

УДК 004.932.2

АДАПТИВНЫЙ МЕТОД ОЦЕНКИ ДВИЖЕНИЯ В ВИДЕО

© 2008 г. К. А. Симонян, С. В. Гришин, Д. С. Ватолин

{simonyan, sgrishin, dmitriy}@graphics.cs.msu.ru

*Кафедра Автоматизации систем вычислительных комплексов,
Лаборатория Компьютерной графики и мультимедиа*

1 Введение

В настоящее время оценка движения (ОД) широко применяется в сжатии и обработке видео. У обеих областей применения есть своя специфика.

Поскольку соседние кадры видеопоследовательности, как правило, очень похожи, можно существенно повысить эффективность алгоритмов сжатия видео за счет кодирования лишь разницы между соседними кадрами (межкадровой разницы), а не каждого кадра по отдельности. В настоящее время с целью повышения эффективности кодирования видеокодеки сжимают не межкадровую разницу, а скомпенсированную. Скомпенсированной разницей называют разницу между текущим и скомпенсированным кадрами. Скомпенсированный кадр аппроксимирует текущий кадр и построен из фрагментов предыдущего кадра с использованием информации о движении между этими кадрами. Таким образом, ОД имеет решающее значение для повышения эффективности алгоритмов сжатия видео. При этом основное требование, предъявляемое к алгоритму ОД, — вычислить параметры движения, минимизирующие скомпенсированную разницу. Таким образом, параметры движения могут не соответствовать истинному движению объектов в видеопоследовательности.

Другой сферой применения ОД является ее использование в алгоритмах обработки видео. ОД играет важную роль в таких задачах, как преобразование частоты кадров (ПЧК), сегментация видео (выделение и сопровождение объектов в кадре), преобразование чересстрочной развертки в прогрессивную, шумоподавление, восстановление сжатого видео, повышение качества видео, оцифрованное со старых киноплёнок, а также во многих других. Например, использование информации о движении при шумоподавлении позволяет производить обработку не только в пространственной области, но и во временной. В ПЧК информация о движении используется для определения положения объектов на вычисляемых в процессе обработки кадрах. Очевидно, что при этом, в отличие от алгоритмов сжатия видео, критическое значение имеет правильность определения параметров движения.

2 Основные методы оценки движения. Методы сопоставления блоков

Можно выделить несколько основных групп методов ОД: методы оптического потока (optical flow) [4], методы фазовой корреляции (phase correlation) [5], методы сопоставления блоков (block matching algorithms) [6].

Наиболее широко на практике применяются методы сопоставления блоков из-за простоты аппаратной реализации и высокой вычислительной эффективности. Общая схема работы этих методов такова:

1. текущий кадр разбивается на множество непересекающихся блоков;
2. для каждого блока текущего кадра производится поиск наиболее похожего блока (реперного блока) в предыдущем кадре.

Разность между позициями текущего и реперного блоков называют вектором движения текущего блока (Рис. 1).

