

УДК 004.627

## ИЕРАРХИЧЕСКИЙ АДАПТИВНЫЙ АЛГОРИТМ ШАБЛОННОГО ПОИСКА ДЛЯ ОЦЕНКИ ДВИЖЕНИЯ ПРИ АНАЛИЗЕ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ

Нгуен Ван Чьонг<sup>а</sup>, А.А. Тропченко<sup>а</sup>

<sup>а</sup> Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

Адрес для переписки: thientruong.mars@gmail.com

### Информация о статье

Поступила в редакцию 01.03.16, принята к печати 21.03.16

doi: 10.17586/2226-1494-2016-16-3-474-481

Язык статьи – русский

**Ссылка для цитирования:** Нгуен Ван Чьонг, Тропченко А.А. Иерархический адаптивный алгоритм шаблонного поиска для оценки движения при анализе видеопоследовательности // Научно-технический вестник информационных технологий, механики и оптики. 2016. Т. 16. № 3. С. 474–481. doi: 10.17586/2226-1494-2016-16-3-474-481

### Аннотация

**Предмет исследования.** Рассматриваются алгоритмы оценки движения для анализа видеопоследовательностей в стандартах сжатия MPEG-4 Visual и H.264. На основе анализа достоинств и недостатков существующих алгоритмов был предложен собственный алгоритм. **Метод.** Алгоритм, названный нами иерархическим адаптивным алгоритмом шаблонного поиска (иерархический ARPS, Hierarchical ARPS), включает в себя классический адаптивный алгоритм шаблонного поиска ARPS и иерархический поиск MP (Hierarchical search or Mean pyramid). Алгоритмы оценки движения реализованы с помощью пакета MATLAB и протестированы на нескольких видеопоследовательностях. **Основные результаты.** В качестве критериев оценки алгоритмов использованы скорость работы, соотношение сигнал/шум, среднеквадратичная ошибка и среднее абсолютное отклонение. Предложенный метод показал гораздо большее быстродействие при сравнимой ошибке и отклонении. Соотношение сигнал/шум оказалось на разных видеопоследовательностях как больше, так и ниже показателей известных алгоритмов, что требует дальнейших исследований. **Практическая значимость.** При использовании взамен стандартного алгоритма в кодеках MPEG-4 и H.264 предложенный алгоритм позволяет значительно снизить время сжатия и может быть рекомендован для применения в телекоммуникационных системах для задач хранения, передачи и обработки мультимедиа данных.

### Ключевые слова

компенсация движения, алгоритмы сопоставления блоков, иерархический адаптивный алгоритм шаблонного поиска, сопоставление блоков, оценка подобию

## HIERARCHICAL ADAPTIVE ROOD PATTERN SEARCH FOR MOTION ESTIMATION AT VIDEO SEQUENCE ANALYSIS

Nguyen Van Truong<sup>а</sup>, A.A. Tropchenko<sup>а</sup>

<sup>а</sup> ITMO University, Saint Petersburg, 197101, Russian Federation

Corresponding author: thientruong.mars@gmail.com

### Article info

Received 01.03.16, accepted 21.03.16

doi: 10.17586/2226-1494-2016-16-3-474-481

Article in Russian

**For citation:** Nguyen Van Truong, Tropchenko A.A. Hierarchical adaptive rood pattern search for motion estimation at video sequence analysis. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2016, vol. 16, no. 3, pp. 474–481. doi: 10.17586/2226-1494-2016-16-3-474-481

### Abstract

**Subject of Research.** The paper deals with the motion estimation algorithms for the analysis of video sequences in compression standards MPEG-4 Visual and H.264. A new algorithm has been offered based on the analysis of the advantages and disadvantages of existing algorithms. **Method.** The algorithm is called hierarchical adaptive rood pattern search (Hierarchical ARPS, HARPS). This new algorithm includes the classic adaptive rood pattern search ARPS and hierarchical search MP (Hierarchical search or Mean pyramid). All motion estimation algorithms have been implemented using MATLAB package and tested with several video sequences. **Main Results.** The criteria for evaluating the algorithms were: speed, peak signal to noise ratio, mean square error and mean absolute deviation. The proposed method showed a much better performance at a comparable error and deviation. The peak signal to noise ratio in different video sequences shows better and worse results than characteristics of known algorithms so it requires further investigation. **Practical Relevance.** Application of this algorithm in MPEG-4 and H.264 codecs instead of the standard can significantly reduce compression time. This feature enables to recommend it in telecommunication systems for multimedia data storing, transmission and processing.

