

**А.Е. Щелкунов, В.В. Ковалев, И.В. Сидько, Н.Е. Сергеев**

**УСКОРЕНИЕ ПРЯМОГО ПРОХОДА ПРИ РЕАЛИЗАЦИИ СНС  
НА ОГРАНИЧЕННОМ ВЫЧИСЛИТЕЛЬНОМ РЕСУРСЕ**

*Работа посвящена оптимизации архитектуры нейронной сети для ее запуска на ограниченном вычислительном ресурсе. Рассмотрено несколько подходов оптимизации, приведены оценки сложности и времени выполнения прямого прохода нейронной сети. Приведены сравнительные оценки сложности сети при применении разных подходов оптимизации. В работе представлен анализ выбранной архитектуры сети, получены оценки вычислительной сложности отдельных составных частей (модулей) архитектуры. Произведен анализ возможных методов оптимизации каждого модуля. Описаны параметры рассмотренных модулей, размеры входных и выходных тензоров. Для оптимизации модуля выделения признаков протестировано несколько архитектур, ResNet 50, ResNet 18, MobileNet v3 small, MobileNet v3 large. Представлен сравнительный анализ вычислительной сложности и времени выполнения прямого прохода для каждой архитектуры. Замеры времени выполнения прямого прохода осуществлялись на встраиваемом вычислительном устройстве Jetson AGX Xavier от компании Nvidia. Представлены оценки времени выполнения прямого прохода для каждого модуля рассматриваемых нейронных сетей. В работе приведены результаты сравнения оценок точности нейронной сети до и после оптимизации архитектуры. Набор данных для тестирования состоит из 100 видео записей. В тестовых видеозаписях участвует 5 различных типовых объектов, для каждого класса объекта записано 10 различных сценариев. Для каждой из разработанных архитектур получены оценки точности, произведен сравнительный анализ. В работе намечены пути для последующей оптимизации архитектуры сети.*

*Оптимизация выполнения прямого прохода сверточной нейронной сети; трекинг.*

**A.E. Shchelkunov, V.V. Kovalev, I.V. Sidko, N.E. Sergeev**

**ACCELERATION OF THE DIRECT PASSAGE IN THE IMPLEMENTATION  
OF CNN ON A LIMITED COMPUTING RESOURCE**

*The work is devoted to the optimization of the neural network architecture for its launch on a limited computing resource. Several optimization approaches are considered, estimates of the complexity and execution time of the forward pass of the neural network are given. Comparative estimates of the complexity of the network using different optimization approaches are given. The paper presents an analysis of the selected network architecture, and estimates of the computational complexity of individual components (modules) of the architecture are obtained. An analysis of possible optimization methods for each module was made. The parameters of the considered modules, the sizes of the input and output tensors are described. Several architectures were tested to optimize the feature extraction module, ResNet 50, ResNet 18, MobileNet v3 small, MobileNet v3 large. A comparative analysis of the computational complexity and execution time of the forward pass for each architecture is presented. Forward pass times were measured on Nvidia's Jetson AGX Xavier embedded computing device. Estimates of the execution time of the direct pass for each module of the considered neural networks are presented. The paper presents the results of comparing neural network accuracy estimates before and after architecture optimization. The test data set consists of 100 video recordings. 5 different typical objects are involved in test videos, 10 different scenarios are recorded for each object class. For each of the developed architectures, accuracy estimates were obtained, and a comparative analysis was made.*

*Optimization of the execution of the direct pass of the CNN; tracking.*

**Введение.** Решение задач с применением алгоритмов на основе методов глубокого машинного обучения необходимо при построении многих систем видео аналитики, во многих сферах жизнедеятельности человека. В настоящее время проходит большое количество международных соревнований, в которых разработ-