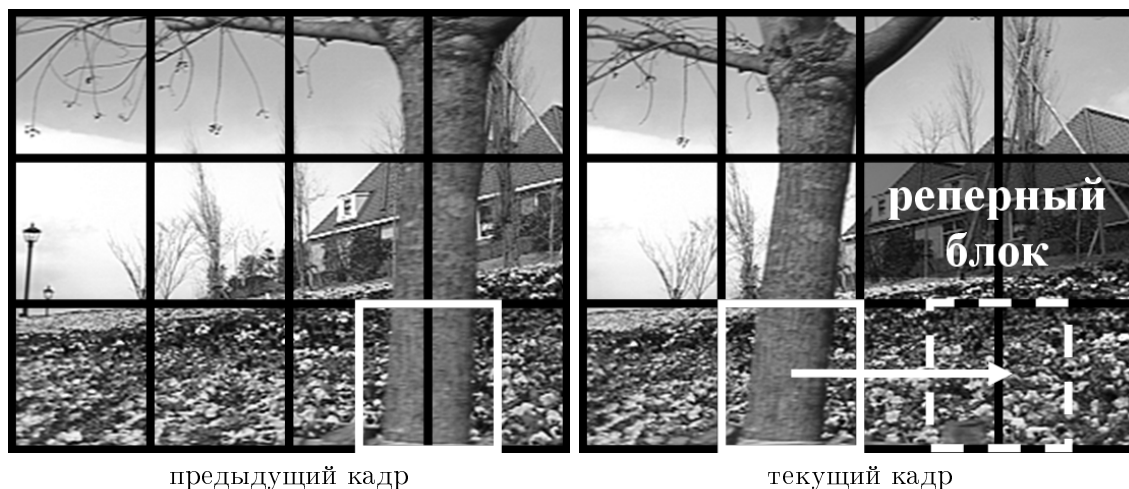


Рис. 1: Текущий и реперный блоки и соответствующий вектор движения

В современных алгоритмах сопоставления блоков для ускорения поиска вектора движения блока используется сходство между векторами движения в пространстве и во времени. Ускорение достигается за счет перебора лишь небольшого числа векторов движения вместо проверки всех возможных. Поиск вектора движения для каждого блока в таких алгоритмах производится следующим образом:

1. формируется множество векторов-кандидатов;
2. в нем осуществляется поиск лучшего вектора-кандидата; (*)
3. производится его уточнение.

Классические ВМА-методы имеют определенные недостатки, ограничивающие их применение в алгоритмах обработки видео:

- неспособность вычислить истинное движение блоков в однородных областях;
- использование блоков постоянного размера приводит либо к недостаточной точности поля векторов на границах объектов (в случае применения блоков большого размера), либо к высокой чувствительности к шуму и уменьшению скорости обработки (в случае использования блоков малого размера).

3 Разработанный алгоритм

В настоящей работе описывается метод оценки истинного движения, использующий адаптивно меняющийся размер блока и адаптивную стратегию поиска вектора движения, зависящую от контрастности блока. Общая схема алгоритма совпадает со схемой (*), каждый из этапов которой подробно описан в трех следующих подразделах. В отдельном подразделе описан метод определения размера блока.

3.1 Формирование множества векторов-кандидатов

Множество векторов-кандидатов состоит из следующих векторов (Рис. 2):

- векторы движения соседних блоков текущего кадра;
- векторы движения блоков предыдущего кадра;
- нулевой вектор, если блок достаточно контрастен.

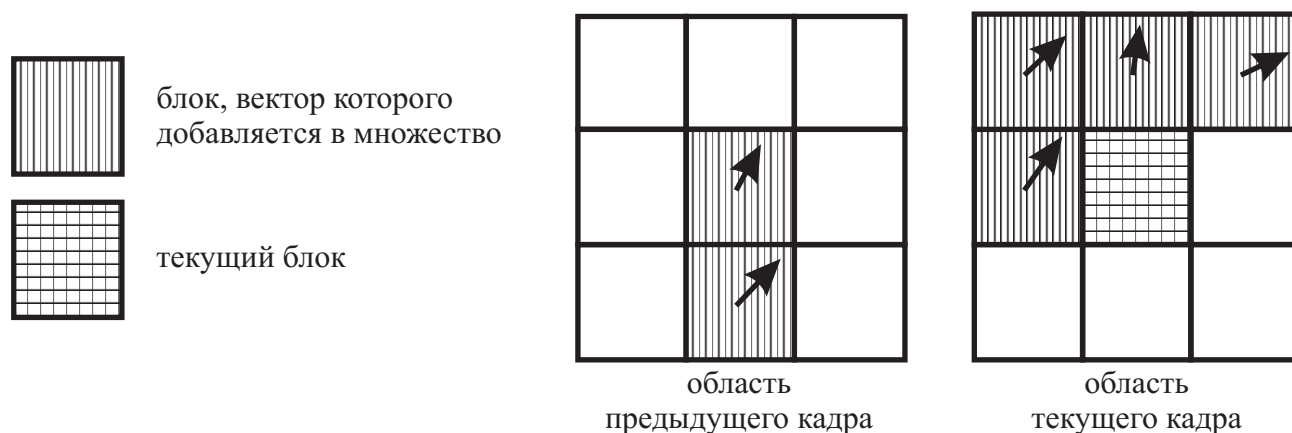


Рис. 2: Множество векторов-кандидатов

3.2 Поиск лучшего вектора-кандидата

Поиск лучшего вектора движения во множестве векторов-кандидатов производится следующим образом:

1. для каждого вектора из множества кандидатов удаляются векторы, достаточно близкие к нему по норме L_1 . При этом подсчитывается число вхождений этих векторов в множество кандидатов (частота появления кандидата);
2. для каждого вектора из множества кандидатов вычисляется значение функции ошибки. В качестве функции ошибки обычно используется SAD (Sum of Absolute Differences, сумма абсолютных разностей):

$$SAD(t, B, \vec{d}^B) = \sum_{(x,y) \in B} |I(t, x, y) - I(t-1, x + d_x^B, y + d_y^B)|,$$