**Keywords**

motion compensation, block matching algorithms, hierarchical adaptive rood pattern search, block matching, similarity evaluation

**Введение**

Кодирование с предсказанием используется для устранения большой временной и пространственной избыточности, которая существует в видеопоследовательностях. В обычных кодированиях с предсказанием разница между текущим кадром и кадром-прогнозом (на основе предыдущего кадра) кодируется и передается. Чем лучше кадр-прогноз, тем меньше ошибка. Если сцена остается, то хороший прогноз текущего пикселя есть пиксель в предыдущем кадре, а ошибка равна нулю. Тем не менее, при движении в последовательности пиксель на той же части движущегося объекта служит хорошим прогнозом для текущего пикселя [1].

Существует большое количество алгоритмов оценки движения для кодирования с предсказанием. Данная работа ориентируется на алгоритмы, которые называются алгоритмами сопоставления блоков. Алгоритм сопоставления блоков является способом нахождения соответствия блоков в видеопоследовательности для оценки движения. Алгоритм включает в себя разделение текущего кадра на блоки и сравнение каждого из них с соответствующим блоком в соседнем кадре видео. Создается вектор, который описывает движение блока из одного места в другое. Этот процесс выполняется для всех блоков кадра.

Оценка движения требует довольно большого объема вычислений и может потреблять до 80% от вычислительной мощности кодера, если используется полный перебор (FS), который оценивает все возможные блоки-кандидаты в пределах окна поиска. Из-за этого недостатка начали искать другие эффективные алгоритмы. Многие варианты таких алгоритмов [2–10] были разработаны для оценки движения на основе блока. В настоящей работе рассмотрим такие алгоритмы быстрой оценки движения, как полный перебор (FS), трехшаговый поиск (TSS), четырехшаговый поиск (FSS), ромбический поиск (DS), перекрестный ромбический поиск (CDS) и адаптивный метод шаблонного поиска (ARPS). На основе этих алгоритмов предлагается иерархический адаптивный алгоритм шаблонного поиска для оценки и компенсации движения.

**Алгоритмы сопоставления блоков**

Метод сопоставления блоков является наиболее популярным и практичным методом оценки движения в кодировании видео. На рис. 1 показано, как он работает. Текущий кадр делится на блоки размером  $b \times b$  (обычно  $b = 8$  или  $16$ ). Затем производится сравнение каждого блока текущего кадра с соответствующими блоками соседнего кадра (предыдущего или последующего) для нахождения блока соответствия. В результате сравнения создается вектор, который описывает движение блока из одного места в другое [11–14].

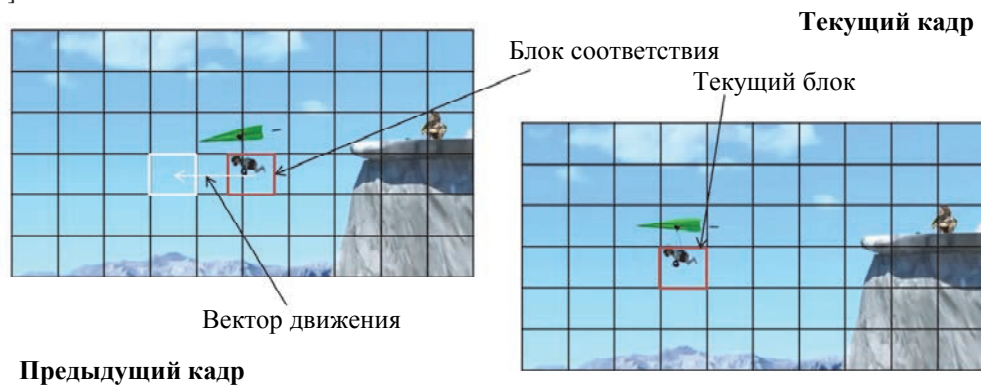


Рис. 1. Оценка движения на основе сопоставления блоков

Для ограничения масштаба поиска определяется область поиска. Предположим, что блок с размером  $b \times b$  имеет вектор движения с максимальным допустимым смещением  $\pm s$  пикселей ( $s$  – параметр поиска) как в горизонтальном, так и в вертикальном направлениях, тогда существуют  $(2s+1)^2$  возможных блоков-кандидатов внутри области поиска (рис. 2).