тики и исследователи соревнуются в точности и устойчивости разработанных ими алгоритмов. Самым популярным международным соревнованием является Visual Object Tracking Challenge (VOT) [1]. Одним из недостатков, применяемых в таких соревнованиях методов оценки алгоритмов является то, что в них не учитывается вычислительная сложность и время обработки кадра. В случае применения алгоритмов на основе методов глубокого машинного обучения во встраиваемых системах, из-за большой вычислительной сложности или неоптимальности некоторых математических операций, использование данных алгоритмов становится затруднительным или невозможным. Для устранения этих недостатков в статье представлены методы оптимизации алгоритма Dimp [2], который на протяжении нескольких лет был призером соревнований VOT [3, 4]

**Оптимизация архитектуры.** Реализация перспективных алгоритмов обработки данных во встраиваемых ЭВМ встречает ряд трудностей, обусловленных ограничениями, накладываемыми на состав аппаратных ресурсов, энергопотребление и массогабаритные характеристики. Для преодоления указанных трудностей, в настоящее время активно развиваются методы оптимизации нейросетевых алгоритмов обработки данных на различных аппаратных ресурсах.

Применяемые в настоящем способы оптимизации нейросетевой обработки данных условно могут быть разделены на 3 группы, представленные в табл. 1.

Таблица 1

**Пути оптимизации нейросетевых алгоритмов**

Математические	Определение базиса операций и поиск архитектуры (NAS)	Прореживание нейронных связей и нейронов (Pruning)	Декомпозиция слоёв (Low Rank Approximation)
Алгоритмические	Объединение нескольких слоёв СНС в один (Layer Fusion)	Быстрые алгоритмы вычисления множественных операций	Совместное использование весов. (Weights Sharing)
Программные	Квантование нейросети и переход к вычислениям в пониженной точности	Параллельные вычисления однотипных операций на доступных аппаратных ресурсах	Организация оптимального конвейера загрузки, обработки и выгрузки

В данной работе проработаны методы декомпозиции слоёв и архитектурный поиск NAS. На первом этапе оптимизации архитектура экстрактора признаков была заменяется с ResNet-18 на MobileNet v3. На втором этапе производится анализ наиболее вычислительно сложных слоёв остальных частей алгоритма и производится оптимизации этих слоёв путем замены вычислительно ёмких операций на их аппроксимированные аналоги. В перспективе планируется исследовать влияние на качество программных методов оптимизации.

**Анализ архитектуры сети.** Архитектура СНС представлена на рис. 1 [2].

Архитектура сети состоит из следующих блоков:

- ◆ Выделение признаков из исходного изображения или изображений (Feature extractor);
- ◆ Обучение прогностической модели объекта (Model Predictor);
- ◆ Модуль классификации (Conv 2d);
- ◆ Модуль определения размеров объекта (Optimize boxes).

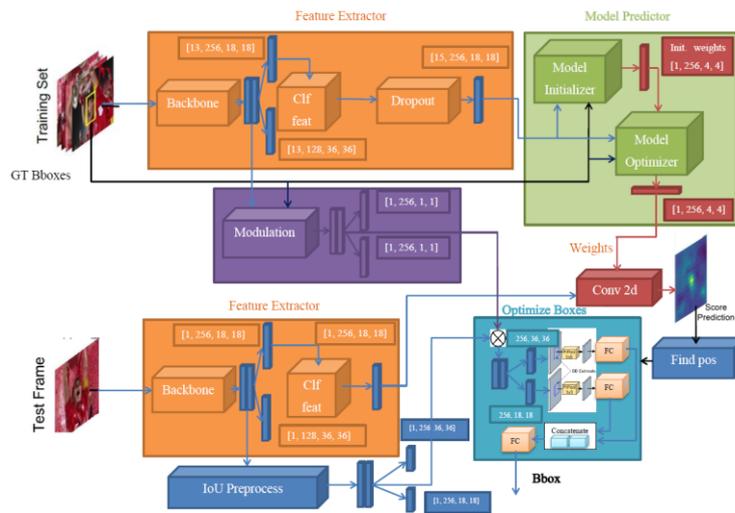


Рис. 1. Архитектура сети

Обучение или дообучение, модели выделения и сопоставления признаков объекта осуществляется в блоках: Feature extractor и Model Predictor. Реализация механизмов предсказания текущего местоположения объекта интереса осуществляется в блоках: Feature extractor, классификации и блоке определения размеров объекта.