где \vec{d}^B — вектор движения блока B ,

$I(t, x, y)$ — яркость точки (x, y) кадра с номером t ;

3. ошибка каждого вектора-кандидата умножается на его вес $w \in (0; 1]$. Вес зависит от частоты появления соответствующего кандидата и контрастности блока:
 - если блок достаточно контрастный, то вес всех векторов-кандидатов полагается равным единице;
 - в противном случае (блок однородный) вес каждого вектора-кандидата обратно пропорционален его частоте появления;
4. в множестве кандидатов производится поиск лучшего вектора-кандидата — вектора с наименьшей взвешенной ошибкой.

Использование весов позволяет повысить качество работы алгоритма по сравнению с обычным способом выбора вектора из множества кандидатов, в котором учитываются только ошибки векторов. В данном случае повышение качества состоит в том, что векторное поле на выходе алгоритма больше соответствует истинному движению между кадрами, что подтверждается результатами сравнения (см. разд. 4).

3.3 Уточнение лучшего вектора-кандидата

Уточнение лучшего вектора-кандидата производится с помощью следующего итеративного процесса. Вычисляются значения функции ошибки для векторов, достаточно близких по норме L_1 к лучшему вектору, полученному на предыдущем шаге. Из них выбирается вектор, дающий наименьшее значение ошибки, после чего, в случае необходимости, операция повторяется для него. Стратегия уточнения вектора выбирается адаптивно и зависит от контрастности блока. При этом для менее контрастных блоков максимальное количество итераций уточнения меньше, чем для более контрастных блоков.

Благодаря адаптивной схеме поиска вектора движения достигается повышение качества поля векторов движения в однородных областях (Рис. 7).

3.4 Разбиение блока

Разбиение блока на четыре подблока производится адаптивно:

- в областях со сложным движением;
- в областях, где текущий размер блока не может обеспечить приемлемый уровень ошибок.

Для поиска областей со сложным движением вычисляется дисперсия координат векторов, входящих в множество векторов-кандидатов для блока. Области со сложным движением определяются путем сравнения дисперсии с порогом: если она превышает порог, то в окрестности блока имеет место сложное движение.

Приемлемый уровень ошибок для вектора движения блока определяется следующим образом. Согласно [2], имеет место зависимость между величиной

$$VAR(t, B) = \frac{1}{4} \sum_{i,j \in \{-2;2\}} \sum_{(x,y) \in B} |I(t, x, y) - I(t, x + j, y + i)|$$

и средним значением ошибки вектора. Таким образом, вычислив величину VAR для блока, становится возможным адаптивно определить порог для ошибки вектора, в случае превышения которого блок необходимо разбить на блоки меньшего размера.

Адаптивный выбор размера блока позволяет значительно повысить точность векторного поля (например, на границах объектов), не повышая при этом чувствительность к шуму [2]. Иными словами, блоки малого размера используются только в тех областях кадра, где это необходимо.

4 Сравнение

Было проведено сравнение разработанного метода с алгоритмами FAME [1] и E3DRS [3] как с одними из наиболее популярных алгоритмов ОД того же класса. Для сравнения использовался стандартный набор видеопоследовательностей. Были рассмотрены два сценария применения ОД:

1. использование блоков размера 16×16 .
Данный сценарий подразумевает разбиение области кадра только на блоки размера 16×16 ;

2. использование блоков размера 4×4 .
В данном случае сравнение с алгоритмом FAME не проводилось, поскольку он предназначен только для работы с блоками размера 16×16 . Алгоритм E3DRS работает с блоками произвольного *фиксированного* размера, поэтому для него использовалось разбиение области кадра на блоки размера 4×4 . Поскольку разработанный алгоритм способен *адаптивно* определять размер блока (см. разд. 3.4), для него использовалось начальное разбиение области кадра на блоки размера 16×16 , а во время работы алгоритма происходило адаптивное разбиение блоков на подблоки по схеме $16 \times 16 \rightarrow 8 \times 8 \rightarrow 4 \times 4$.