Соответствие между текущим блоком и одним из блоков-кандидатов определяется на основе критериев оценки подобия блоков. Самые популярные и менее сложные по вычислениям критерии представлены ниже:

– среднее абсолютное отклонение (MAD, Mean Absolute Difference),

$$MAD = \frac{1}{b} \sum_{i=1}^b \sum_{j=1}^b |I(i, j) - K(i, j)|;$$

– среднеквадратическая ошибка (MSE, Mean Square Error),

$$MSE = \frac{1}{b^2} \sum_{i=1}^b \sum_{j=1}^b [I(i, j) - K(i, j)]^2,$$

где  $I(i, j)$  и  $K(i, j)$  – яркости исходного и восстановленного кадра в точке  $(i, j)$ , а  $b$  – размер блока;

– пиковое соотношение сигнал/шум (PSNR, peak signal to noise ration),

$$PSNR = 10 \log_{10} \left[ \frac{MAX_I^2}{MSE} \right],$$

где  $MAX_I$  – это максимальное значение, принимаемое пикселем изображения.

Критерии MAD и MSE берутся по минимуму, а PSNR – по максимуму.

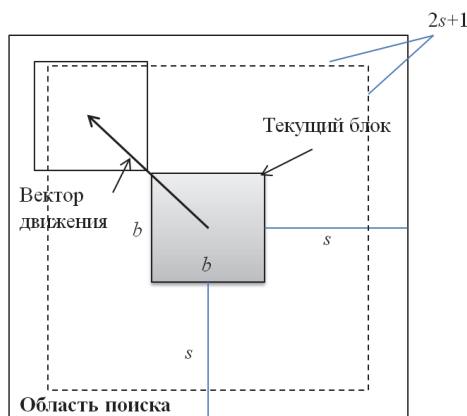


Рис. 2. Процесс поиска блока соответствия

**Полный перебор (FS).** Берется каждый блок текущего кадра и сравнивается с соответствующим блоком соседнего, в итоге получаем лучший блок. Хотя этот алгоритм является лучшим выбором с точки зрения качества предсказанного кадра и простоты, он требует очень большого объема вычислений (как уже было сказано, до 80% вычислительных затрат видеокодека).

**Трехшаговый поиск (TSS).** Был разработан в 1981 году и до сих пор является популярным благодаря своей простоте, надежности и высокой производительности. Основная проблема алгоритма заключается в удаленности равномерно распределенных точек шаблона, что делает его неэффективным для областей малого движения.

**Четырехшаговый поиск (FSS).** Алгоритм основан на таком свойстве большинства видеопоследовательностей, как ориентированность к центру кадра. Как правило, данный алгоритм показывает большую надежность с сохранением эффективности для сложных вариантов движения и операций масштабирования. Это делает FSS привлекательной стратегией для выборки блоков в схемах компенсации движения.

**Ромбический поиск (DS).** Алгоритм работает практически как FSS. Тем не менее, алгоритм может иметь неограниченное число шагов. DS использует 2 фиксированных шаблона (рис. 3, а): малый шаблон ромбического поиска (SDSP) и большой шаблон ромбического поиска (LDSP).

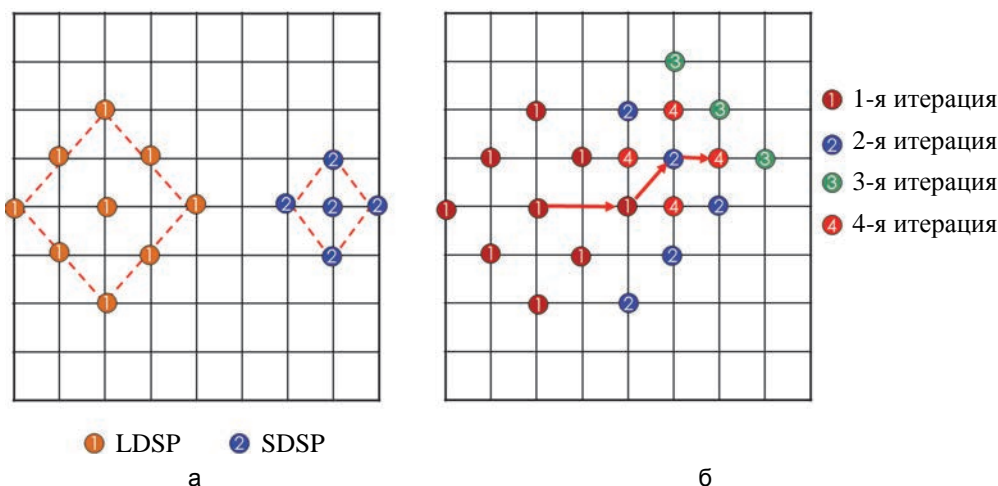


Рис. 3. LDSP и SDSP (а); пример работы алгоритма DS (б)

Алгоритм работает следующим образом.

