описывающий многоугольник).

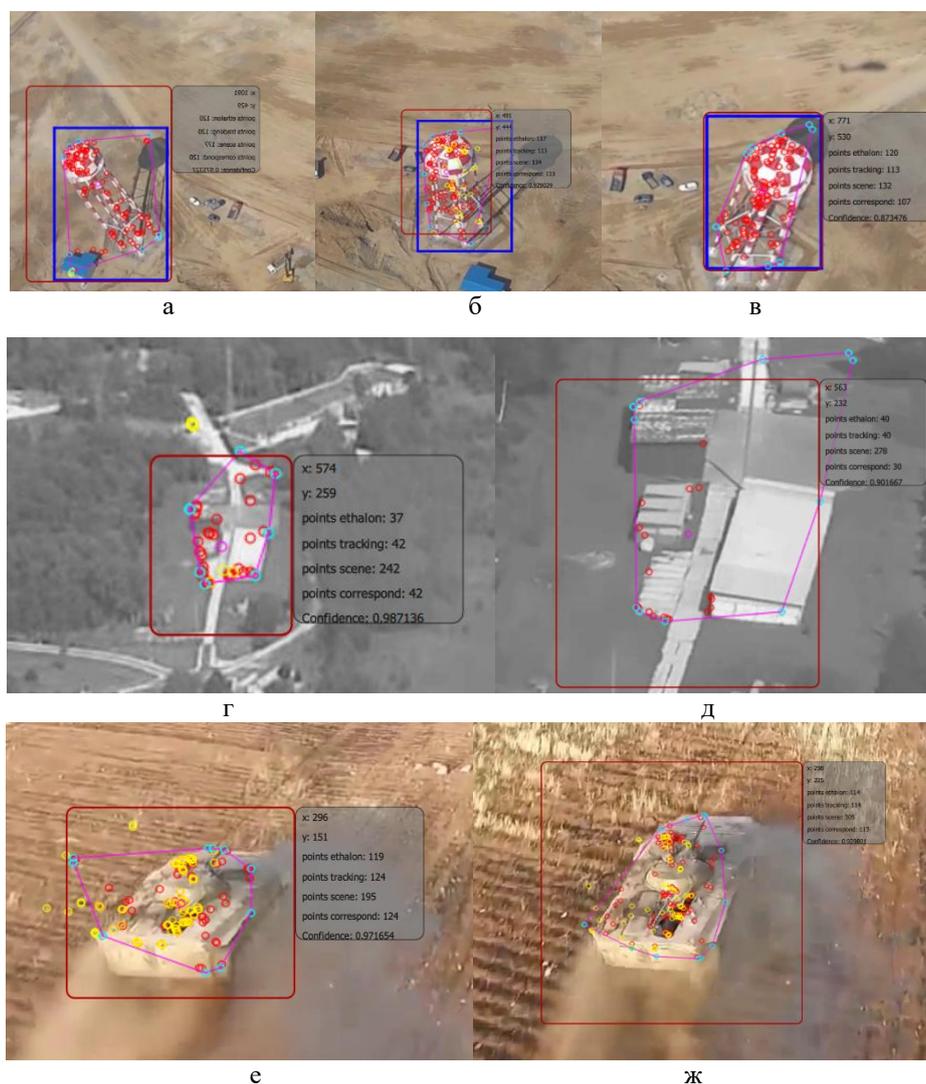


Рис. 6. Пример работы разработанного алгоритма автоматического сопровождения. Слева направо: а–в – этапы сопровождения объекта типа «Вышка»; г, д – этапы сопровождения группы объектов типа «Здание»; е, ж – этапы сопровождения объекта типа «БМП»

Заключение. Резюмируя изложенное, можно сказать, что разработанный многоагентный алгоритм позволяет выполнять высокоскоростную обработку изображений в реальном масштабе времени, что дает возможность использовать данный алгоритм, например, в составе специального программного обеспечения оптико-электронных систем беспилотных и пилотируемых летательных аппаратов.

Использование предложенного многоагентного алгоритма показало высокие результаты в сравнении конкурирующими подходами и с классическими алгоритмами на основе сравнения по шаблону для средне и крупноразмерных объектов.