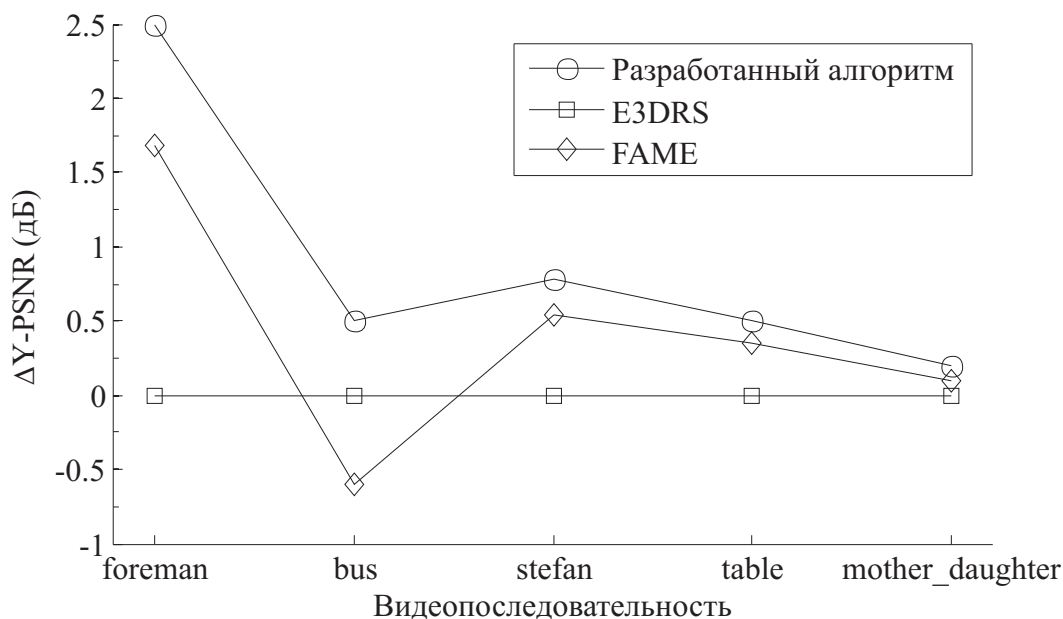


Рис. 3: Сравнение ΔY -PSNR для различных алгоритмов (размер блоков 16×16)

В сравнении применялась метрика Y-PSNR, вычисляемая для каждой пары скомпенсированный/исходный кадр по их яркостным компонентам. Большие значения метрики соответствуют меньшей скомпенсированной разнице. Иными словами, чем выше показатель Y-PSNR, тем выше точность найденных векторов движения. Для удобства сравнения была также вычислена метрика ΔY -PSNR, которая представляет собой разность значений метрики Y-PSNR сравниваемого алгоритма и алгоритма E3DRS. Также было проведено сравнение скорости обработки кадров указанными методами.

При использовании первого сценария разработанный алгоритм демонстрирует лучшие значения Y-PSNR — соответствующая ветвь на графике (Рис. 3) лежит выше остальных. Несмотря на некоторое отставание разработанного алгоритма по скорости обработки (Рис. 4), он, тем не менее, осуществляет обработку кадров видеопотока с высокой скоростью.

Как видно из Рис. 5, при использовании второго сценария разработанный алгоритм опережает алгоритм E3DRS по метрике Y-PSNR на всех видеопоследовательностях, кроме mother_daughter. Однако, субъективное сравнение (Рис. 7) демонстрирует преимущество описанного метода на этой видеопоследовательности. На приведенном кадре легко видеть неверно найденные алгоритмом E3DRS векторы движения в однородных областях (они минимизируют скомпенсированную разницу, но истинному движению не соответствуют). Поле векторов движения, полученное с помощью разработанного алгоритма, в этих областях найдено верно. Как видно, размер блоков уменьшается на границах объектов, что подтверждает эффективность работы этапа разбиения блоков. По скорости обработки (Рис. 6) разработанный метод опережает E3DRS на видео, где присутствуют достаточно большие однородные области, поскольку в этих областях используются блоки максимального размера. На видео, где таких областей нет, описанный метод уступает по скорости E3DRS. Однако скорость остается приемлемой для использования алгоритма в целях обработки видео в реальном времени.