- Стадия 1 (поиск по LDSP). Начинается с шагом шаблона 2. Найти 8 точек вокруг центра области поиска, которые имеют координаты  $(x,y)$  и  $|x|+|y|=2$ . Найти точку с наименьшим отклонением. Если найденная точка окажется центром, то перейти к стадии 2; в противном случае переместить центр в эту точку и повторить поиск по LDSP.
- Стадия 2 (поиск по SDSP). Установить шаг шаблона на 1. Найти точку с минимальным отклонением из установленных точек по SDSP. Построить вектор движения.

На рис. 3, б, представлен пример процесса построения траектории движения блока из 4 итераций алгоритма DS. Здесь и далее стадия – это этап выполнения алгоритма, а итерация – число исполнений алгоритма.

Этот алгоритм применим в большей степени для последовательности изображений с широким диапазоном движения. Он превосходит алгоритм TSS с точки зрения объема вычислений при одновременном выполнении примерно на 20–25%.

**Перекрестный ромбический поиск (CDS).** Алгоритм использует перекрестный шаблон поиска (CSP) для начальной стадии, а LDSP и SDSP для последующих. Шаблон CSP имеет 9 точек с центром области поиска, который представлен на рис. 4, а.

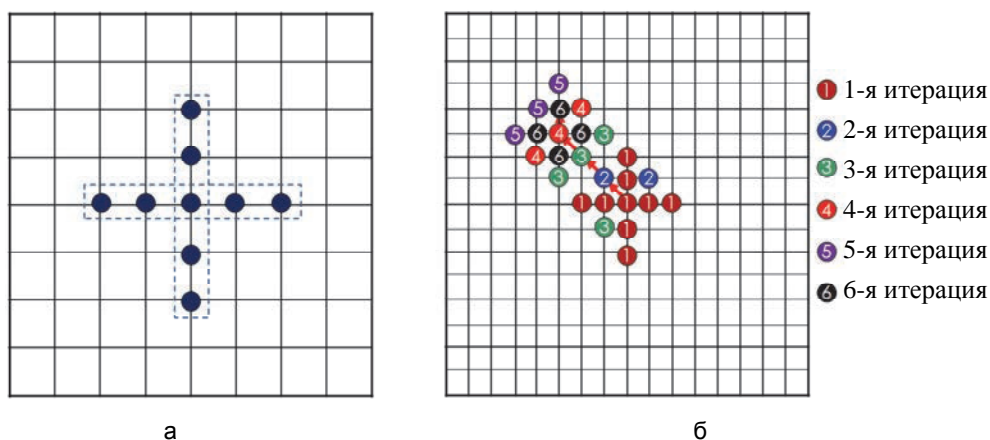


Рис. 4. CSP (а); пример процедуры алгоритма CDS (б)

Алгоритм можно описать следующим образом.

- Стадия 1. Найти точку с минимальным отклонением из 9 точек на шаблоне CSP. Если она кажется центром, то процесс закончится; в противном случае перейти к стадии 2.
- Стадия 2. Добавить 2 точки с координатами типа  $(\pm 1;\pm 1)$  близко к найденной точке в стадии 1 (поиск по LDSP). Если необходимая точка окажется центром стороны CSP, т.е. точка с координатами типа  $(\pm 1;0)$  или  $(0;\pm 1)$ , то процесс останавливается. В противном случае перейти к стадии 3.
- Стадия 3. Продолжить поиск по LDSP. Если необходимая точка находится в центре шаблона LDSP, то перейти к стадии 4. В противном случае повторить эту стадию.
- Стадия 4. Проводить поиск по SDSP. Построить вектор движения с помощью найденной точки.

На рис. 4, б, представлен пример процесса построения траектории движения блока с помощью алгоритма CDS из 6 итераций.

**Адаптивный метод поиска шаблона (ARPS).** Алгоритм использует тот факт, что общее движение в кадре, как правило, поступательное, т.е. если блоки вокруг текущего блока двигаются в определенном направлении, то существует высокая вероятность того, что текущий блок также будет иметь подобный вектор движения. Этот алгоритм использует вектор движения блока, находящегося непосредственно слева, для предсказания собственного вектора движения.

