

БЛОЧНЫЕ АЛГОРИТМЫ ОЦЕНКИ ДВИЖЕНИЯ

Аннотация: Субоптимальные алгоритмы оценки движения разрабатываются, исходя из компромисса между сложностью вычислений, качеством работы и объемом выходного потока данных. В данной статье рассматриваются наиболее известные блочные алгоритмы оценки движения, такие как трехшаговый, четырехшаговый, ортогональный, логарифмический и иерархический, сопоставляется их сложность и качество работы.

Задача двумерной оценки движения состоит в нахождении для изображений $f(\vec{x}, n-1)$ и $f(\vec{x}, n)$, где \vec{x} - двумерная позиция на изображении и n - номер изображения, такого поля векторов $\vec{d}(\vec{x}, n)$, что выполняется выражение (1).

$$f(\vec{x} - \vec{d}(\vec{x}, n), n-1) \approx f(\vec{x}, n) \quad (1)$$

Широко применяется подход к решению этой задачи, при котором полагается, что вектор движения одинаков для некоторого блока пикселей, т.е. $\vec{d}(\vec{x}, n) = \vec{d}(\vec{x}', n)$ при $\forall \vec{x}' \in B(\vec{x})$, где $B(\vec{x})$ - блок пикселей на позиции \vec{x} . На этом принципе основывается большое количество алгоритмов поиска соответствия блоков.

Выбор размера блока основывается на компромиссе между объемом вычислений и точностью. Чем больше размер блока, тем меньше их в кадре и меньшее количество векторов движения нуждается в передаче. Но границы движущихся объектов обычно не совпадают с границами блоков, поэтому при больших размерах блока необходимо передать больше корректирующей информации [2]. Блокам малого размера сопутствует большее количество векторов движения, однако вероятность соответствия опорного и искомого блоков больше, поэтому корректирующей информации потребует передать меньше. Разбивка кадра на слишком маленькие блоки приводит к тому, что вся система сжатия становится очень восприимчивой к шуму [3]. Поэтому размер блока является компромиссом между минимальным числом векторов движения и максимальным качеством поиска совпадения между блоками.

При сегментация текущий кадр видеопоследовательности разбивается на неперекрывающиеся блоки одинакового размера, чаще всего используются квадратные блоки. Предпочтительны размеры, являющиеся степенью двойки, доминирующими являются 8 и 16 пикселей. Есть примеры использования блока размером 16 на 8 пикселей, основываясь на том факте, что движение чаще всего происходит в горизонтальном направлении по рас-

тру, нежели в вертикальном [1]. Стандарты MPEG-2 и H.261 используют блоки 16 на 16 пикселей.

Вектор движения для блока выбирается из условия минимальности определяющей критерий совпадения блоков функции стоимости [5], которой в общем случае является сумма абсолютных разностей (SAD) между элементами текущего и опорного блоков (2).

$$SAD(V_x, V_y) = \sum_{x,y} |F(x, y, t) - F(x - V_x, y - V_y, t - 1)| \quad (2)$$

Здесь F - значение яркости, t - временной индекс кадра, (x, y) - пространственные координаты пикселей в кадре, V_x, V_y - вектора движения.

Поиск соответствия блоков – наиболее затратная по времени операция. Зона поиска необязательно должна быть квадратной, так как движение в основном наблюдается в горизонтальном направлении [1], прямоугольные области используются чаще всего. Например, кодер стандарта MPEG-2 CLM4500 использует зону поиска с ± 48 пикселей по горизонтали и ± 24 пикселя по вертикали с полупиксельной точностью. Дробная точность достигается за счет интерполяции, обычно используется билинейная интерполяция, определяемая следующими соотношениями:

$$U(2x_0, 2y_0) = K_j(x_0, y_0) ;$$

$$U(2x_0 + 1, 2y_0) = [K_j(x_0, y_0) + K_j(x_0 + 1, y_0)] / 2 ; \quad (3)$$

$$U(2x_0, 2y_0 + 1) = [K_j(x_0, y_0) + K_j(x_0 + 1, y_0)] / 2 ;$$

$$U(2x_0 + 1, 2y_0 + 1) = [K_j(x_0, y_0) + K_j(x_0 + 1, y_0) + K_j(x_0, y_0 + 1) + K_j(x_0 + 1, y_0 + 1)] / 4$$

где U – интерполированный кадр. На рис. 1а и 1б показаны области кадра K_j до билинейной интерполяции и после интерполяции, x_0, y_0 - координаты верхней левой точки.

