

УДК 621.397

# ОБНАРУЖЕНИЕ И СОПРОВОЖДЕНИЕ ДВИЖУЩИХСЯ ОБЪЕКТОВ МЕТОДОМ СОПОСТАВЛЕНИЯ БЛОКОВ

**Н. А. Обухова,**

канд. техн. наук, доцент

Санкт-Петербургский государственный университет аэрокосмического приборостроения

*Описан подход к обнаружению и сопровождению движущихся объектов на основе метода сопоставления блоков, применяемого в стандартах видеокodирования: MPEG 1-2 и H. 261/262/263. Приведены алгоритмы сегментации объектов по признакам пространственной и временной корреляции и их дальнейшего сопровождения. Проанализированы причины возникновения аномальных векторов движения и предложены пути, уменьшающие их число: предварительная классификация блоков и медианная фильтрация полученного поля векторов. Рассмотрена адаптация предложенных алгоритмов для сегментации и сопровождения малоразмерных объектов, обладающих медленным движением.*

*The paper describe an approach to detection and tracking moving object based on block matching algorithm, which is the motion estimation algorithm employed in such standards compression as MPEG 1-2 and H. 261/262/263. The algorithm of segmentation based on spare and time correlation is considered. The causes of incorrect moving vectors generation are analyzed and procedures to reduce their quantity such as blocks classification and vector median filter are offered. The adaptation of these algorithm for detection and tracking very slow moving object with small size (point – object) are described.*

## Введение

В настоящее время все более широкое распространение получают системы наблюдения и контроля, использующие ТВ-камеру в виде датчика. К ним относятся различные диспетчерские системы, системы визуального контроля, а также системы наблюдения за объектами, расположенными в недоступных и труднодоступных местах. Область применения систем сегментации и сопровождения объектов существенно расширилась за счет использования их в охранных видеосистемах и в системах мониторинга дорожного движения.

Огромный объем информации, поступающей с ТВ-датчика, требует специальной обработки видеосигнала с целью устранения его избыточности и выделения информации о контролируемом объекте. Соответственно, необходимо решать задачу разработки и исследования новых методов анализа изображений. Кроме того, стремительный прогресс вычислительных устройств, имеющий место в последние годы, снимает ограничения на сложность алгоритмов обработки изображений, что позволяет разрабатывать и применять методы, дающие качественно новый результат.

В статье рассматривается группа алгоритмов, решающих задачу обнаружения, сегментации и сопровождения объектов по признаку движения. Для его оценки используется поле векторов движения, описывающее так называемое видимое движение объектов в кадре. Нахождение векто-

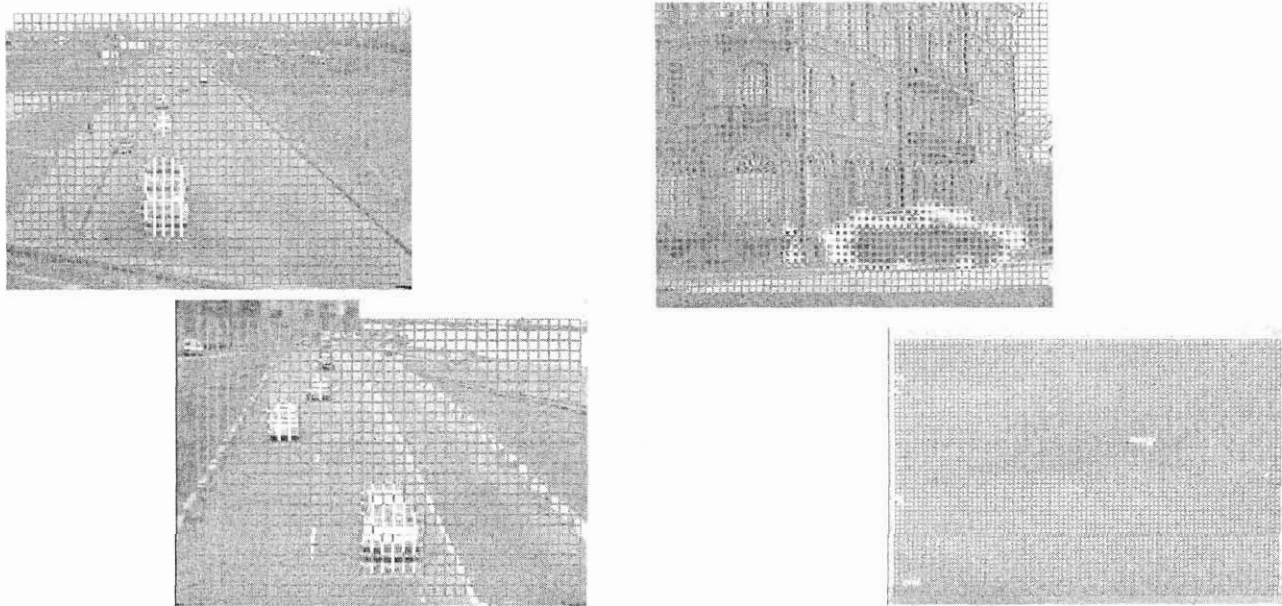
ров движения выполняется методом совмещения блоков, принятым в стандартах видеокodирования: MPEG 1-2 и H. 261/262/263.