Данный алгоритм реализуется следующим образом (рис. 5).

- Стадия 1. Найти предсказанный вектор движения блока. Установить шаг шаблона на значение  $\max(|x|,|y|)$ , где  $(x,y)$  – координаты предсказанного вектора движения. Найти точки вокруг центра, находящиеся на расстоянии шага шаблона от центра. Найти точку с минимальным отклонением, в которую затем переместить центр.
- Стадия 2. Производить поиск по SDSP вокруг точки центра. Повторить поиск по SDSP до того, как точка с минимальным отклонением будет находиться в центре.

С точки зрения объема вычислений ARPS превосходит DS, так как в этом случае предсказанный вектор движения имеет координаты  $(0;0)$ , ARPS сразу переходит к поиску по SDSP (не нужно проводить поиск по LDSP). Тем не менее, если вектор движения находится далеко от центра, то этот алгоритм быстрее переходит к поиску по SDSP, чем алгоритм DS.

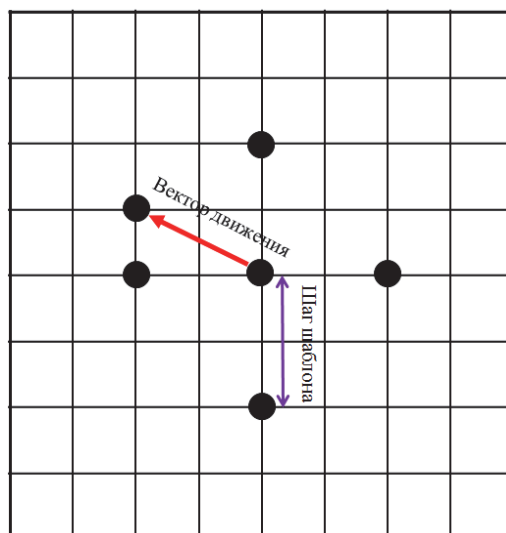


Рис. 5. Пример процедуры алгоритма ARPS

**Иерархический адаптивный алгоритм шаблонного поиска (иерархический ARPS или HARPS)**

Для устранения недостатков существующих алгоритмов авторами предлагается алгоритм, названный «иерархический ARPS», который включает в себя алгоритм ARPS и иерархический поиск MP (Mean pyramid).

Вначале для устранения шумового эффекта получают изображение меньшего разрешения с помощью фильтра низких частот. В дальнейшем для получения многоуровневой иерархии изображений меньшего разрешения используется простое усреднение пикселей изображения предыдущего уровня по принципу

$$g_L(p, q) = \left[ \frac{1}{4} \left( \sum_{u=0}^1 \sum_{w=0}^1 g_{L-1}(2p+u, 2q+w) \right) \right],$$

где  $g_L(p, q)$  – значение пикселя уровня  $L$  в позиции  $(p, q)$ . Таким образом, если использовать три уровня иерархии, один пиксель 2-го уровня соответствует блоку  $4 \times 4$  0-го уровня и блоку  $2 \times 2$  – 1-го уровня соответственно. В то же время блок размером  $16 \times 16$  0-го уровня будет соответствовать блоку  $(16/2L) \times (16/2L)$  уровня  $L$  [15]. Общая схема работы алгоритма MP представлена на рис. 6.

После проведения MP применяем алгоритм ARPS на новых созданных кадрах.