**Оптимизация архитектуры блоков для уменьшения вычислительной сложности.** Для оптимизации блока Feature extractor были опробованы архитектуры ResNet 50, ResNet 18, MobileNet v3 large и small. Для остальных блоков применялся подход декомпозиции слоев. Блок Optimize boxes не подвергался оптимизации. Основная вычислительная сложность рассматриваемых блоков приходится на Residual блоки [4]. Из этого следует, что для оптимизации вычислительной сложности, в первую очередь необходимо оптимизировать трехмерную операцию сверки.

Для оптимизации сверточных слоев можно применить несколько подходов:

- ◆ Замена сверточного слоя на Depth Wise [5] свертку;
- ◆ Изменение количества каналов во входном тензоре;
- ◆ Изменение размеров ядра фильтра;
- ◆ Декомпозиция слоёв (применение BottleNeck блока).

В табл. 2 приведены оценки вычислительной сложности блока «Clf Feat» при различных размерах входных данных и количества каналов в функции свертки.

Таблица 2

**Вычислительная сложность блока «Clf Feat»**

Вх. кол. каналов	Вх. тензор	Батч	Ядро фильтра	Выходное кол. каналов	groups	Вычислительная сложность [GFlops]
256	18, 18	13	3x3	256	1	3.7297
256	18, 18	13	3x3	256	256	2.4924
128	18, 18	13	3x3	128	1	0.9332
128	18, 18	13	3x3	128	128	0.6251
64	18, 18	13	3x3	64	1	0.2337
64	18, 18	13	3x3	64	64	0.1573

Как видно из таблицы применение Depth Wise [5] свертки позволяет уменьшить вычислительную сложность. Изменение количества каналов приводит к большему уменьшению. Однако, стоит отметить, что уменьшение количества каналов может привести к существенному снижению оценок точности и устойчивости всей исследуемой модели.

Другим способом оптимизации является декомпозиция слоя (рис. 2,а) путем применения Bottle Neck блока (рис. 2,б). На рис. 2 представлены два блока, слева Residual Block справа Bottle Neck блок.

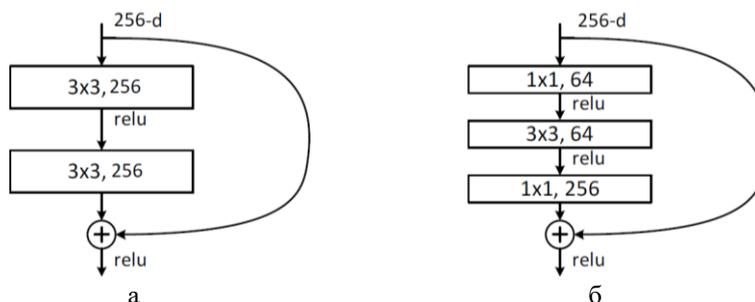


Рис. 2. Сравнение Residual блока и Bottle Neck блока

Архитектура рассматриваемого блока состоит из Residual блока [4] показали, что для оптимизации такого блока можно использовать Bottle Neck блок, архитектура которого показана на рис. 2. На основе этих данных предлагается использовать модифицированную архитектуру блока «Clf Feat». Применение данной архитектуры позволяет сократить вычислительную сложность до 0.2981 [GFlops], без изменения размеров входных и выходных данных. Вышеописанный подход применялся ко всем Residual блокам. В табл. 3 представлены данные вычислительной сложности рассматриваемых архитектур.

Таблица 3

**Вычислительная сложность алгоритмов различных модификаций алгоритма «Dimp»**

Архитектура / этап обработки	Вычислительная сложность [GFlops]				
	DIMP 50	PrDimp 18	PrDimp mobilenet v3 large	PrDimp mobilenet v3 large опт.	PrDimp mobileNet v3 small опт.
Инициализация	68.13	23.12	10.51	3.63	1.74
Обнаружение	10.92	2.33	1.36	0.44	0.4
Дообучение	1.73	0.15	0.15	0.075	0.075
Доопределение размеров объекта	0.11	0.11	0.11	0.11	0.11
Суммарная сложность	80.78	25.6	12.02	4.21	2.215

В табл. 4 представлены оценки времени выполнения рассматриваемых архитектур сети.