## Получение поля векторов движения

Метод совмещения блоков состоит из следующих основных шагов: текущий кадр делится на неперекрывающиеся квадратные блоки размером  $M \times N$  пикселей (обычно  $N=M=8$ , также следует отметить, что размер блока должен быть согласован с размером объектов); для каждого блока формируется область поиска в предыдущем кадре, которая имеет размер:  $(2d+M+1)(2d+N+1)$ , где  $d$  – максимально возможное смещение в горизонтальном и вертикальном направлении; выполняется совмещение блоков текущего кадра с блоками предыдущего кадра видеопоследовательности; определяется, какому блоку в области поиска он лучше всего соответствует, и оценивается величина смещения положения блока в текущем кадре относительно предыдущего – вектор движения.

В качестве критерия, оценивающего степень соответствия между блоком текущего кадра и блоком предыдущего кадра, используется средняя абсолютная разность (MAD):

$$MAD(x, y) = \frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} |F_c(k+i, l+j) - F_p(k+x+i, l+y+j)|,$$



■ Рис. 1. Результат разделения блоков кадра по критерию (1)

где  $F_c$  и  $F_p$  – яркость пикселя в текущем и предыдущем кадре, соответственно;  $(k, l)$  – координаты пикселя левого верхнего угла текущего блока;  $N$  – размер блока ( $N=M$ );  $(x, y)$  – один из возможных векторов движения.

Существует множество различных алгоритмов, позволяющих выполнить согласование блоков. Самым простым и надежным является полный перебор, но из-за большого объема проводимых вычислений он обладает низкой скоростью. Кроме того, существует значительное число алгоритмов, которые пытаются оптимизировать стратегию поиска минимума. Эти алгоритмы можно разделить на следующие группы:

уменьшающие число контрольных точек, основываясь на предположении об унимодальности целевой функции: трехшаговый алгоритм [1], алгоритмы логарифмического [2], ортогонального и поперечного поиска [3, 4], алгоритм поиска по квадрантам [5];

учитывающие возможность присутствия в видеопоследовательности медленных движений (в этом случае векторы движения тяготеют к центральной точке области поиска): блочный градиентный поиск [6], трехшаговый алгоритм [7], четырехшаговый алгоритм [8];

предсказывающие начальное приближение (в этом случае сужается область поиска): иерархические алгоритмы [9] и алгоритмы с предсказанием [10].

Наиболее часто векторы движения определяют с помощью алгоритма 3SS (трехшаговый поиск).

При определении векторов движения путем совмещения блоков неизбежно возникновение аномальных векторов. Присутствие аномальных векторов не устраняется даже при использовании полного перебора. Их возникновение характерно для тех блоков кадра, где отсутствует движение как таковое, а изменение целевой функции обусловлено наличием шумов. Как показало исследование, в основном такими блоками являют-

ся блоки фона, имеющие мультимодальную целевую функцию. Для уменьшения вероятности возникновения аномальных векторов, а также для уменьшения вычислительной сложности алгоритма целесообразно исключить из рассмотрения блоки фона, выполнив предварительное разделение всех блоков кадра на блоки фона и блоки объекта.

Для этого определяется степень корреляции между блоком в текущем кадре и блоком предыдущего кадра, находящимся в этой же позиции. Если уровень корреляции больше некоторого порога, то блок является статичным – фоном и ему присваивается нулевой вектор движения. В противном случае выполняется поиск вектора движения.