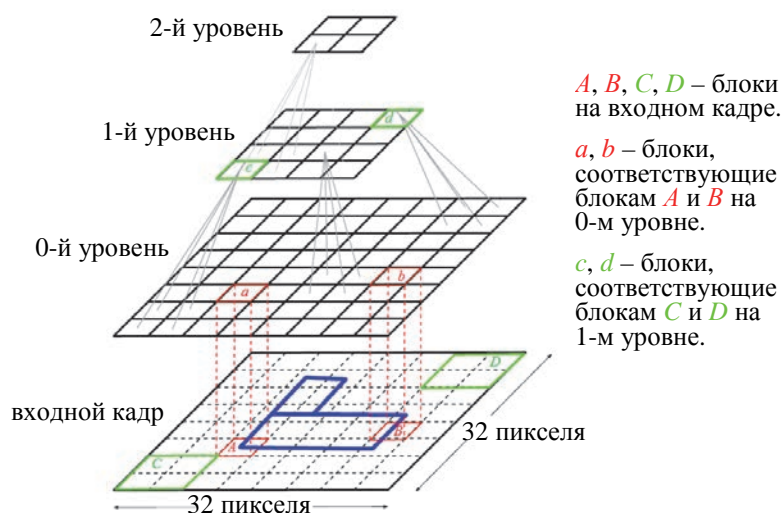


Рис. 6. Схема работы алгоритма MP

**Экспериментальные результаты**

Рассмотренные выше алгоритмы были реализованы с помощью MATLAB и протестированы с несколькими видеопоследовательностями, в том числе видео Tiger (34 кадра), видео Ronaldinho (34 кадра), видео Walkonwater (34 кадра), видео Gun (34 кадра), видео Football (34 кадра) и видео Tennis (36 кадров).



Первая видеопоследовательность имеет пространственное разрешение 320×256 пикселей и временную частоту дискретизации 25 кадров/с. Вторая последовательность имеет пространственное разрешение 160×128 пикселей и временную частоту дискретизации 25 кадров/с. Третья последовательность имеет пространственное разрешение 320×256 пикселей и временную частоту дискретизации 30 кадров/с. Последние три последовательности имеют пространственное разрешение 256×160 пикселей и временную частоту дискретизации 29 кадров/с.

Для всех алгоритмов размер блока для оценки движения составил 16×16, параметр поиска  $s = 7$ . Сравнение средних количеств точек поиска по блоку представлено на рис. 7.

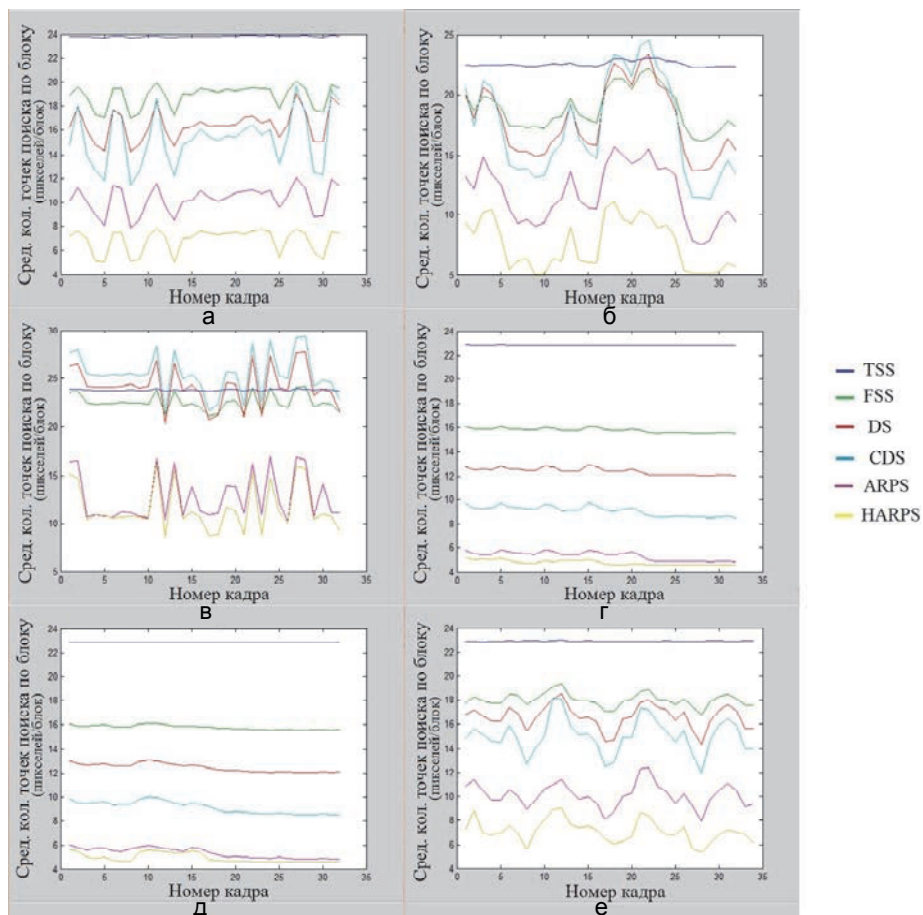


Рис. 7. Среднее количество точек поиска по блоку кадров видео: Tiger (а); Ronaldinho (б); Walkonwater (в); Gun (г); Football (д); Tennis (е)

Предлагаемый алгоритм HARPS превосходит остальные за счет уменьшения размера области поиска с помощью алгоритма MP. Следовательно, скорость выполнения этого алгоритма выше, чем у других. Покадровый коэффициент PSNR видео показан на рис. 8 и в таблице для различных тестовых видео. Согласно результатам, можно увидеть, что предлагаемый алгоритм HARPS дает в некоторых тестовых видео улучшение коэффициента PSNR, а в некоторых – ухудшение по сравнению с другими алгоритмами, что требует дополнительных исследований.