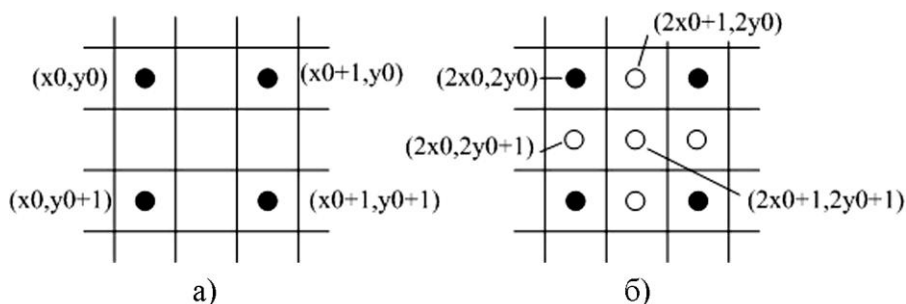


Рис. 1. Интерполяция пикселей.

Поиск соответствия блоков, при котором перебираются все блоки из зоны поиска называется полным (FS). Чем больше зона поиска, тем больше вероятность найти совпадающие блоки, но при увеличении зоны поиска квадратично увеличивается число переби-

раемых блоков. По качеству предсказанного изображения полный поиск дает наилучший результат, однако имеет очень большой объем вычислений (см. Табл.1). Существуют так называемые субоптимальные алгоритмы оценки движения, основывающиеся на компромиссе между сложностью вычислений и качеством работы.

Например, бинарный поиск, при котором зона поиска разбивается на несколько частей, в одной из которых осуществляется полный поиск (рис. 2а). Сначала вычисляются функции стоимости для девяти начальных точек (черные кружки на рис. 2а). В зоне, соответствующей точке с наименьшим значением функции стоимости, выполняется полный поиск.

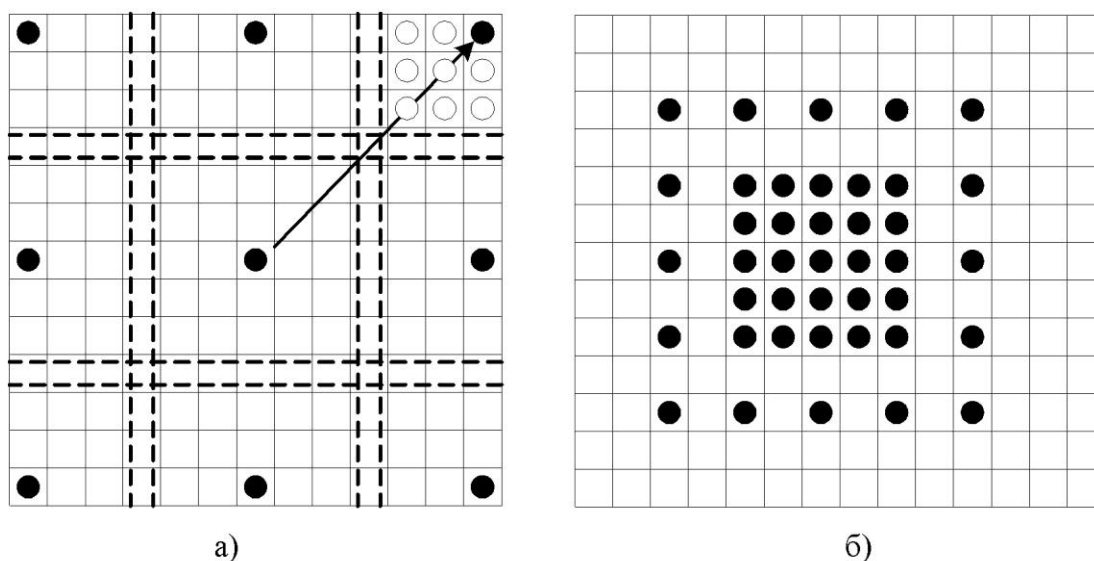


Рис. 2. Возможные пути сходимости бинарного поиска (а) и поиска с «грубым квантованием» вектора движения (б).