Таблица 4

**Время выполнения прямого прохода**

Архитектура / этап обработки	Время выполнения [мс]				
	DIMP 50	PrDimp 18	PrDimp mobilenet v3 large	PrDimp mobilenet v3 large опт.	PrDimp mobileNet v3 small опт.
Выделение признаков (Feature extractor)	27.24	22.96	15.36	12.18	10.71
Определение центра объекта	25.71	20.34	12.84	5.78	5.73
Доопределение размеров объекта	15.37	15.28	15.12	15.12	15.12
Суммарное время	68.32	58.58	43.82	33.08	31.56

В табл. 3 и 4 представлены архитектуры:

DIMP 50 – исходная архитектура с backbone ResNet 50;

PrDimp 18 – архитектура с backbone ResNet 18;

PrDimp mobilenet v3 large – архитектура с замененным backbone с ResNet на MobileNet v3;

PrDimp mobilenet v3 large и small опт. – архитектура с замененным backbone с ResNet на MobileNet v3 и проведенной декомпозицией слоев.

Как видно из табл. 4 время выполнения прямого прохода блока уточнения размеров объекта вносит самый значительный вклад в общее время выполнения прямого прохода. Это связано с тем, что в данном блоке итеративно (10 раз) уточняются размеры объекта путем их корректировки за счет обратного распространения ошибки. Для уменьшения времени выполнения данного блока в дальнейшем планируется заменить его на другой, не итеративный метод.

**Тестирование алгоритмов.** Для оценки точности использовался тестовый набор данных, сгенерированный в 3D симуляторе. Набор данных состоит из 100 видео записей. В тестовых видеозаписях участвует 5 различных типовых объектов, для каждого класса объекта записано 10 различных сценариев:

- ◆ ускорение;
- ◆ замедление;
- ◆ остановка за препятствием;
- ◆ остановка;
- ◆ смена ракурса;
- ◆ смена направления;
- ◆ перекрытие другими классами;
- ◆ перекрытие другими классами;
- ◆ перекрытие аналогичным классом;

На рис. 3 представлен график оценки точности-устойчивости тестируемых алгоритмов [8].

На рис. 3 DIMP 50 – исходная архитектура с backbone ResNet 50, PrDimp 18 – архитектура с backbone ResNet 18, PrDimpMb – архитектура с замененным backbone с ResNet на MobileNet v3, PrDimpMbCut. – архитектура с замененным backbone с ResNet на MobileNet v3 large и проведенной декомпозицией слоев, PrDimpMbCut2 – архитектура с замененным backbone с ResNet на MobileNet v3

small и проведенной декомпозицией слоев. Из рисунка можно сделать вывод, что примененные методы оптимизации привели к снижению оценок точности и устойчивости. Архитектура DIMP 50 обладает наилучшими оценками устойчивости, а архитектура PrDimp18 лучшими оценками точности. При этом оценки точности и устойчивости версий архитектур с backbone MobileNet v3 large с декомпозицией и без нее существенно не снизились относительно архитектуры DIMP 50. Из этого следует, что оптимальной архитектурой по критерию соотношения оценок точности устойчивости к времени выполнения является архитектура с backbone MobileNet v3 large и примененной декомпозицией.

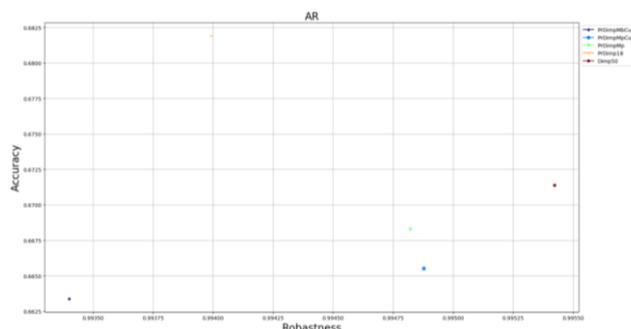


Рис. 3. Оценки точности-устойчивости

В табл. 5 представлены оценки тестируемых алгоритмов.