5 Результаты

Разработан адаптивный алгоритм оценки истинного движения, не имеющий некоторых недостатков, присущих алгоритмам этого типа, что подтверждается результатами объективного и субъективного сравнений. Тем не менее, остается большое пространство для улучшения предложенного алгоритма. Дальнейшее повышение точности планируется достичь за счет использования многокадровой ОД и фильтрации поля векторов движения во временной

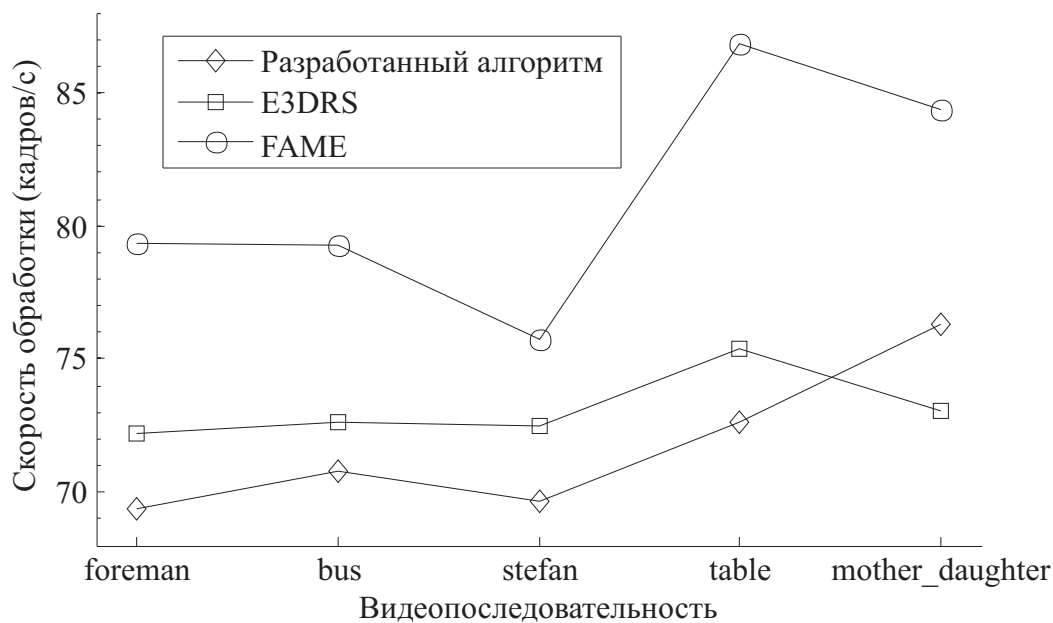


Рис. 4: Сравнение скорости обработки для различных алгоритмов (размер блоков 16×16)

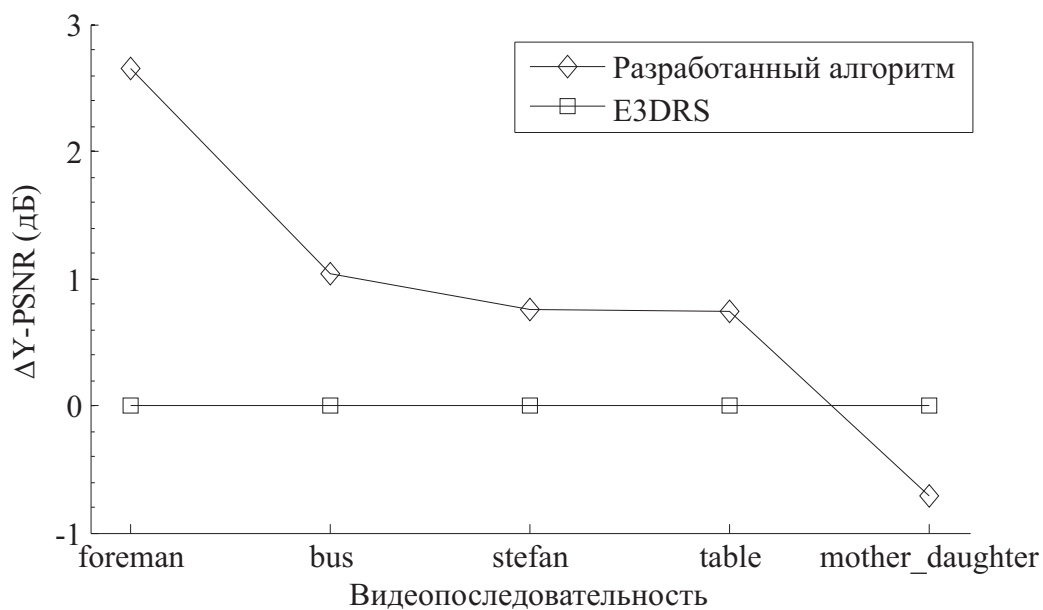


Рис. 5: Сравнение ΔY -PSNR для различных алгоритмов (размер блоков 4×4)