Для снижения объемов вычислений существуют алгоритмы, использующие свойство зрения человека, нечувствительное к небольшим погрешностям при оценке движения. Такие алгоритмы осуществляют так называемое грубое квантование векторов движения, при котором вблизи центра зоны поиска перебираются все блоки, а на определенном расстоянии от центра некоторые блоки не участвуют в поиске (рис. 2б).

Трехшаговый поиск [4] ищет вектора движения, переходя от более грубого к более точному шаблону поиска. Он состоит из следующих шагов: 1 - рассматриваются восемь блоков на начальном расстоянии от центра (текущего блока). 2 - начальное расстояние между центрами блоков уменьшается вдвое, центр поиска сдвигается в точку с минимальными искажениями. Этапы 1 и 2 повторяются до тех пор, пока начальный шаг не будет меньше единицы. На рис. 3б приведена возможная схема сходимости трехшагового алго-

ритма поиска. Недостатком трехшагового поиска является однородное расположение точек в первом шаге, которое неэффективно в случае медленного движения.

У четырехшагового поиска начальное расстояние между точками (центрами блоков) равно двум. Процесс поиска включает в себя следующие ступени. 1 – выбрать девять точек в зоне поиска. Вычислить искажения и найти точку с наименьшими искажениями. Если эта точка является центром зоны поиска, то перейти к шагу 4, иначе – к шагу 2. 2 - сместить центр в точку с наименьшими искажениями. Шаг остается неизменным. Если точка минимума искажений расположена с края зоны поиска, то рассматриваются 5 точек, если точка минимума искажений расположена посередине горизонтальной или вертикальной оси предыдущего окна поиска, то дополнительно рассматриваются 3 точки (см. Рис. 2).

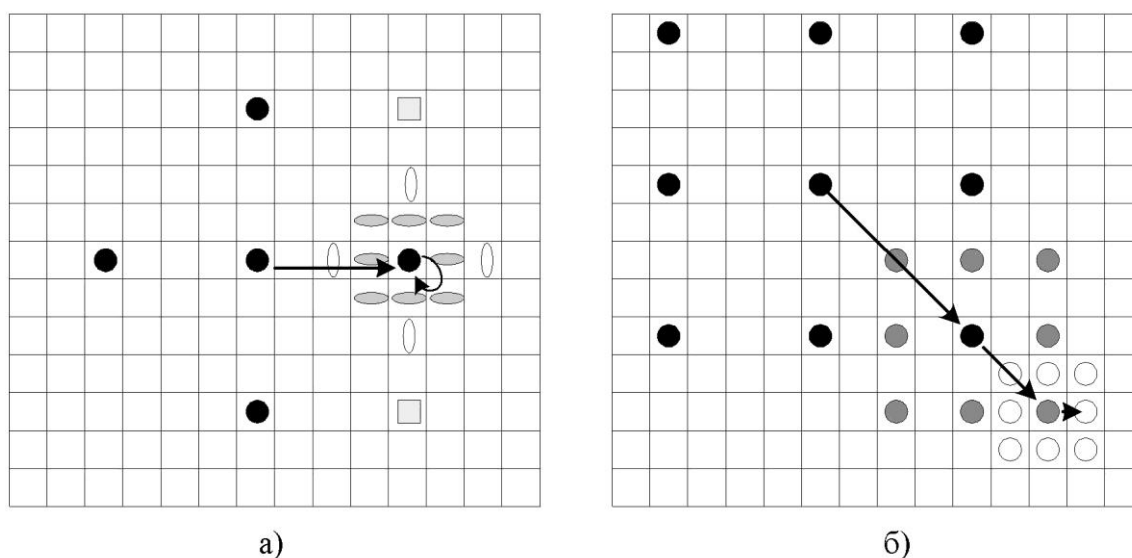


Рис. 3. Возможные путь сходимости двумерного логарифмического (а) и трехшагового (б) алгоритмов поиска.