Видео	Алгоритмы						
	ES	TSS	FSS	DS	CDS	ARPS	HARPS
Tiger	35,4468	35,0898	35,1518	35,3339	35,2920	35,2723	33,9322
Ronaldinho	28,5570	27,8704	28,1583	28,1932	28,1806	27,9857	27,5616
Walkonwater	23,7942	23,5356	23,4151	23,4602	23,4325	23,5679	23,4080
Gun	35,0982	34,8786	34,8860	34,8744	34,8070	34,7522	34,5140
Football	33,0506	32,9904	32,9493	32,9391	32,9238	32,9197	33,1809
Tennis	27,8711	27,3982	27,3774	27,4704	27,3929	27,3691	27,7339

Таблица. Среднее значение покадрового коэффициента PSNR, дБ

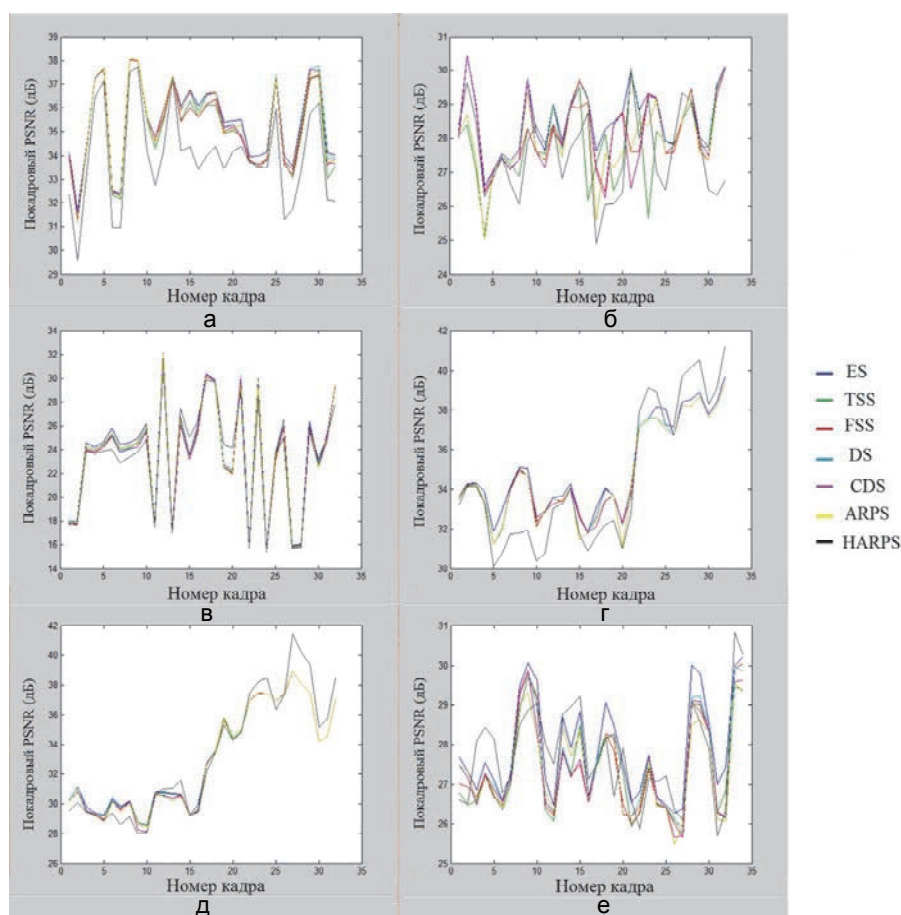


Рис. 8. Покадровый коэффициент PSNR видео: Tiger (а); Ronaldinho (б); Walkonwater (в); Gun (г); Football (д); Tennis (е)

### Заключение

В работе был предложен новый алгоритм оценки и компенсации движения HARPS и его применение для видеосжатия. Алгоритм HARPS, созданный на основе алгоритмов ARPS и MP, был протестирован с несколькими видеопоследовательностями. Экспериментальные результаты показали, что предлагаемый алгоритм работает значительно быстрее, при этом коэффициент PSNR сопоставим с другими методами.