Таблица 5

Оценки тестируемых алгоритмов

Алгоритм / Метрика	$\sigma_{IoU}$	$M_{IoU}$	$\frac{\sigma_{Cerror}}{\overline{\sigma_{Cerror}}}$	$\frac{M_{Cerror}}{\overline{M_{Cerror}}}$	FP-Частота ложных обнаружений при видимости объекта меньше 50%	FN-Частота срывов
PrDimp mobilenet v3 large	0.159	0.704	4.687, 0.155	4.747, 0.179	0.641	0.005
PrDimp mobilenet v3 large опт.	0.164	0.707	4.696, 0.154	4.795, 0.180	0.737	0.005
PrDimp18	0.163	0.726	4.984, 0.160	4.702, 0.178	0.653	0.006
Dimp50	0.151	0.712	4.419, 0.150	4.525, 0.171	0.504	0.005
PrDimp mobilenet v3 small опт.	0.161	0.708	4.970, 0.171	4.927, 0.186	0.255	0.007

В табл. 5: Dimp50 – полная архитектура с backbone ResNet 50, PrDimp18 – полная архитектура с backbone ResNet 18, PrDimpMr – полная архитектура с backbone MobileNet v3 [5], PrDimpMrCut – оптимизированная архитектура с backbone MobileNet v3. В таблице представлены оценки СКО и МО коэффициента Жаккара,

СКО и МО ошибки определения центра, FP и FN [8]. Из полученных результатов можно сделать вывод, что примененные методы оптимизации не привели к существенному снижению оценок точности.

**Заключение.** Оптимизация архитектуры сети под встраиваемую платформу является важной частью в разработке системы с применением нейросетевых алгоритмов. Современные разработки в области архитектур нейронных сетей часто опираются только на метрики точности, что усложняет применение разработанных архитектур на встраиваемых платформах с ограниченным вычислительным ресурсом. В работе представлены методы позволяющие оптимизировать архитектуру сети, с целью ее запуска на системе на модуле в условиях профиля ограниченного энергопотребления. Полученные оценки точности позволяют говорить, что оптимизация существенно не повлияла на точность и устойчивость сети. Представленные методы оптимизации позволили сократить вычислительную сложность сети и время выполнения прямого прохода, однако, время выполнения блока уточнения размеров объекта не изменилось. Для дальнейшей оптимизации архитектуры предполагается заменить этот блок. В качестве альтернативы рассматриваются подход, описанный в статье [7]. Также предполагается применить методы оптимизации, основанные на квантовании нейросети и переход к вычислениям с ограниченной точностью. В результате оптимизации время прямого прохода сократилось с 68.32 до 31.56 [мс], вычислительная сложность снизилась с 80.78 до 2.21 [GFlops]. Исходя из всех полученных оценок, для дальнейшей доработки была выбрана архитектура с backbone MobileNet v3 large и примененной декомпозицией.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. <https://www.votchallenge.net/vot2021/>.
2. *Bhat G., Danelljan M., Luc Van Gool Radu Timofte.* Learning Discriminative Model Prediction for Tracking // IEEE/CVF International Conference on Computer Vision (ICCV). – 2019.
3. *Kristan M., Matas J., Leonardis A., et. all.* The Ninth Visual Object Tracking VOT2021 Challenge Results. VOT2021 challenge workshop, ICCV workshops, 2021.
4. *Kristan M., Matas J., Leonardis A., et. all.* The Eighth Visual Object Tracking VOT2020 Challenge Results. ECCV2020 workshops, 2020.
5. *Howard et al A.* Searching for MobileNetV3 // 2019 IEEE/CVF International Conference on Computer Vision (ICCV). – 2019. – P. 1314-1324. – DOI: 10.1109/ICCV.2019.00140.
6. *Jiang B., Luo R., Mao J., Xiao T.* Acquisition of Localization Confidence for Accurate Object Detection. ECCV 2018.
7. *Bhat G., Danelljan M., Khan F.* IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). – 2019.
8. *Щелкунов А.Е., Ковалев В.В., Морев К.И., Сидько И.В.* Метрики оценки алгоритмов автоматического сопровождения // Известия ЮФУ. Технические науки. – 2020. – № 1. – С. 233-245.
9. *Kasturi R., Goldgof D., Soundararajan P., Manohar V., Garofolo J., Bowers R., Boonstra M., Korzhova V., Zhang J.* Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol // TPAMI. – 2009. – Vol. 31, No. 2. – С. 319-326.
10. *Vencel E.S., Ovcharov L.A.* Probability theory and its engineering applications. Handbook for universities. – 2-nd. – М.: High school, 2000. – 480 p.
11. *Nawaz T., Cavallaro A.* A protocol for evaluating video trackers under real-world conditions // Image Processing IEEE. – 2013. – Vol. 22, No. 4. – P. 1354-1361
12. *Yang M., Wu Y., Hua G.* Context-aware visual tracking // IEEE transactions on pattern analysis and machine intelligence. – 2006. – No. 31. – P. 1195-1209.
13. *Stauffer C., Grimson W.* Learning patterns of activity using real-time tracking // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2000. – P. 747-757.
14. *Sundaresan A., Chellappa R.* Multi-camera tracking of articulated human motion using shape and motion cues // IEEE Transactions on Image Processing. – 2009. – P. 2114-2126.