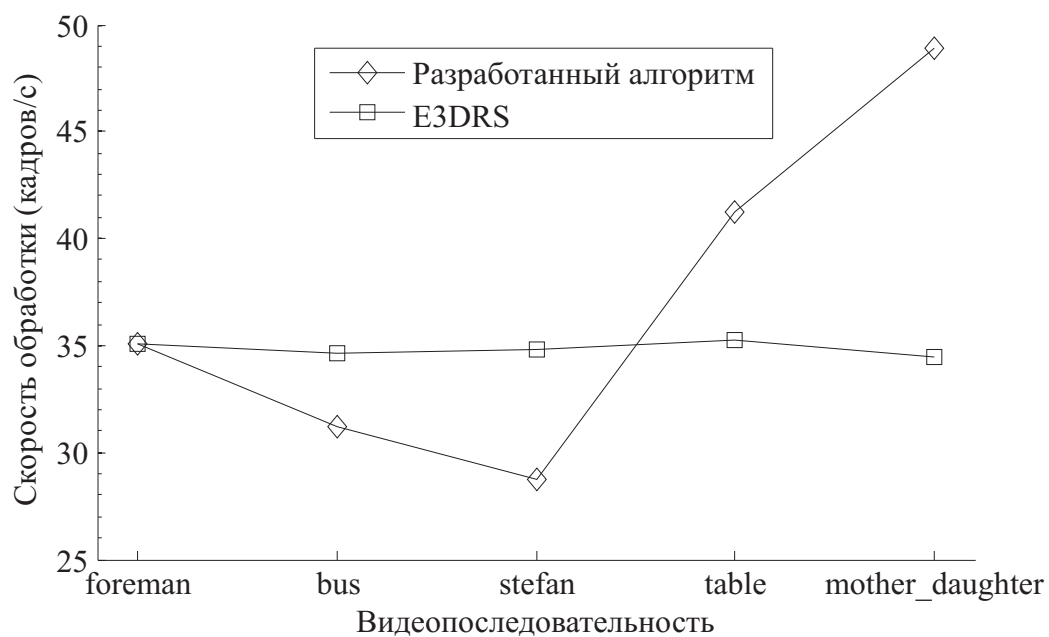


Рис. 6: Сравнение скорости обработки для различных алгоритмов (размер блоков 4×4)



Рис. 7: Визуальное сравнение разработанного алгоритма с E3DRS (кадр видеопоследовательности mother_daughter)

области. В многокадровой ОД повышение качества достигается за счет возможности поиска реперных блоков не только в предыдущем кадре, но и в нескольких других. В ряде случаев это является единственным способом получить правильные векторы движения. В частности, многокадровая ОД позволяет корректно вычислить векторы движения в областях, где движущиеся объекты перекрывают друг друга. Анализ изменения поля векторов движения во времени позволяет выявить ошибочные векторы движения, которые затем могут быть скорректированы (фильтрация поля векторов движения).

Разработанный алгоритм реализован в виде фильтра для программы обработки видео VirtualDub. Фильтр свободно доступен по адресу http://www.compression.ru/video/motion_estimation/index.html.

Список литературы

- [1] I. Ahmad, W. Zheng, J. Luo, and M. Liou. *A fast adaptive motion estimation algorithm*. IEEE Transactions on CSVT, March 2006, pp. 420–438.
- [2] R. Braspenning and G. de Haan. *Efficient Motion Estimation with Content-Adaptive Resolution*. In Proceedings of ISCE'02, Sep. 2002, pp. E29–E34.
- [3] S. Olivieri, L. Albani, and G. de Haan. *A low-complexity motion estimation algorithm for H.263 video coding*. In Proceedings of Philips Conference on DSP, Veldhoven, Nov. 1999, paper 17.3.
- [4] E. Memin and P. Perez. *Dense estimation and object-based segmentation of the optical flow with robust techniques*. IEEE Transactions on Image Processing, 7(5):703–719, 1998.
- [5] M. Li, M. Biswas, S. Kumar, and Nguyen Truong. *DCT-based phase correlation motion estimation*. In Proceedings of ICIP 2004, Vol. 1, pp. 445–448.
- [6] С. Путилин. *Быстрый алгоритм нахождения движения в видеопоследовательностях*. Труды конференции Graphicon-2006, с. 407–410, Новосибирск, Академгородок, 1–4 Июля 2006.