

**Гришин С.В., Ватолин Д.С., Лукин А.С.,  
Путилин С.Ю., Стрельников К.Н.**  
**Обзор блочных методов оценки движения в  
цифровых видео сигналах**

**1. Введение**

Цифровое видео сегодня широко распространено, в основном благодаря спутниковому, кабельному и домашнему телевидению, а на территории США аналоговое телевидение планируется прекратить уже в 2009 году. Кроме того, на цифровое видео быстро переходят многие бытовые устройства, такие как видеопроекторы (замена VHS кассет на DVD и MPEG-4 диски), видеокамеры (съемка в цифровые форматы DV и MPEG-2); и даже телевизоры с аналоговой электронно-лучевой трубкой быстро сменяются LCD телевизорами, цифровыми проекционными системами и плазменными панелями. Во всех этих устройствах используются алгоритмы обработки и сжатия видео.

Задача оценки движения является ключевой при работе с цифровым видео. Причиной этому является исключительная важность информации о движении при анализе временной составляющей видеосигнала, которая, в свою очередь, является важнейшей характеристикой видеосигналов и во многом определяет специфику задач цифровой обработки видео. Поясним это на примере задачи сжатия видео.

В первых системах сжатия цифрового видео кадры обрабатывались независимо друг от друга. Каждый кадр кодировался как изображение, а не как часть видео потока. Затем появились алгоритмы, использующие вычитание соседних кадров. В них кодировались не сами кадры, а разница каждого кадра с предыдущим. Этот прием обусловил значительный рост эффективности алгоритмов сжатия благодаря тому факту, что соседние кадры видео потока, как правило, очень похожи, и их разница часто близка к нулю. Исключением из этого правила были случаи наличия движения между соседними кадрами. Следующим шагом стало появление алгоритмов, использующих компенсацию движения. Компенсацией движения называется преобразование одного из пары кадров, использующее информацию о движении между этими кадрами, и осуществляемое таким образом, чтобы все объекты в кадре имели позиции на момент времени второго кадра пары. Другими словами, компенсация движения делает один из кадров пары максимально похожим на другой, используя информацию о движении между ними. Таким образом, компенсация движения позволяет использовать при сжатии избыточность видео

потока во времени даже при наличии движения между кадрами, чего не могли делать алгоритмы сжатия видео предыдущего поколения.

Следует особо выделить две области применения алгоритмов оценки движения (ОД): сжатие и обработка видео. В этих областях к алгоритму ОД предъявляются различные требования. В сжатии видео критическое значение имеет размер информации о движении и скомпенсированной межкадровой разнице (разница между текущим кадром и скомпенсированным). При этом не важно, соответствует ли направление найденных векторов движения реальному движению объектов в видео потоке: главное требование к векторам движения – минимизация скомпенсированной межкадровой разницы. В алгоритмах обработки видео, напротив, объем информации о движении не играет никакой роли, основное значение здесь имеет точность найденной информации о движении, соответствие найденных векторов реальному движению в видео последовательности. Это обусловило появление группы алгоритмов ОД, предназначенных для поиска «истинного» движения (true motion estimation). Специфика указанных требований к алгоритмам ОД очень важна и найдет свое отражение в обзоре алгоритмов, представленной в основной части данной статьи.