На следующем этапе развития алгоритма планируется разработать и внедрить алгоритмы с применением KD-деревьев для хранения и сопоставления ключевых точек, что позволит в значительной степени увеличить скорость работы алгоритма. Также

перспективным является улучшение в части применения классификаторов другого типа. Наиболее перспективным направлением видится использование «Мешка слов» (Bag-of-Words) и вычисления гистограммы ориентированных градиентов (Histogram of oriented Gradients).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas.* Tracking-Learning-Detection // IEEE transactions on pattern analysis and machine intelligence. – January 2010. – Vol. 6, No. 1.
2. *Georg Nebehay, Roman Pflugfelder.* Consensus-based Matching and Tracking of Keypoints for Object Tracking. Safety and Security Department. AIT Austrian Institute of Technology, Austria, 2014.
3. *Sam Hare, Amir Saffari, Philip H.S. Torr.* Struck: Structured Output Tracking with Kernels. Oxford Brookes University. Oxford, UK 2Sony Computer Entertainment Europe, London, UK. 2014.
4. *Бондаренко В.А., Каплинский Г.Э., Павлова В.А., Туников В.А.* Метод поиска и сопоставления ключевых особенностей изображений для распознавания образов и сопровождения объектов // Известия ЮФУ. Технические науки. – 2019. – № 1 (213). – С. 281-293.
5. *Rodehorst V., Koschan A.* Comparison and evaluation of feature point detectors // Proceedings of 5th International Symposium Turkish-German Joint Geodetic Days, 2006.
6. *Tuytelaars T., Mikolajczyk K.* Local Invariant Feature Detectors – Survey // Foundations and Trends in Computer Graphics and Vision. – 2008. – No. 3(3). – P. 177-280.
7. *Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski.* ORB: an efficient alternative to SIFT or SURF. Willow Garage, Menlo Park, California. 2011.
8. *Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, Xiaowei Xu.* DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN // ACM Trans. Database Syst. – 2017.
9. *Amila Jakubovic, Jasmin Velagic.* Image Feature Matching and Object Detection using Brute-Force Matchers. – Faculty of Electrical Engineering/University of Sarajevo Sarajevo, Bosnia and Herzegovina. 2018.
10. *Lucas B.D. and Kanade T.* An iterative image registration technique with an application to stereo vision // International Joint Conference on Artificial Intelligence. – 1981. – P. 674-679.
11. *Eibe Frank, Remco R. Bouckaert.* Naive Bayes for Text Classification with Unbalanced Classes. Computer Science Department, University of Waikato, New Zealand, Xtal Mountain Information Technology, Auckland, New Zealand. 2006.
12. *Lowe D.G.* Object recognition from local scale-invariant features // Proceedings of the International Conference on Computer Vision. – 1999. – No. 2. – P. 1150-1157.
13. *Sung-Joo Y., Kim T.* Development of Stereo Visual Odometry Based on Photogrammetric Feature Optimization // Remote Sensing. – 2019. – No. 11.
14. *Horn B.K.P. and Schunck B.G.* Determining optical flow // Artificial intelligence. – 1981. – Vol. 17, No. 1-3. – P. 185-203.
15. *Shi J. and Tomasi C.* Good features to track // Conference on Computer Vision and Pattern Recognition. – 1994.
16. *Leutenegger S., Chli M., and Siegwart R.Y.* BRISK: Binary robust invariant scalable keypoints // In ICCV. – 2011.
17. *Shahed Nejhum S.M., Ho J., and Yang M.-H.* Visual tracking with histograms and articulating blocks // In CVPR. – 2008.
18. *Zhang L. and van der Maaten L.* Structure preserving object tracking // In CVPR. – 2013.
19. *Martin Ester, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu.* A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Institute for Computer Science, University of Munich Oettingenstr. 67, D-80538 Miinchen, Germany, 1996.
20. *Taheri S., Mammadov M.* Learning the naive Bayes classifier with optimization models. Centre for Informatics and Applied Optimization School of Science, Information Technology and Engineering, University of Ballarat, Victoria 3353, Australia. 2.
21. *Andrew Moores.* Efficient memory-based learning for robot control. 15 JJ Thomson Avenue Cambridge CB3 0FD United Kingdom. 1990.