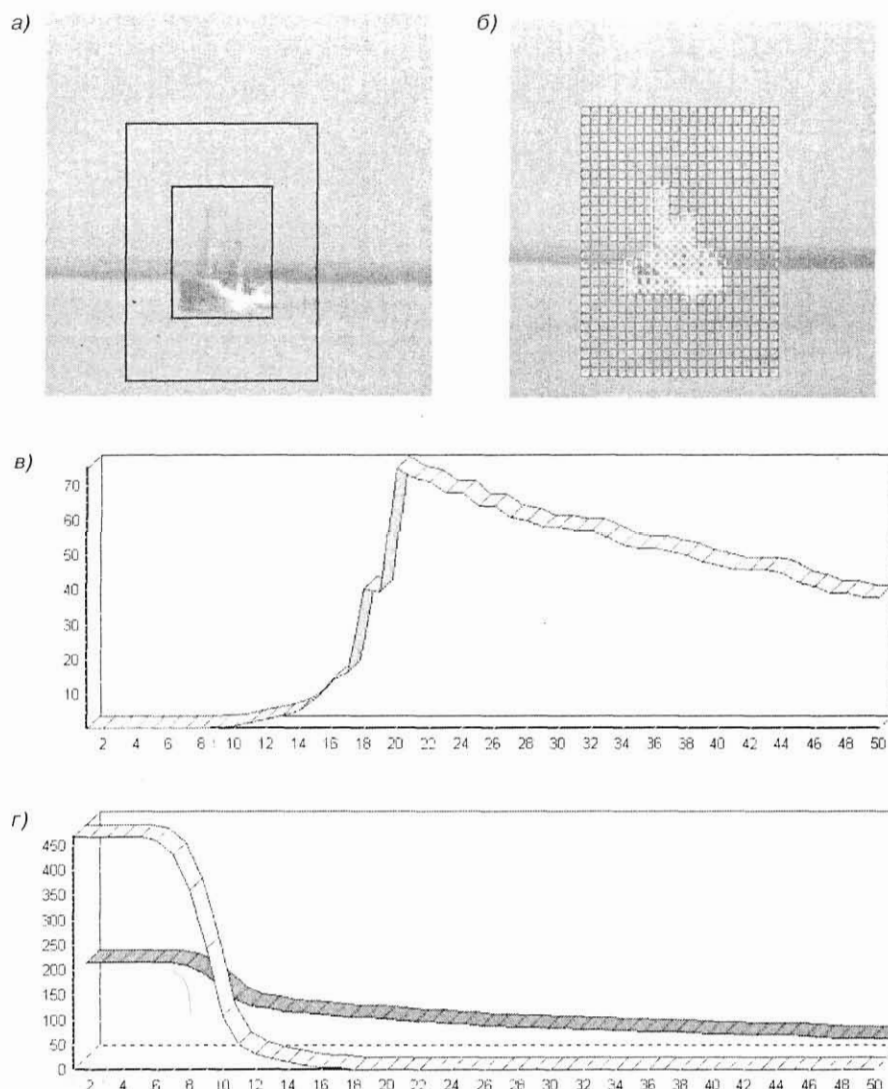
Степень корреляции оценивается по энергии движения для каждого блока (рис. 1). Находится  $MAD$  в каждом блоке, определяется уровень средней  $MAD$  и устанавливается порог 5–10 % от найденного среднего уровня:

$$Porog = 0,05 \frac{1}{Q} \sum_{q=0}^Q \frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} |F_{cq}(k+i, l+j) - F_{pq}(k+i, l+j)|, \quad (1)$$

где  $F_c$  и  $F_p$  – яркость пикселя в текущем и предыдущем кадре соответственно;  $(k, l)$  – координаты пикселя левого верхнего угла текущего блока;  $N$  – размер блока;  $Q$  – число блоков в кадре.

Если в процессе обнаружения объектов принимает участие оператор, указывающий с помощью строга объект, для которого далее решается задача сопровождения, то порог может быть определен автоматически (рис. 2). Алгоритм определения порога заключается в максимизации целевой функции вида

$$Porog = \operatorname{argmax} \frac{P_n}{P_n + P_d + \varepsilon}, \quad (2)$$



■ Рис. 2. Разделение блоков по критерию (1): а – строб и охранная зона; б – результат разделения блоков; в – вид целевой функции; г – число тревог в стробе и охранной зоне: ■ – тревоги в стробе; □ – тревоги в зоне

где  $P_{и}$  – число блоков с истинным движением ( $MAD_{блока} > Порог$  и блок находится внутри строба, накинутого оператором);  $P_{л}$  – число блоков с ложным движением ( $MAD_{блока} > Порог$ , блок принадлежит охранной зоне, автоматически выделяемой вокруг строба, площадь которой должна соответствовать площади строба);  $P_{д}$  – допустимое число блоков с ложным движением;  $\epsilon$  – малая величина для избежания ситуации деления на нуль.

Разделение блоков позволяет:

уменьшить число обрабатываемых блоков: из обработки исключается до 90 % блоков кадра;

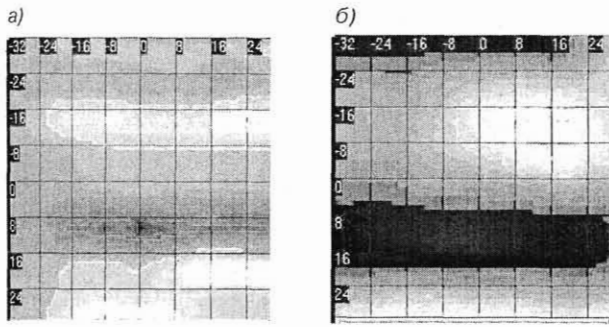
удалить блоки, имеющие мультимодальную целевую функцию, и использовать для поиска векторов движения алгоритмы, предназначенные для поиска минимума унимодальной функции и обладающие высокой скоростью сходимости (например, 3SS);

снизить число аномальных (некорректно определенных) векторов, которые могут появиться из-за присутствия шумов.