Логарифмический поиск стал известен в то же время, что и трехшаговый. Он требует большее количество шагов выполнения, но возможно достижение большей точности, особенно когда зона поиска достаточно большая. Схема выполнения логарифмического поиска следующая. 1 - сначала рассматриваются пять блоков, расположенных в форме знака «+». 2 - если блок в середине является наилучшим кандидатом среди остальных, то расстояние между блоками сокращается вдвое. Если лучшим является один из краевых блоков – центр поиска смещается туда, и шаг 1 повторяется. 3 - когда расстояние между блоками становится равным единице, выполняется поиск среди всех девяти блоков, окру-

жающих центр поиска. Существует несколько реализаций этого алгоритма, которые отличаются выбором расстояния между блоками.

Ортогональный поиск (рис. 4б) является гибридом трехшагового и логарифмического поисков. В нем попеременно пробуются горизонтальное и вертикальное направления. 1 - сначала выбирается начальное расстояние между блоками, обычно оно равно половине максимально возможному смещению в зоне поиска. После перебора трех блоков, центр поиска смещается в точку, соответствующую наилучшему блоку. 2 - такая же операция повторяется для вертикального направления. 3 - расстояние между блоками уменьшается вдвое, если оно больше единицы, иначе поиск закончен.

Также среди известных схем поиска присутствуют спиральный поиск, комбинирующий идеи трехшагового и бинарного поисков, и кросс-поиск, похожий на логарифмический, но отличающийся от него шаблоном.

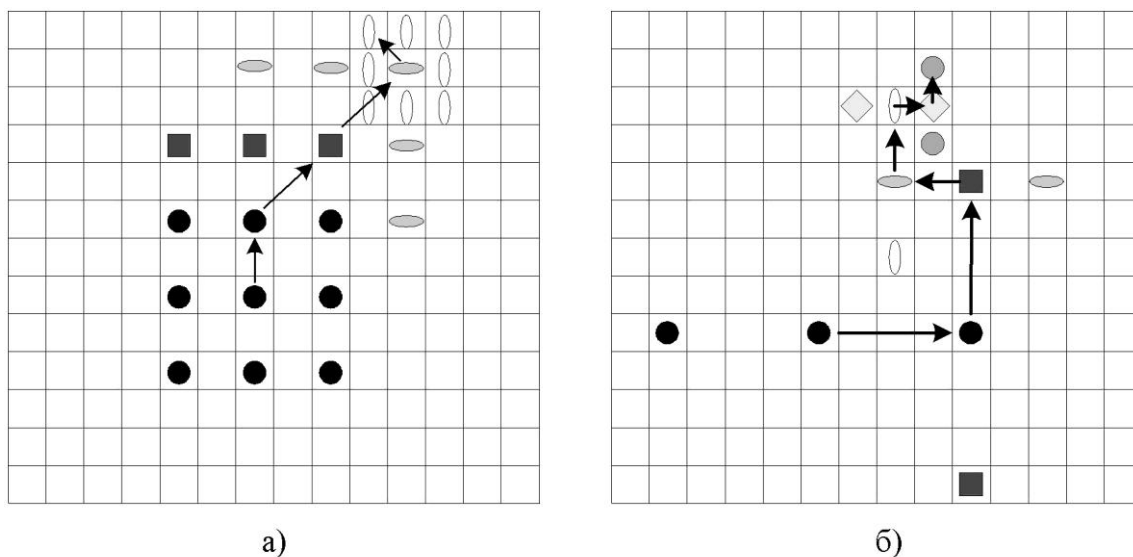


Рис. 4. Возможные пути сходимости четырехшагового (а) и ортогонального (б) алгоритмов поиска.