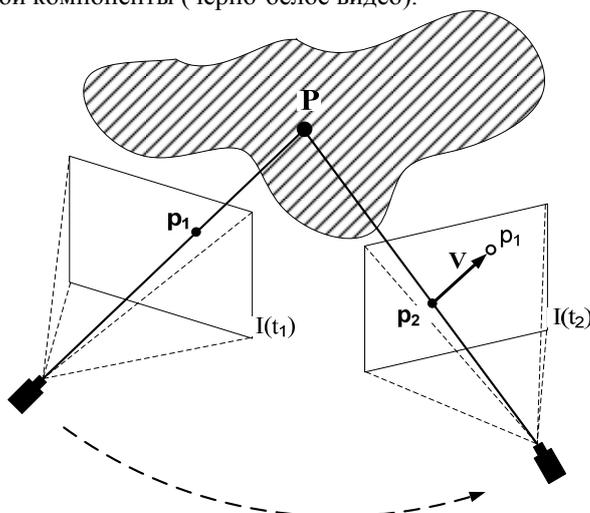
В виду особой важности задачи ОД, к настоящему моменту было разработано великое множество соответствующих алгоритмов, которые можно разделить на следующие группы: блочные методы, методы оптического потока, фазовой корреляции, глобальной оценки движения, слежения за особенностями, многокадровой ОД, а также алгоритмы, комбинирующие приемы методов указанных категорий. Наиболее многочисленной из перечисленных является группа блочных методов. Это обусловлено универсальностью, невысокой вычислительной сложностью и сравнительно высокой эффективностью алгоритмов этой категории. Не последнюю роль сыграла также простота их аппаратной реализации. Именно по этим причинам в данной работе представлен обзор блочных алгоритмов ОД.

## 2. Понятие информации о движении

Цифровое видео представляет собой упорядоченный набор кадров. Именно поэтому применительно к нему часто используют термин видео последовательность. В данной работе также будем придерживаться этой терминологии. Видео последовательность будем обозначать, как  $I(t)$ , где  $t$  – порядковый номер кадра. Каждый кадр – это матрица пикселей, размер этой матрицы обозначим  $W \times H$ .

Будем использовать следующие обозначения:  
 $p = (x, y)$ ,  $(x, y) \in [0, W - 1] \times [0, H - 1]$  – вектор координат пикселя в кадре,  $I(p, t)$  – яркость пикселя с координатами  $p = (x, y)$

в кадре  $I(t)$ . Вообще говоря, яркость – это только один из цветовых параметров пикселя. Существует множество цветовых моделей (RGB, YUV, Lab и т.д.), в каждой из которых цвет определяется несколькими компонентами. Однако человеческий глаз наиболее чувствителен к яркостной компоненте. По этой причине, а также для простоты изложения, в данной работе будем предполагать наличие только яркостной компоненты (черно-белое видео).



**Рис. 1. Схема вычисления вектора движения**

Информацией о движении в обработке видео называют двумерный массив *векторов движения*, размер которого равен размеру кадра  $W \times H$ . При этом под вектором движения в заданной точке понимается вектор изменения координат этой точки между двумя заданными кадрами. Поясним смысл этого понятия с помощью Рис. 1. Рассмотрим трехмерную сцену без движения и два кадра  $I_1 = I(t_1)$ ,  $I_2 = I(t_2)$ , полученных при помощи камеры из разных точек. Точке  $P$  трехмерной сцены в кадре  $I_1$  соответствует пиксель с координатами  $p_1$ , будем его называть образом точки  $P$  на кадре  $I_1$ . Тогда образом точки  $P$  на кадре  $I_2$  будет пиксель с координатами  $p_2$ . Сразу заметим, что под образом здесь понимается не само изображение точки  $P$  на кадре, а вектор координат ее проекции на матрицу камеры. Это означает, что образ точки  $P$  определен даже тогда, когда она ее изображение на кадре отсутствует, например, вследствие наличия

какого-либо объекта на переднем плане. Вектор движения  $V$  в точке  $p_2$  для пары кадров  $I_1$  и  $I_2$  определяется так:

$$V_{I_1}^{I_2}(p_2) = (u, v) = p_1 - p_2.$$

В данном примере рассмотрен случай неподвижной сцены и движущейся камеры. В общем случае, изменение координат точек трехмерной сцены на кадрах может быть обусловлено как движением камеры, так и движением объектов сцены.

### 3. Общая схема алгоритмов блочной оценки движения

Каждый кадр видео последовательности разбивается на множество неперекрывающихся блоков  $B_{i,j}$  заданного размера, где  $i, j$  – координаты блока. Разбиение производится так, что все блоки покрывают весь кадр, т.е. их суммарная площадь равна площади кадра.