15. *Henriques F., Caseiro R., Martins P., Batista J.* High-Speed Tracking with Kernelized Correlation Filters // *TPAMI*. – 2014. – Vol. 42, No. 5. – P. 345-362.
16. *Frey J. Dueck D.* Clustering by Passing Messages Between Data Points // *Science Today*. – 2007. – Vol. 315. – P. 972-976.
17. *Smith K., Gatica-Perez D., Odobez J.* Evaluating Multi-Object Tracking // *CVPR Work.* – IEEE, 2005. – Vol. 3. – P. 32-36.
18. *Black J., Ellis T., Rosin P.* A novel method for video tracking performance evaluation // *VS-PETS*. – 2003. – P. 125-132.
19. *Kao E., Daggett M., Hurley M.* An information theoretic approach for tracker performance evaluation // *CVPR*. – 2009. – P. 1523-1529.
20. *Bashir F. Porikli F.* Performance Evaluation of Object Detection and Tracking Systems // *PETS*. – 2006. – P. 190-203.
21. *Checka N., Wilson K., Rangarajan V., Darrell T.* A probabilistic framework for multi-modal multi-person tracking // *Proceedings of the IEEE Workshop on Multi-Object Tracking (WOMOT '03)*. – 2003. – P. 203-212.

## REFERENCES

1. Available at: <https://www.votchallenge.net/vot2021/>.
2. *Bhat G., Danelljan M., Luc Van Gool Radu Timofte.* Learning Discriminative Model Prediction for Tracking, *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
3. *Kristan M., Matas J., Leonardis A., et. all.* The Ninth Visual Object Tracking VOT2021 Challenge Results. VOT2021 challenge workshop, ICCV workshops, 2021.
4. *Kristan M., Matas J., Leonardis A., et. all.* The Eighth Visual Object Tracking VOT2020 Challenge Results. ECCV2020 workshops, 2020.
5. *Howard et al A.* Searching for MobileNetV3, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314-1324. DOI: 10.1109/ICCV.2019.00140.
6. *Jiang B., Luo R., Mao J., Xiao T.* Acquisition of Localization Confidence for Accurate Object Detection. *ECCV 2018*.
7. *Bhat G., Danelljan M., Khan F.* IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
8. *Shchelkunov A.E., Kovalev V.V., Morev K.I., Sid'ko I.V.* Metriki otsenki algoritmov avtomaticheskogo soprovozhdeniya [Metrics for evaluating automatic tracking algorithms], *Izvestiya YuFU. Tekhnicheskie nauki [Izvestiya SFedU. Engineering Sciences,]* 2020, No. 1, pp. 233-245.
9. *Kasturi R., Goldgof D., Soundararajan P., Manohar V., Garofolo J., Bowers R., Boonstra M., Korzhova V., Zhang J.* Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol, *TPAMI*, 2009, Vol. 31, No. 2, pp. 319-326.
10. *Vencel E.S., Ovcharov L.A.* Probability theory and its engineering applications. Handbook for universities. 2-nd. Moscow: High school, 2000, 480 p.