REFERENCES

1. Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-Learning-Detection, *IEEE transactions on pattern analysis and machine intelligence*, January 2010, Vol. 6, No. 1.
2. Georg Nebel, Roman Pflugfelder. Consensus-based Matching and Tracking of Keypoints for Object Tracking. Safety and Security Department. AIT Austrian Institute of Technology, Austria, 2014.
3. Sam Hare, Amir Saffari, Philip H.S. Torr. Struck: Structured Output Tracking with Kernels. Oxford Brookes University. Oxford, UK 2Sony Computer Entertainment Europe, London, UK. 2014.
4. Bondarenko V.A., Kaplinskiy G.E., Pavlova V.A., Tupikov V.A. Metod poiska i sopostavleniya klyuchevykh osobennostey izobrazheniy dlya raspoznavaniya obrazov i soprovozhdeniya ob"ektov [Method for searching and matching key features of images for image recognition and object tracking], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2019, No. 1 (213), pp. 281-293.
5. Rodehorst V., Koschan A. Comparison and evaluation of feature point detectors, *Proceedings of 5th International Symposium Turkish-German Joint Geodetic Days, 2006*.
6. Tuytelaars T., Mikolajczyk K. Local Invariant Feature Detectors – Survey, *Foundations and Trends in Computer Graphics and Vision*, 2008, No. 3 (3), pp. 177-280.
7. Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski. ORB: an efficient alternative to SIFT or SURF. Willow Garage, Menlo Park, California. 2011.
8. Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, Xiaowei Xu. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN, *ACM Trans. Database Syst.*, 2017.
9. Amila Jakubovic, Jasmin Velagic. Image Feature Matching and Object Detection using Brute-Force Matchers. Faculty of Electrical Engineering/University of Sarajevo Sarajevo, Bosnia and Herzegovina. 2018.
10. Lucas B.D. and Kanade T. An iterative image registration technique with an application to stereo vision, *International Joint Conference on Artificial Intelligence*, 1981, pp. 674-679.
11. Eibe Frank, Remco R. Bouckaert. Naive Bayes for Text Classification with Unbalanced Classes. Computer Science Department, University of Waikato, New Zealand, Xtal Mountain Information Technology, Auckland, New Zealand. 2006.
12. Lowe D.G. Object recognition from local scale-invariant features, *Proceedings of the International Conference on Computer Vision*, 1999, No. 2, pp. 1150-1157.
13. Sung-Joo Y., Kim T. Development of Stereo Visual Odometry Based on Photogrammetric Feature Optimization, *Remote Sensing*, 2019, No. 11.
14. Horn B.K.P. and Schunck B.G. Determining optical flow, *Artificial intelligence*, 1981, Vol. 17, No. 1-3, pp. 185-203.
15. Shi J. and Tomasi C. Good features to track, *Conference on Computer Vision and Pattern Recognition*, 1994.
16. Leutenegger S., Chli M., and Siegwart R.Y. BRISK: Binary robust invariant scalable keypoints, *In ICCV*, 2011.
17. Shahed Nejhum S.M., Ho J., and Yang M.-H. Visual tracking with histograms and articulating blocks, *In CVPR*, 2008.
18. Zhang L. and van der Maaten L. Structure preserving object tracking, *In CVPR*, 2013.
19. Martin Ester, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Institute for Computer Science, University of Munich Oettingenstr. 67, D-80538 München, Germany, 1996.
20. Taheri S., Mammadov M. Learning the naive Bayes classifier with optimization models. Centre for Informatics and Applied Optimization School of Science, Information Technology and Engineering, University of Ballarat, Victoria 3353, Australia. 2.
21. Andrew Moores. Efficient memory-based learning for robot control. 15 JJ Thomson Avenue Cambridge CB3 0FD United Kingdom. 1990.

Статью рекомендовал к опубликованию к.т.н. С.Н. Крюков.

Тупиков Владимир Алексеевич – АО «Научно-производственное предприятие «Авиационная и морская электроника»; e-mail: tupikov@nppame.ru; 198097, г. Санкт-Петербург, ул. Маршала Говорова, 29, лит. «О», а/я 51; тел.: 88123274667; д.т.н.; профессор, зам. ген. директора; директор научно-производственного комплекса робототехнических систем специального назначения (НПК РТС СН).

Павлова Валерия Анатольевна – e-mail: pavlova@nppame.ru; к.т.н.; зам. директора НПК РТС СН по НИОКР.

Бондаренко Владимир Александрович – e-mail: bondarenko@nppame.ru; начальник центра средств интеллектуальной обработки изображений.

Лизин Алексей Игоревич – e-mail: lizin_a@nppame.ru, научный сотрудник.

Ельцова Дарья Константиновна – e-mail: eltsova@nppame.ru; научный сотрудник.

Созинова Мария Владимировна – e-mail: sozinova@nppame.ru; начальник отдела перспективных методов обработки изображений.

Tupikov Vladimir Alekseevich – SPE "Research and Production Enterprise "Air and Marine Electronics"; e-mail: tupikov@nppame.ru; 198097, Saint Petersburg, street Marshal Govorov, 29, lit. "O", P.O.B 51; phone: 88123274667; dr. of eng. sc.; professor; Deputy Director General; Director of Research and production complex of special purpose robotic systems.

Pavlova Valeria Anatolyevna – e-mail: pavlova@nppame.ru; cand. of eng. sc.; Deputy Director of research at Research and production complex of special purpose robotic systems.

Bondarenko Vladimir Alexandrovich – e-mail: bondarenko@nppame.ru; chief of intellectual image processing center.

Lizin Aleksey Lizin – e-mail: lizin_a@nppame.ru; researcher.

Eltsova Darya Konstantinovna – e-mail: eltsova@nppame.ru; researcher.

Sozinova Maria Vladimirovna – e-mail: sozinova@nppame.ru; chief of the department of Advanced Image Processing Techniques.