Задача ОД сводится к задаче поиска вектора движения  $v_{i,j}$  для каждого блока  $B_{i,j}$ . При этом вектора  $v_{i,j}$  определяются соотношением (1):

$$v_{i,j} = \arg\left(\min_{v_{i,j} \in O} (F(t, i, j, v_{i,j}))\right), \quad (1)$$

$$O = \{(x, y) | x \in [-u_{\max}, u_{\max}], y \in [-v_{\max}, v_{\max}]\}, \quad (2)$$

$$SAD(t, i, j, v) = \sum_{p \in B_{i,j}} |I(p, t) - I(p + v, t-1)|, \quad (3)$$

где

$O$  – область поиска векторов движения,  $u_{\max}, v_{\max}$  – целые положительные числа;

$F(t, i, j, v_{i,j})$  – функция соответствия блоков, это мера близости блоков текущего и предыдущего кадров. Примером такой функции является SAD (Sum of Absolute Differences), определяемая формулой (3);

Суть работы алгоритмов данной группы заключается в следующем. Для каждого блока текущего кадра производится минимизация функции соответствия блоков по 4-му аргументу, при этом область минимизации может быть любой, единственным ограничением является то, что она должна быть подмножеством области поиска  $O$ . В качестве вектора движения для каждого блока выбирается аргумент минимума функции соответствия, вычисленный в этом блоке. Фактически при вычислении функции соответствия производится определение «похожести» двух блоков: блока текущего

кадра и блока предыдущего кадра, смещенного на вектор  $v_{i,j}$ . Таким образом, процесс минимизации функции соответствия является поиском блока предыдущего кадра, наиболее «похожего» на текущий блок.

Важно заметить, что размер области поиска определяет максимальный модуль векторов движений. На практике нередки случаи, когда алгоритм ОД не в состоянии найти верные вектора движения только потому, что амплитуда движения в видео слишком велика.

Для удобства дальнейшего изложения кадр, для блоков которого производится поиск соответствий, будем называть текущим, а кадр, в котором производится поиск – опорным.

#### **4. Базовые методы**

В этом разделе будут рассмотрены базовые подходы блочной ОД, такие как полный перебор, шаблонные методы, метод иерархического поиска и методы, использующие вектора-кандидаты. Описание начнем с наиболее простого и очевидного алгоритма полного перебора.

##### **4.1. Алгоритм полного перебора**

Поскольку область поиска  $O$  конечная, то наиболее очевидным методом минимизации функции соотношения блоков является полный перебор всех значений аргумента  $v \in O$ .

Данный подход имеет свои достоинства и недостатки. Достоинством данного метода является гарантированное нахождение глобального минимума функции соответствия для каждого блока. Однако, как было замечено во введении, не во всех приложениях важно найти именно глобальный минимум. В обработке видео критическое значение имеет определение «истинных» векторов движения, независимо от величины соответствующих им значений функции соответствия.

Очевидным недостатком является вычислительная сложность данного метода. Даже в свете высокой мощности современных процессоров, полный перебор может быть неприемлем для обработки в режиме реального времени в случае высокого разрешения видео и большой области поиска.

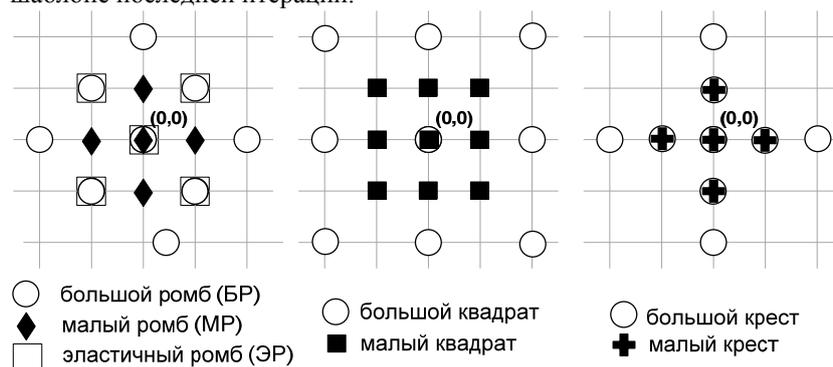
Логическим продолжением алгоритма полного перебора являются методы шаблонного поиска.