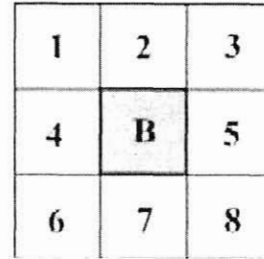
### Предварительная обработка

Исключение из дальнейшей обработки блоков фона существенно снижает количество аномальных векторов, но не исключает их полностью. Это объясняется присутствием у объектов слабо текстурированных блоков. У этих блоков целевая функция является унимодальной, но имеет ярко выраженный плоский участок в районе точки минимума, что затрудняет поиск вектора движения. Рельефы целевой функции блоков с высоким и низким уровнем текстурирования приведены на рис. 3.

Поэтому перед сегментацией объекта по полученному полю векторов необходимо удалить (скорректировать) аномальные векторы, так как их присутствие может вызвать значительную ошибку сегментации, например, привести к расщеплению одного объекта на несколько (рис. 4). Удаление аномальных векторов проводится методом векторной медианной фильтрации [11, 12, 13].



■ Рис. 3. Вид целевой функции у блоков с высоким (а) и низким (б) уровнем текстурирования



■ Рис. 5. Блок В и его восемь «соседей»

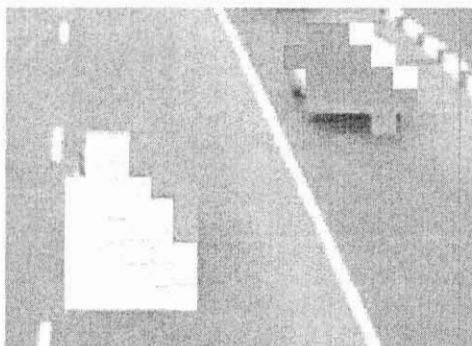
Под векторной медианой множества векторов понимается такой вектор из рассматриваемого множества, у которого сумма расстояний до всех других векторов минимальна. Расстояния между двумя векторами  $\vec{u}(x_u, y_u)$  и  $\vec{v}(x_v, y_v)$  вычисляются на основе нормы  $L_2$ :

$$\|\vec{u} - \vec{v}\|_{L_2} = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}.$$

В применении к задаче удаления аномальных векторов под медианной фильтрацией понимается замена каждого вектора движения векторной медианой множества, составленного из самого вектора и восьми его ближайших «соседей» (рис. 5). При вычислении расстояний используются только ненулевые векторы, т. е. каждый ненулевой вектор превращается в векторную медиану, вычисленную с помощью его восьми ненулевых «соседей». Это делается для того, чтобы избежать замены ненулевых векторов на нулевые в случае, когда в этом соседстве доминируют нулевые векторы. Удаление аномальных векторов с помощью векторной медианной фильтрации значительно ослабляет шум и формирует множество блоков, имеющих более сходные векторы движения. На рис. 6 приведены исходное и обработанное с помощью медианной фильтрации поля векторов.

Таким образом, алгоритм получения поля векторов движения включает следующие шаги:

- выявление блоков, в которых произошли какие-либо изменения, для последующей обработки;
- определение векторов движения с помощью алгоритма 3SS;
- удаление аномальных векторов с помощью медианной фильтрации.



■ Рис. 4. Результат некорректной разметки

### Алгоритм сегментации объектов по полю векторов движения

После получения поля векторов движения необходимо решить задачу выделения объектов (выполнить сегментацию). Признаками для сегментации блоков объекта являются их пространственная и временная корреляция. Блоки, принадлежащие одному объекту, во-первых, должны располагаться в одной области кадра и образовывать связанную группу; во-вторых, должны иметь сонаправленные и близкие по амплитуде векторы движения.

Эти два свойства положены в основу алгоритма разметки. Для оценки близости векторов движения по направлению и амплитуде предлагается использовать норму  $L_2$ .

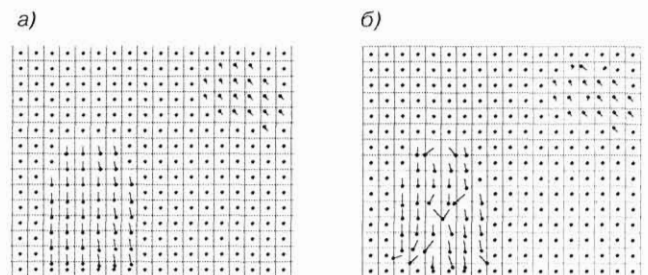
Для учета свойства связности при обработке каждого блока рассматриваются восемь окружающих его блоков – соседей. На рис. 5 показаны 8 «соседей» блока В.

Результатом разметки является матрица с числом элементов, равным числу блоков, содержащая метку для каждого блока. Блоки, образующие связанную группу и имеющие близкие векторы движения, имеют одинаковые метки.

Разметка выполняется на основе следующей рекурсивной процедуры.

Шаг 1. Последовательно перебираются все блоки кадра. Для каждого блока, имеющего ненулевой вектор движения, проверяется наличие метки. Если блок не имеет метки, то формируется новая метка (номер метки увеличивается на единицу).