На практике также используются иерархические алгоритмы поиска, в которых вектор движения ищется от грубого на изображениях с низкими разрешениями к точному на оригинальном изображении. Пример такого алгоритма – средняя пирамида [6]. Пирамида строится с помощью фильтра нижних частот (4), где $g_L(x, y)$ - уровень яркости пиксела с координатами (x, y) уровня L пирамиды.

$$g_L(x, y) = \frac{1}{4} \cdot \left(\sum_{u=0}^1 \sum_{v=0}^1 g_{L-1}(2x+u, 2y+v) \right) \quad (4)$$

Один пиксел уровня 2 соответствует блоку 4x4 и блоку 2x2 пикселов на нулевом и первом уровнях соответственно. На верхнем уровне пирамиды используется трехшаговый поиск с некоторыми изменениями [6], на остальных уровнях – полный поиск в небольших окнах. Иллюстрация средней пирамиды приведена на рис. 5. Иерархический поиск хорошо приспособлен для совместного использования с вейвлет-преобразованиями.



Рис. 5. Схема иерархического поиска.

Табл. 1. Сравнение алгоритмов оценки движения по параметрам качество и время выполнения.

Алгоритм	Качество	Время
Полный	100	100
Трехшаговый	81,1	3,65
Четырехшаговый	79	2,2
Логарифмический	73,6	1,27
Ортогональный	78,3	2,09
Иерархический (средняя пирамида)	96	3,5

В таблице 1 приведены нормированные показатели качества, исходя из средних значений функции стоимости, и времени вычислений для некоторых из рассмотренных алгоритмов оценки движения на примере 10-кадровой QCIF видеопоследовательности. Из нее следует, что лучшим по скорости является ортогональный алгоритм, а лучшим по качеству – иерархический. Эти два показателя являются главными, на которые опираются при сравнении алгоритмов.

Дальнейшие улучшения блочных алгоритмов можно вести по разным направлениям. Во-первых изменяя шаблон поиска, например, использовать шестиугольный. Вторым вариантом может быть адаптивное уменьшение размеров блока в зависимости от участков

движения на изображении. Улучшение методов интерполяции, используемой для мелко-пиксельной точности также ведет к улучшению всего алгоритма. Ресурсы для улучшения есть у алгоритмов оценки движения, использующих пространственную, временную и фазовую корреляции.

Литература

1. E.Chan, A. Rodriguez, R.Ghandi, S.Panchanathan. Experiments on block-matching techniques for video coding. *Multimedia Systems*, Volume 2, Number 5, December 1994, pp. 228 – 241.
2. D. Lin, Cheng-Tie Chen, T. Hsing. Video on phone lines: technology and applications. *Proceedings of the IEEE*, Volume 83, Number 2, February 1995, pp. 175 – 193.
3. F. Lallauet, D. Barba/ Motion compensation by block matching and vector post-processing in sub-band coding of TV signals at 15 Mbit/s. *Proc. SPIE*, Vol. 1605, pp. 26 – 36.
4. Т.Коба, К. Иинумо, А. Хирано, Y. Иижима, Т. Ишигуро. Motion-compensated interframe coding for video conferencing. *Proc. NTC81*, New Orleans, LA, Nov. 1981, pp.C9.6.1 - 9.6.5.
5. Зубарев Ю.Б., Дворкович В.П., Нечепанев В.В., Соколов А.Ю. Методы анализа и компенсации движения в динамических изображениях // *Электросвязь*, №11, 1998, С. 15 – 21.
6. Kwon Moon Nam, Joon-Seek Kim, Rae-Hong Park. A Fast Hierarchical Motion Vector Estimation Algorithm Using Mean Pyramid. *IEEE Transactions on Circuits and Systems for Video technology*, vol. 5, no.4, August 1995, pp. 344 – 351.