##### **4.2. Методы шаблонного поиска**

Данные методы представляют собой дискретные варианты покоординатного спуска. Они основываются на предположении, что функция соответствия достаточно гладкая для каждого блока, без локальных минимумов.

Перед началом описания общей схемы алгоритмов данного класса следует ввести понятие шаблона. Шаблоном является набор координат точек, причем координаты точек отсчитываются от центра шаблона. Таким образом, для произвольно заданной точки по шаблону можно определить набор координат нескольких точек, число этих точек зависит от используемого шаблона.

Поиск вектора в каждом блоке является итеративным процессом. На каждой итерации вычисляется координата центра шаблона, координаты всех точек шаблона, и, затем, значения функции соответствия в каждой из точек шаблона. Центр шаблона на первой итерации называют центром поиска, он обычно равен  $(0, 0)$ . В качестве центра шаблона для следующей итерации выбирается та точка шаблона, в которой был достигнут минимум функции соответствия. Затем проверяется условия останова поиска, и в зависимости от результата производится переход к следующей итерации или завершение поиска вектора в данном блоке. При этом в качестве результата выбирается вектор, соответствующий точке минимума функции соответствия на шаблоне последней итерации.



**Рис. 2. Примеры шаблонов поиска**

Таким образом, шаблонный алгоритм ОД определяется используемым шаблоном. Наиболее часто используемыми на практике шаблонами являются большой и малый ромбы (БР и МР), большой и малый квадраты, а так же большой и малый кресты (см. Рис. 2).

Основным недостатком методов данного класса является их склонность к нахождению локальных минимумов функции соответствия вместо глобальных. В каждом блоке, как правило, выбирается вектор, соответствующий одной из ближайших к центру поиска точек локального минимума функции соответствия, а вовсе не «истинному» движению или ее глобальному минимуму. Однако, у данного класса методов есть существенное достоинство: они

значительно сокращают перебор возможных векторов движения, тем самым ускоряя алгоритм.

В современных алгоритмах оценки движения шаблонный поиск, в основном, используется для финального уточнения векторов, полученных на предыдущих шагах. При этом на разных итерациях могут использоваться различные шаблоны. Наиболее интересные примеры алгоритмов данного класса можно найти в статьях [1], [2].

#### 4.3. Иерархический поиск

Идея алгоритмов данной группы заключается в следующем. Перед началом поиска производится вычисление  $N-1$  уменьшенных «копий» текущего и опорного кадров, при этом каждая очередная копия в  $2n$  ( $n$  – натуральное число) раз меньше предыдущей (см. Рис. 3). Парты кадров одинакового размера будем называть уровнями, т.е. на одном уровне опорный и текущий кадры одинакового размера. Тогда все множество пар кадров можно представить  $N$  уровнями. Пронумеруем уровни согласно размеру содержащихся в них кадров от меньшего к большему: 1-й уровень будет содержать кадры минимального размера,  $N$ -й – кадры исходного размера. Процесс оценки движения состоит из  $N$  итераций, на каждой из которых обрабатывается пара кадров из уровня с соответствующим номером, т.е. обработка идет от кадров меньшего размера к большему. На каждой итерации производится ОД каким либо из известных методов, например шаблонным поиском. При этом в качестве стартовой точки на каждой итерации выбирается векторное поле, полученное с предыдущей итерации. Другими словами, каждая очередная итерация производит уточнение векторов, вычисленных на предыдущей итерации. При переходе на очередную итерацию размеры области поиска и блоков, для которых оцениваются вектора, обычно увеличиваются в  $2n$  раз для того, чтобы число блоков в кадре на каждой итерации не менялось.