### Литература

1. Turaga D., Alkanhal M. Search Algorithms for Block Matching Estimation. Mid-term Project, 1998.
2. Toivonen T., Heikkilä J., Silvén O. A new algorithm for fast full search block motion estimation based on number theoretic transforms // Proc. 9<sup>th</sup> International Workshop on Systems: Signals and Image Processing, Manchester, United Kingdom, 2002. P. 90–94.
3. Chen M.-J., Chiueh T.-D. One-dimensional full search motion estimation algorithm for video coding // IEEE Transactions on Circuits and Systems for Video Technology. 1994. V. 4. N 5. P. 504–509. doi: 10.1109/76.322998
4. Li R., Zheng B., Liou M.L. A new three-step search algorithm for block motion estimation // IEEE Transactions on Circuits and Systems for Video Technology. 1994. V. 4. N 4. P. 438–442. doi: 10.1109/76.313138
5. Chau L.-P., Jing X. Efficient three-step search algorithm for block motion estimation in video coding // Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Hong Kong, 2003. V. 3. P. 421–424.
6. Po L.-M., Ma W.-C. A novel four-step search algorithm for fast block motion estimation // IEEE Transactions on Circuits and Systems for Video Technology. 1996. V. 6. N 3. P. 313–317. doi: 10.1109/76.499840
7. Zhu S., Ma K.-K. A new diamond search algorithm for fast block-matching motion estimation // IEEE Transactions on Image Processing. 2000. V. 9. N 2. P. 287–290. doi: 10.1109/83.821744
8. Cheung C.-H., Po L.-M. A novel cross-diamond search algorithm for fast block motion estimation // IEEE Transactions on Circuits and Systems for Video Technology. 2002. V. 12. N 12. P. 1168–1177. doi: 10.1109/TCSVT.2002.806815

9. Nie Y., Ma K.-K. Adaptive rood pattern search for fast block-matching motion estimation // IEEE Transactions on Image Processing. 2002. V. 11. N 12. P. 1442–1449. doi: 10.1109/TIP.2002.806251
10. Nan K.M., Kin J.S., Pari R.H., Shin Y.S. A fast hierarchical motion vector estimation algorithm using mean pyramid // IEEE Transactions on Circuits and Systems for Video Technology. 1995. V. 5. N 4. P. 344–351. doi: 10.1109/76.465087
11. Moschetti F., Kunt M., Debes E. A statistical adaptive block-matching motion estimation // IEEE Transactions on Circuits and Systems for Video Technology. 2003. V. 13. N 4. P. 417–431. doi: 10.1109/TCSVT.2003.811363
12. Babu D.V., Subramanian P., Karthikeyan C. Performance analysis of block matching algorithms for highly scalable video compression // Proc. Int. Symposium on Ad Hoc and Ubiquitous Computing. Surathkal, India, 2006. P. 179–182. doi: 10.1109/ISAHUC.2006.4290669
13. Barjatya A. Block Matching Algorithms For Motion Estimation // DIP 6620 Final Project Paper in Digital Image Processing. Utah State University. P. 1–6.
14. Cuevas E., Zaldivar D., Pérez-Cisneros M., Olive D. Block-matching algorithm based on differential evolution for motion estimation // Engineering Applications of Artificial Intelligence. 2013. V. 26. N 1. P. 488–498. doi: 10.1016/j.engappai.2012.08.003
15. Нгуен В.Ч., Тропченко А.А. Методы и алгоритмы сокращения временной избыточности видеоданных // Сборник статей II Международной мультидисциплинарной научно-практической конференции «Актуальные проблемы науки XXI века». Москва, 2015. Часть 2. С. 36–41.

<i>Нгуен Ван Чьонг</i>	– аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, thientruong.mars@gmail.com
<i>Тропченко Андрей Александрович</i>	– кандидат технических наук, доцент, доцент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, zayka_98rus@mail.ru
<i>Nguyen Van Truong</i>	– postgraduate, ITMO University, Saint Petersburg, 197101, Russian Federation, thientruong.mars@gmail.com
<i>Andrey A. Tropchenko</i>	– PhD, Associate professor, Associate professor, ITMO University, Saint Petersburg, 197101, Russian Federation, zayka_98rus@mail.ru