Шаг 2. Вокруг анализируемого блока (блока-«затравки») формируется область из восьми сосед-



■ Рис. 6. Поле векторов до (а) и после (б) медианной фильтрации



■ Рис. 7. Результат сегментации цветом (а) и стробом (б)

них блоков. Для каждого из этих блоков проверяется условие  $L_2 < Th$ . Если условие выполнено, то блоку присваивается метка блока-«затравки». Эта процедура повторяется для каждого вновь размеченного блока, который рассматривается как блок-«затравка». Шаг повторяется до тех пор, пока есть хотя бы один вновь размеченный блок, для которого не проанализированы его восемь «соседей». В противном случае происходит переход на шаг 1.

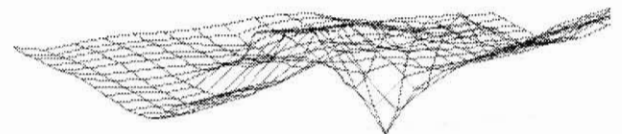
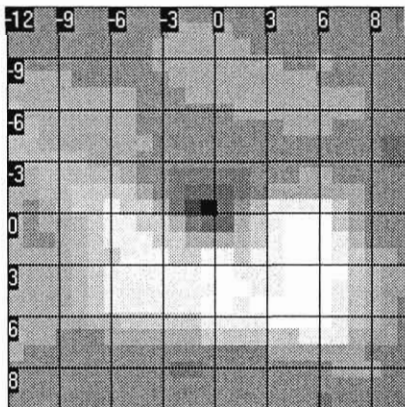
Алгоритм повторяется до тех пор, пока есть хотя бы один неразмеченный блок с ненулевым вектором движения. Результат разметки показан на рис. 7.

Присутствие трех признаков сегментации (скорость, направление движения и пространственная корреляция блоков) позволяет обнаружить и сегментировать несколько движущихся объектов одновременно, даже если они находятся в непосредственной близости друг к другу. При этом чис-

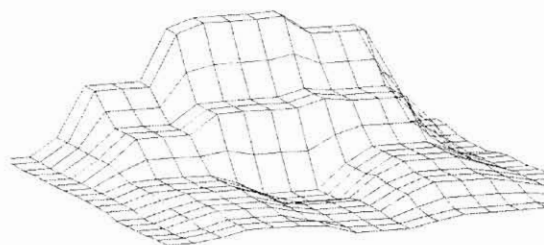
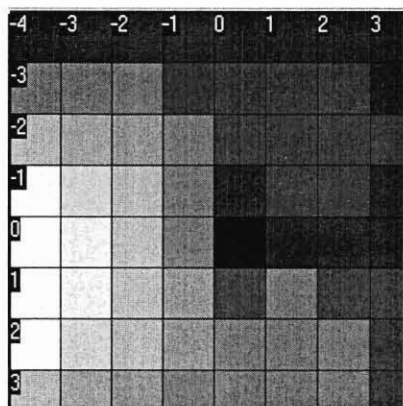
ло различных присвоенных меток определяет количество обнаруженных объектов.

### Особенности определения векторов движения в случае медленно движущихся малоразмерных объектов

В случае медленно движущихся объектов, скорость которых не превышает один пиксель за кадр, использование алгоритмов, обычно применяемых для поиска векторов движения (например, 3SS, полный перебор и др.), приводит к тому, что величина последних будет оцениваться как нулевая. Это связано с тем, что смещение блока в зоне поиска при определении вектора движения может быть только дискретным, и его минимальная величина равна 1 пикселю. Следовательно, вектор движения может быть найден только в случае, если смещение блока (объекта) больше одного пикселя. Отсутствие ненулевых векторов движения не позволяет выполнить сегментацию объек-



■ Рис. 8. Вид целевой функции при медленном движении (блок 8x8)



■ Рис. 9. Вид целевой функции при медленном движении (блок 4×4)

тов, так как ее основой является признак сонаправленности значимых векторов движения.

Нахождение векторов движения является задачей минимизации, в которой необходимо определить минимум целевой функции, характеризующей степень соответствия (совпадения) двух блоков, на множестве различных положений обрабатываемого блока в области поиска. Проведенное исследование рельефа целевой функции позволило сделать вывод о несимметричном характере целевой функции в районе нуля. Как видно из рис. 8, минимум целевой функции в случае, если блоку присуще медленное движение, находится на интервале между 0 и -1, а не локализован в нуле, что характерно для статичных блоков. В случае величины блока не 8×8 пикселей, а 4×4 смещение точки минимума относительно нуля будет еще более очевидным (рис. 9). Это означает, что вектор движения существует и может быть найден путем интерполяции точки минимума на интервале [0; -1].

Для оценки вектора движения целесообразно использовать квадратичную интерполяцию. По каждому направлению определяется значение целевой функции в трех равноотстоящих друг от друга точках  $a, b, c$  ( $a = b - h, c = b + h, b$  соответствует несмещенному положению блока по данному направлению,  $h = 1$ ). Если полученные значения целевой функции удовлетворяют условию  $f(a) > f(b), f(c) > f(b)$ , то положение точки минимума уточняется согласно выражению

$$\delta = b - \frac{h}{2} \frac{f(c) - f(a)}{f(c) - 2f(b) + f(a)}$$

Так, для рельефа, приведенного на рис. 9, значения целевой функции следующие: горизонталь - 0,241, 0,00, 0,049; вертикаль - 0,09, 0,00, 0,174; вектор движения, соответственно, равен (0,33 - 0,15).