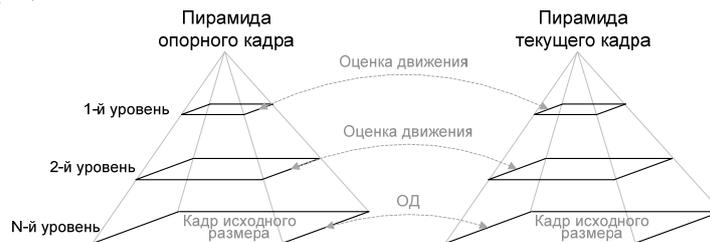


Рис. 3. Схема иерархических уровней

Достоинством алгоритмов данной группы является перебор сокращенного числа векторов, т.е. повышенная вычислительная эффективность. Однако это достоинство нивелируется в случае использования постоянного размера блоков и областей поиска при переходе с одного уровня на другой. Тем не менее, есть аргумент в

пользу того, чтобы фиксировать размер блоков и областей для всех итераций. При фиксированном размере блоков повышается устойчивость векторов в гладких областях, поскольку вероятность попадания контрастных деталей возрастает вместе с ростом площади блока. Это позволяет частично решить одну из основных проблем блочных методов ОД – проблему поиска векторов в гладких областях. Для блоков из таких областей функция соответствия принимает очень близкие (или даже равные, в случае абсолютно гладких областей) значения для различных векторов движения, т.е. эффективность использования ее значений в качестве основного критерия выбора вектора сводится к минимуму. Повысить вероятность успешного нахождения вектора движения позволяет использование блоков большего размера, но это эффективно лишь в случаях, когда размер блока больше размера гладких областей. Возвращаясь к иерархическому подходу, можно сказать, что выполнение ОД при фиксированном размере блока на уровнях с уменьшенными кадрами позволяет повысить вероятность попадания в блок контрастных деталей (и тем самым увеличить вероятность успешного нахождения вектора) по сравнению со случаем изменения размера блока при переходе на следующий уровень.

Постоянный размер области поиска позволяет увеличить максимальную амплитуду векторов движения по сравнению со случаем, когда размер области поиска уменьшается вдвое при переходе на очередной уровень.

Дополнительным преимуществом данного метода является устойчивость к шуму, поскольку во время уменьшения изображений, как правило, удаляются высокочастотные шумы. Однако вместе с высокочастотными шумами могут пропасть и мелкие детали, что приведет к неправильному определению движения в детализированных областях.

Методы данного класса находят широкое применение во многих задачах обработки видео [3], [4]. В работе 3 описывается оригинальный способ уменьшения вычислительной сложности иерархического поиска. Авторы предложили две идеи: улучшенный критерий определения неверно найденных векторов и метод выбора начального уровня иерархии. Уменьшение сложности определения «плохих» векторов достигается за счет использования функции соответствия блоков с меньшей вычислительной сложностью по сравнению с SAD. Начальный уровень иерархии выбирается на основе предположения о близости значений функции соответствия для векторов из небольшой окрестности. Авторы предлагают выбирать начальный уровень иерархии на основе анализа данных, полученных при вычислении значений функций соответствия для векторов из небольшой окрестности.

#### **4.4. Методы, использующие векторы-кандидаты**

Для большинства видео последовательностей справедливо утверждение, что вектора движения соседних блоков очень похожи, так как эти блоки зачастую принадлежат одному движущемуся объекту. Это утверждение привело к появлению целого класса методов ОД, использующих векторы-кандидаты.

Основная идея алгоритмов этой группы очень проста. Перед вычислением информации о движении для текущего блока формируется набор, состоящий из уже вычисленных векторов движения соседних блоков. При этом соседние блоки могут выбираться как в пространственной области, так и во временной. Сформированный набор векторов называется набором кандидатов. В качестве вектора движения в каждом блоке выбирается лучший вектор из набора кандидатов. В качестве критерия поиска обычно используется функция соответствия. Наиболее яркими представителями алгоритмов данной группы являются 3DRS [5] и E3DRS (Enhanced 3DRS, [6]).