11. *Nawaz T., Cavallaro A.* A protocol for evaluating video trackers under real-world conditions, *Image Processing IEEE*, 2013, Vol. 22, No. 4, pp. 1354-1361
12. *Yang M., Wu Y., Hua G.* Context-aware visual tracking, *IEEE transactions on pattern analysis and machine intelligence*, 2006, No. 31, pp. 1195-1209.
13. *Stauffer C., Grimson W.* Learning patterns of activity using real-time tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, pp. 747-757.
14. *Sundaresan A., Chellappa R.* Multi-camera tracking of articulated human motion using shape and motion cues, *IEEE Transactions on Image Processing*, 2009, pp. 2114-2126.
15. *Henriques F., Caseiro R., Martins P., Batista J.* High-Speed Tracking with Kernelized Correlation Filters, *TPAMI*, 2014, Vol. 42, No. 5, pp. 345-362.
16. *Frey J. Dueck D.* Clustering by Passing Messages Between Data Points, *Science Today*, 2007, Vol. 315, pp. 972-976.
17. *Smith K., Gatica-Perez D., Odobez J.* Evaluating Multi-Object Tracking, *CVPR Work.* IEEE, 2005, Vol. 3, pp. 32-36.
18. *Black J., Ellis T., Rosin P.* A novel method for video tracking performance evaluation, *VS-PETS*, 2003, pp. 125-132.
19. *Kao E., Daggett M., Hurley M.* An information theoretic approach for tracker performance evaluation, *CVPR*, 2009, pp. 1523-1529.

20. *Bashir F. Porikli F.* Performance Evaluation of Object Detection and Tracking Systems, *PETS*, 2006, pp. 190-203.
21. *Checka N., Wilson K., Rangarajan V., Darrell T.* A probabilistic framework for multi-modal multi-person tracking, *Proceedings of the IEEE Workshop on Multi-Object Tracking (WOMOT '03)*, 2003, pp. 203-212.

Статью рекомендовал к опубликованию д.т.н., профессор Я.Е. Ром.

**Щелкунов Александр Евгеньевич** – Акционерное общество Научно–конструкторское бюро вычислительных систем (АО НКБ ВС); e-mail: alexandrshcelkunov@mail.ru; г. Таганрог, Россия; программист.

**Ковалев Владислав Владимирович** – e-mail: vlad.kovalev94@mail.ru; программист.

**Сидько Иван Владимирович** – e-mail: ivan.brokenroll@gmail.com; к.т.н.; зав. отделом.

**Сергеев Николай Евгеньевич** – e-mail: nesergeev@sfedu.ru; д.т.н.; профессор.

**Shchelkunov Alexander Evgen'evich** – Joint Stock Company «Scientific Design Bureau of Computing Systems» (JSC SDB CS); e-mail: alexandrshcelkunov@mail.ru; Taganrog, Russia; programmer.

**Kovalev Vladislav Vladimirovich** – e-mail: vlad.kovalev94@mail.ru; programmer.

**Sidko Ivan Vladimirovich** – e-mail: ivan.brokenroll@gmail.com; cand. of eng. sc.; department head.

**Sergeev Nikolay Evgenievich** – e-mail: nesergeev@sfedu.ru; dr. of eng. sc., professor.