Таким образом, задача нахождения вектора движения решается в два этапа. На первом этапе вектор ищется одним из алгоритмов, позволяющих оценить значение вектора движения дискретно, например, 4SS. Если полученный на первом этапе вектор смещения равен нулю, то определяется наличие дробной части, отдельно для горизонтальной и вертикальной составляющей.

Алгоритм 4SS выбран для поиска векторов движения на первом этапе вместо используемого

ранее 3SS, так как он учитывает возможность присутствия в обрабатываемой видеопоследовательности медленного движения. В нем используется такой же шаблон поиска, как и в 3SS, но начальный шаг берется в два раза меньше - четверть от максимального смещения  $d$  ( $[d/4]$ ) и первая итерация локализуется в центре зоны поиска. Дополнительным отличием алгоритма 4SS от 3SS является уменьшение шага в два раза не на каждой итерации, а только когда точка минимума на текущей итерации является центральной.

Основные шаги алгоритма 4SS:

Итерация 1. Определяется точка минимума из девяти контрольных точек, расположенных в центре области поиска, согласно шаблону алгоритма 3SS при  $d = 4$ . Если найденная точка минимума является центральной, то переход к итерации 4; иначе - к итерации 2.

Итерация 2. Шаг сохраняется равным 4 ( $d = 4$ ). Схема поиска определяется положением точки минимума, найденной на предыдущем шаге.

Если предыдущая точка минимума является угловой, используются пять дополнительных контрольных точек.

Если предыдущая точка минимума расположена в середине горизонтальной или вертикальной оси предыдущего окна поиска, используются три дополнительных контрольных точки.

Если найденная на итерации 2 точка минимума расположена в центре окна поиска, переход к итерации 4; иначе - к итерации 3.

Итерация 3. Стратегия поиска такая же, как и на второй итерации, но после нахождения точки минимума совершается безусловный переход на итерацию 4.

Итерация 4. Шаг поиска принимается равным 1 ( $d = 1$ ). Найденная точка минимума (среди новых девяти контрольных точек) рассматривается в качестве искомого вектора движения.

Для эффективной работы описанного алгоритма необходимо использовать блок размером 4×4 пикселя, который дает более ярко выраженную несимметрию целевой функции в области нуля. С другой стороны, использование маленького блока приводит к существенному росту числа аномальных векторов за счет присутствия шумов.

Для снижения уровня шумов исходные кадры обрабатываются пространственным и временным фильтром, с весовыми коэффициентами, соответствующими функции Гаусса. Первоначально

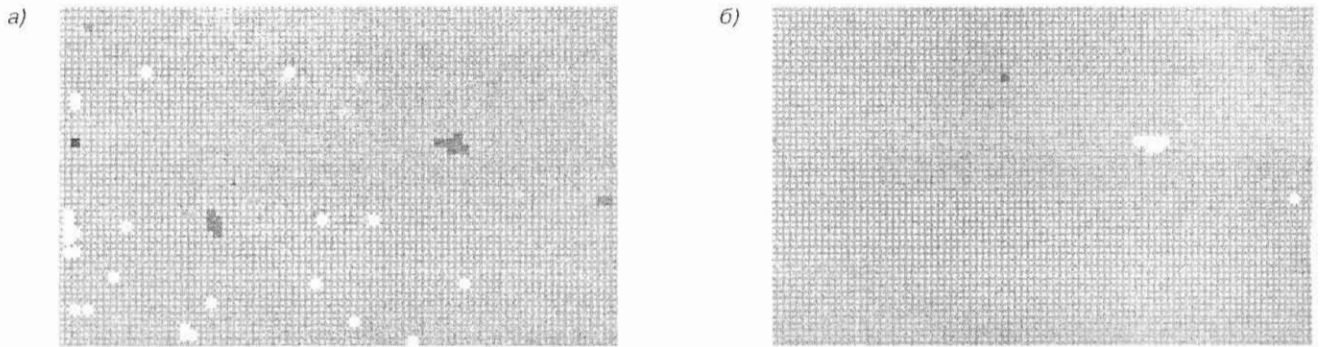


Рис. 10. Результат сегментации до (а) и после (б) пространственно-временной фильтрации

каждый кадр подвергается пространственной фильтрации с маской  $5 \times 5$ . Затем выполняется накопление девяти кадров. Обрабатываемый кадр получается путем взвешенного суммирования пикселей четырех предыдущих и четырех последующих кадров [14].

Результат сегментации в случае применения пространственно-временной фильтрации представлен на рис. 10.