Метод 3DRS (3D recursive search) формирует набор векторов-кандидатов из найденных векторов движения со смещениями  $(-1, -1)$  и  $(1, -1)$  в текущем кадре и  $(-2, 2)$ ,  $(2, 2)$  в предыдущем кадре. К первым двум векторам-кандидатам прибавляется равномерно распределенный случайный вектор с амплитудой до  $\pm 3$  пикселей. После этого из полученных кандидатов выбирается вектор с наименьшей SAD. Использование векторов-кандидатов, взятых с различных направлений, позволяет методу 3DRS достаточно быстро сходиться к реальному направлению движения вблизи границ объектов, по сравнению с более простыми методами.

В методе E3DRS используется похожий набор векторов-кандидатов, однако здесь имеется стадия дополнительного поиска вектора с наилучшей SAD по шаблону «малый квадрат» (см. Рис. 2) вокруг выбранного вектора-кандидата. Это обеспечивает лучшие величины SAD, чем у метода 3DRS.

Методы, использующие векторы-кандидаты, часто имеют низкую вычислительную сложность, но при этом обеспечивают гладкость векторного поля, что делает их пригодными для использования в аппаратуре реального времени.

#### **5. Комбинированные методы**

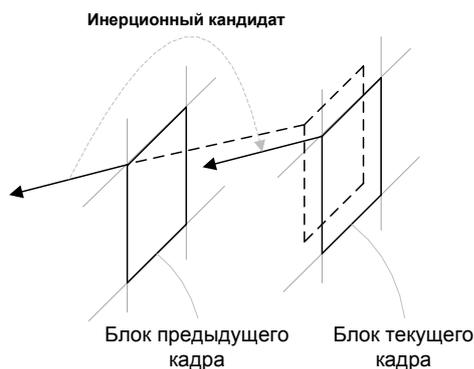
В большинстве современных блочных алгоритмов нахождения движения используются различные комбинации базовых подходов, описанных выше.

Наиболее популярной комбинацией является совместное использование подхода, использующего векторы-кандидаты, и шаблонного поиска. Идея методов данной группы состоит в уточнении лучшего вектора набора с помощью шаблонного поиска. Благодаря

простоте и вычислительной эффективности алгоритмы данной группы достаточно часто становятся предметом интереса исследователей ([7]-[9]). В качестве примера рассмотрим наиболее популярный из современных представителей данной группы алгоритм FAME (Fast Adaptive Motion Estimation), описанный в статье [9].

Этот алгоритм использует приемы для быстрого определения неподвижных блоков, повышения устойчивости поиска векторов в гладких областях, раннего останова поиска для экономии вычислительных ресурсов, а также повышения эффективности использования наборов кандидатов и шаблонов. В рамках данной работы наибольший интерес представляют последняя пара приемов. Рассмотрим их более подробно.

Повышение эффективности использования наборов кандидатов достигается за счет добавления в набор дополнительных векторов. Стандартный набор включает в себя вектора из 3 блоков, находящихся слева, сверху и справа сверху относительно текущего блока. Помимо них в набор добавляются еще два вектора. Первый вычисляется как среднее значение векторов стандартного набора. Второй называется инерционным кандидатом и равен вектору того блока предыдущего кадра, проекция которого на текущий кадр имеет наибольшее пересечение с текущим блоком (см. Рис. 4). При этом проецирование осуществляется вдоль вектора предыдущего кадра. Стоит заметить, что имеется в виду именно предыдущий кадр, а не опорный, т.е. инерционный вектор выбирается из векторного поля, вычисленного алгоритмом ОД для предыдущего кадра. Фактически, этот прием использует предположение о равномерном движении объектов, а сам инерционный кандидат является продолжением траектории движения блока предыдущего кадра. Инерционный кандидат также используется в алгоритмах, описанных в статьях [9], [11], [12]. Наличие указанных двух дополнительных кандидатов позволяет повысить точность поиска векторов.