### Алгоритм сопровождения

Результатом работы алгоритма сегментации является выявление блоков, принадлежащих одному объекту, и присвоение им всем одной и той же метки: группа блоков, составляющих один объект, имеет одинаковую метку. При этом метка, присвоенная группе блоков в кадре  $t$ , будет отличаться от метки, присвоенной этой же группе в кадре  $t-1$ . Такая динамика объясняется появлением новых объектов или исчезновением каких-либо объектов в кадре  $t$ . Алгоритм сегментации просматривает блоки, начиная с левого верхнего угла кадра, и после обнаружения связанной группы блоков увеличивает значение метки, присвоенной ранее. При появлении нового движущегося объекта, расположенного выше и левее сегментированного ранее объекта, метка последнего увеличивается на единицу. Если же происходит исчезновение объекта, расположенного выше и левее обрабатываемого объекта, то метка

последнего уменьшается на единицу. В правой части рис. 11 показана разметка объекта в кадре  $t$ , а в левой – разметка объекта в кадре  $t-1$ .

Для решения задачи сопровождения необходимо найти соответствие между меткой объекта в кадрах  $t$  и  $t-1$ . Алгоритм [15] включает в себя следующие шаги:

блоки, размеченные как один объект в кадре  $t$ , сдвигаются назад на величину найденного для них вектора движения;

определяется, какое количество пикселей, размеченных как один объект в кадре  $t$ , приходится на ту или иную метку кадра  $t-1$ .

ищется метка кадра  $t-1$ , на которую пришлось максимальное количество пикселей, размеченных как один объект в кадре  $t$ ; на нее заменяется метка объекта в кадре  $t$ .

Таким образом, сохраняется предыдущая метка объекта.

Поясним алгоритм на примере (см. рис 11). В кадре  $t$  (текущем) блоки, размеченные как один объект, имеют метку 3. После смещения их назад на величину найденного для них вектора движения в кадре  $t-1$  (предыдущем) им соответствуют блоки, имеющие метки 5 и 6, и неразмеченные блоки. Для меток 5 и 6 определяется, сколько пикселей объекта им соответствует (будет определено  $n$  пикселей, имеющих метку 5, и  $m$  пикселей, имеющих метку 6). Из рисунка видно, что в данном случае  $m$  значительно меньше  $n$ . Следовательно, метка объекта 3 в кад-

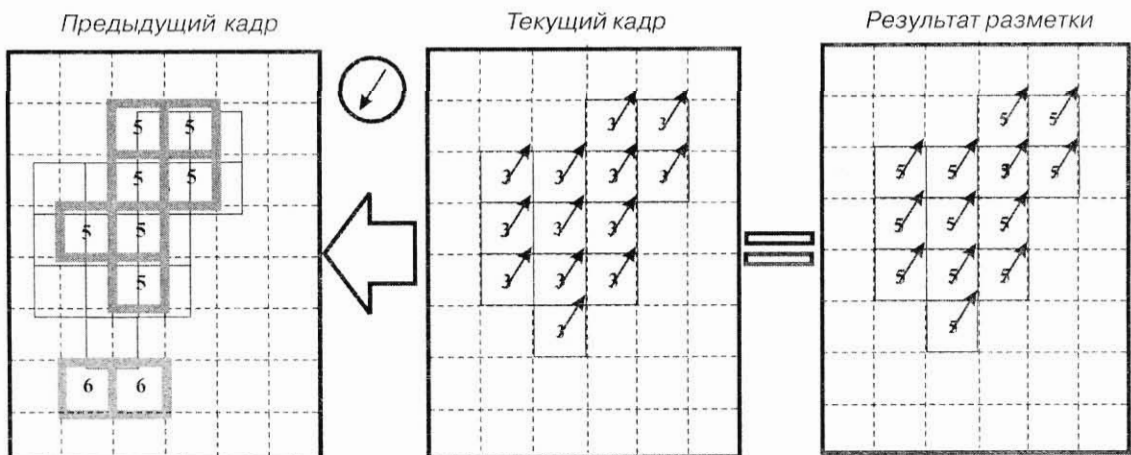
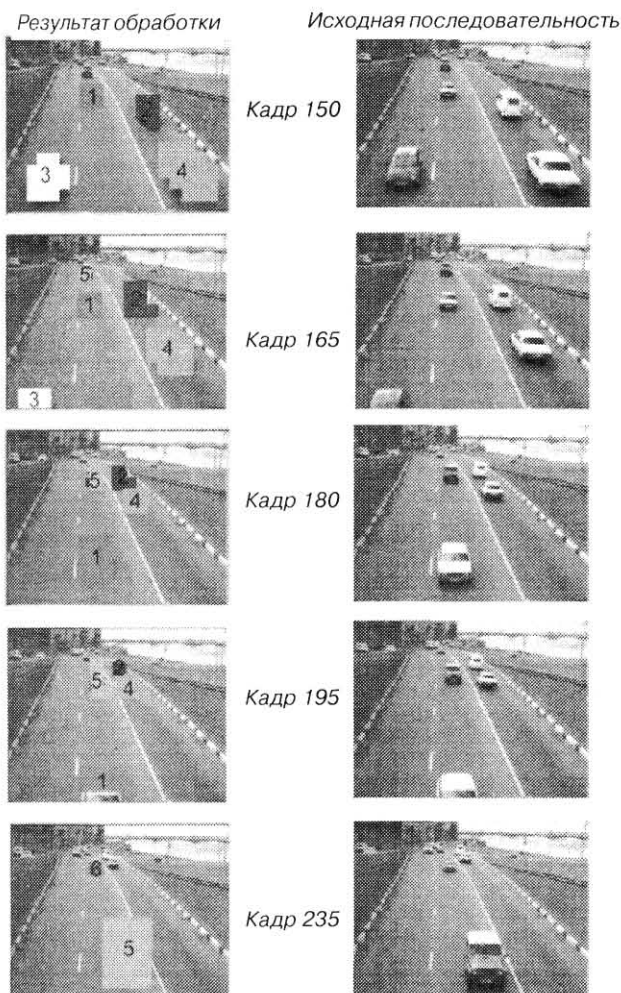


Рис. 11. Сопровождение объекта по составляющим его блокам



■ Рис. 12. Результат работы алгоритма сопровождения

ре  $t$  будет изменена на 5. Результаты работы алгоритма приведены на рис. 12. С момента обнаружения до момента исчезновения каждый из объектов размечен одним цветом, т. е. во всех кадрах видеопоследовательности объекты сохраняют свою метку. Это позволяет накапливать информацию о размерах, направлении и скорости движения объектов, решать задачу траекторного предсказания.

### Выводы

Результаты проведенного моделирования показывают, что рассмотренные алгоритмы (нахождение поля векторов движения с использованием корреляционного порога, медианная фильтрация, сегментация и сопровождение) эффективны при обнаружении и сопровождении протяженных движущихся объектов. Эти алгоритмы были положены в основу программного обеспечения для определения основных характеристик транспортных потоков в ТВ-системе экологического мониторинга: числа движу-

щихся транспортных средств, их типа (грубая классификация) и скорости движения [16].

Использование процедуры интерполяции при определении векторов движения существенно расширяет область применения рассмотренных алгоритмов и позволяет достаточно успешно использовать их для сегментации и сопровождения малоразмерных объектов, обладающих медленным движением.

Основными задачами будущих исследований являются адаптация алгоритмов для работы с медленно движущимися протяженными объектами и решение проблемы перекрытия объектов фоном и друг другом (окклюзии объектов).

### Литература

1. Koga T., Linuma K., Hirano A. and oth. Motion compensated interframe coding for video conferencing // NTC81. – Nov. 1981. – P. G 531–G 535
2. Jain J. R., Jain K. A. Displacement measurement and its application in interframe image coding // IEEE Trans. Commun. – 1981. Vol Com. 29. – P. 1799–1806
3. Puri A., Hang H. M., Shilling D. L. An efficient block matching algorithm for motion – compensated coding // IEEE Int. Conf. Acoustic, Speech and Signal Processing. – 1987. – Vol. 19. – P. 1063–1066
4. Chanbani M. The cross-search algorithm for motion estimation // IEEE Trans. Commun. – 1990. – Vol. 38. – P. 950–953
5. Liu L. K., Feig E. A block-based gradient descent search algorithm for block motion estimation in video coding // IEEE Trans, Circuit Syst. Videotech. – 1996. – Vol. 6. – P. 108–115.
6. Li R., Zeng B., Liou M. L. A new three-step search algorithm for block motion estimation // IEEE Trans, Circuit Syst. Videotech. 1994. – Vol. 4 (1). – P. 438–442
7. Po L. M., Ma W. C. A Novel four-step search algorithm for fast block estimation // IEEE Trans, Circuit Syst. Videotech. – 1996. – Vol. 6 (3). – P. 313–317
8. Shi Y. Q., Xia X. A Thresholding Multiresolution block matching algorithm // IEEE Transactions on circuits and systems for video technology. – 1997. – Vol. 7 (2). – P. 437–440
9. Xu J. B., Po L.M., Cheung C. K. Adaptive motion tracking block matching algorithms for video coding // IEEE Transactions on circuits and system for video technology. – 2000. – Vol. 10 (3). – P. 417–422
10. Astola J., Haavisto P., Neuvo Y. Vector Median Filters // IEEE. 1990. – Vol. 78 (4). – P. 237–260.
11. Bartolini F., Cappellini V., Colombo C., Mecocci A. Enhancement of local optic flow techniques // 4th International Workshop on Time Varying Image Processing and Moving Object Recognition, Italy. 1993. – P. 359–366.
12. McCandless J. W. Detection of aircraft in video sequences using a predictive optical flow algorithm // Optical Engineering. 1999. – N 3. – P. 523–530
13. Stefano L. D., Viarani E. Vehicle detection and tracking using the block matching algorithm // www.labvisione.deis.unibo.it
14. Обухова Н. А. Цифровая обработка изображений для оценки характеристик транспортных потоков в ТВ-системе экологического мониторинга воздушной среды города. Деп. в ВИНТИ № 1065-В2003.