#### **Рис. 4. Схема вычисления инерционного кандидата**

Новизна в использовании шаблонных методов заключается в следующем. Алгоритм использует шаблоны ББ, МБ и шаблон «эластичный ромб» (ЭР, см. Рис. 2). Использование этих шаблонов при поиске вектора в каждом блоке зависит от нескольких характеристик локальной окрестности текущего блока. В частности, используется величина, характеризующая гладкость векторного поля, и ошибки компенсации соседних блоков. В зависимости от условий, справедливых для этих величин, производится выбор одной из 3 стратегий поиска, при этом в рамках каждой стратегии могут быть использованы не все 3 шаблона. Полное описание стратегий поиска можно найти в работе [9].

#### **6. Заключение**

Алгоритмы каждой из рассмотренных групп имеют свои достоинства и недостатки. Ни один из базовых методов, использованных в чистом виде, не дает приемлемых результатов на практике. Для построения эффективного метода оценки движения необходимо комбинировать приемы, использованные в алгоритмах различных типов, так чтобы недостатки одного метода компенсировались достоинствами другого. Примером такого метода является алгоритм FAME [9], сочетающий в себе использование векторов кандидатов, методов шаблонного поиска, условий раннего останова поиска, способов быстрого определения неподвижных блоков и др.

Блочные алгоритмы оценки движения являются весьма выгодным компромиссом по соотношению вычислительная сложность/точность найденных векторов. Комбинирование приемов из алгоритмов различных категорий в рамках класса блочных методов позволяет построить универсальные алгоритмы ОД, обладающие заданными свойствами и легко реализуемые аппаратно.

#### **7. Литература**

1. S. Zhu and K. K. Ma, "A new diamond search algorithm for fast blockmatching motion estimation," IEEE Trans. Image Processing, vol. 9, pp. 287–290, Feb. 2000.
2. Xuan Jing and Lap-Pui Chau, "An Efficient Three-Step Search Algorithm for Block Motion Estimation", IEEE Transactions on Multimedia, Vol. 6, N. 3, June 2004.
3. Lopes F., Ghanbari M., "Hierarchical motion estimation with spatial transforms", in proceedings of IEEE International Conference on Image Processing, Vol. 2, pp. 558-561, Vancouver, BC, Canada, 2000.
4. Tae Gyoung Ahn, Yong Ho Moon, Jae Ho Kim, "Fast Full-Search Motion Estimation Based on Multilevel Successive Elimination

- Algorithm”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, N. 11, November 2004.
5. G. de Haan, P.W.A.C Biezen, H. Huijgen, and O.A. Ojo, “True Motion Estimation with 3-D Recursive Search Block-Matching”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3, October 1993, pp. 368-388.
  6. S. Olivieri, L. Albani, and G. de Haan, “A low-complexity motion estimation algorithm for H.263 video coding”, Proc. Philips Conf. on Digital Signal Processing, Nov. 1999, paper 17.3, Veldhoven (NL).
  7. P. I. Hosur and K. K. Ma, “Motion vector field adaptive fast motion estimation”, presented at the Second Int. Conf. Inf., Commun., Signal Process., Singapore, Dec. 1999.
  8. A. M. Tourapis, O. C. Au, and M. L. Liou, “Predictive Motion Vector Field Adaptive Search Technique (PMVFAST),” presented in ISO/IEC JTC1/SC29/WG11 MPEG2000, Noordwijkerhout, NL, March 2000.
  9. Ishfaq Ahmad, Weiguo Zheng, Jiancong Luo, Ming Liou, “A Fast Adaptive Motion Estimation Algorithm”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, N. 3, March 2006.
  10. Shih-Yu Huang, Chuan-Yu Cho, and Jia-Shung Wang, “Adaptive Fast Block-Matching Algorithm by Switching Search Patterns for Sequences With Wide-Range Motion Content”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, N. 11, November 2005.
  11. Yongfang Liang, Ishfaq Ahmad, Jiancong Luo, Yu Sun, "On Using Hierarchical Motion History for Motion Estimation in H.264/AVC", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, N. 12, December 2005.
  12. С. Путилин, "Быстрый алгоритм нахождения движения в видеопоследовательностях", Труды конференции Graphicon-2006, с. 407-410, Новосибирск, Академгородок, Россия, Июль 2006.