

# Метод повышения эффективности временной сегментации видео на сценах с медленным движением

Гришин С.В., Ватолин Д.С.  
Лаборатория Компьютерной Графики при ВМиК  
Московский Государственный Университет им. М. В. Ломоносова  
Москва, Россия  
{sgrishin, dmitriy}@graphics.cs.msu.ru

## Аннотация

В данной статье описывается метод повышения эффективности алгоритмов сегментации видео на сценах с медленным движением. Улучшение результатов сегментации достигается за счет адаптивного выбора пар кадров для вычисления информации о движении. К достоинствам метода относятся простота, позволяющая использовать его в алгоритме сегментации без существенного увеличения вычислительной сложности последнего, и возможность с его помощью увеличить эффективность произвольного алгоритма сегментации видео, использующего информацию о движении. Результаты использования метода, представленные в конце статьи, доказывают его эффективность.

**Ключевые слова:** автоматическая сегментация, оценка движения, медленное движение.

## 1. ВВЕДЕНИЕ

Алгоритмы автоматической сегментации делятся на автоматические и полуавтоматические (т.е. с участием пользователя). Особый интерес представляют методы автоматической сегментации, именно они рассматриваются ниже. Методы пространственной сегментации используют только информацию о яркости/цвете изображения (например [1], [2]). Выделение объектов в таких методах производится на основе близости (по какой-либо метрике) цветовых характеристик соседних областей кадра. Как показывает практика, цветовой информации бывает недостаточно для получения приемлемых результатов сегментации. Например, в случае пространственной сегментации текстурного объекта или объекта, состоящего из нескольких цветовых регионов с выраженными границами, будет найдено несколько объектов, каждый из которых будет соответствовать определенному цвету (в то время как на самом деле, на изображении один объект). В случае видео, использование информации о движении позволяет значительно повысить качество сегментации. Именно поэтому одним из общепринятых подходов к автоматической сегментации является пространственно-временная сегментация ([4], [5]). Данный подход состоит в выполнении этапов пространственной и временной сегментации с последующим объединением их результатов. Временная сегментация производится на основе информации о движении в кадре. При этом результат временной сегментации тем лучше, чем больше разница между параметрами движения различных объектов. На практике нередки случаи, когда различия в движении объектов и фона минимальны или отсутствуют вообще

(движение такого типа назовем *слабым*). В таких случаях временная сегментация перестает работать правильно.

Метод, описанный в данной статье, позволяет значительно повысить качество временной сегментации на сценах со слабым движением. При этом точность сегментации в условиях *нормального* (не слабого) движения не падает.

## 2. ОПИСАНИЕ АЛГОРИТМА

Основой алгоритмов временной сегментации является информация о движении, которая вычисляется при помощи методов ОД (оценки движения). Метод ОД производит сравнение двух кадров и на основе найденных между ними соответствий вычисляет параметры движения каждой области кадра. Как уже было сказано, в случае слабого движения разница между последовательными кадрами видео незначительна, и поэтому, используя полученную на выходе ОД информацию о движении, даже человеку бывает трудно различить области фона и объектов. Это приводит к тому, что польза от использования информации о движении при сегментации резко снижается.

Описанный в данной статье алгоритм повышения эффективности временной сегментации состоит в адаптивном выборе кадров, которые подаются на вход методу ОД. Основная идея алгоритма состоит в том, что на сценах с медленным движением поиск движения производится не между последовательными кадрами, а между кадрами, отстоящими друг от друга во времени на достаточном расстоянии. Такой прием позволяет увеличить разницу между параметрами движения фона и объектов по сравнению со случаем использования последовательных кадров, тем самым облегчая задачу алгоритму сегментации.

При описании разработанного алгоритма будем предполагать, что фон является статическим. В общем случае, выполнения этого условия можно достичь, используя метод оценки глобального движения (напр. [3]). Для этого в качестве информации о движении следует использовать поле векторов движения (ПВД), полученное в результате вычитания ПВД, вычисленного обычным методом ОД, из ПВД, вычисленного методом глобальной оценки движения.

Расстояние между кадрами, которые используются для вычисления движения, называется *временным шагом ОД*. Таким образом, основной идеей разработанного алгоритма является динамическое изменение временного шага ОД. В качестве алгоритма ОД использовался метод, основанный на поиске соответствия между блоками кадров, поэтому далее будет предполагаться, что вектор движения определен для каждого блока. Традиционно, поле векторов движения

вычисляется для пары кадров – текущего  $t$  и предыдущего  $t-1$ . В разработанном алгоритме вычисление движения дополнительно производится для кадров  $t$  и  $t-k$ , где  $k$  – временной шаг ОД. При этом выбор значения  $k$  должен удовлетворять следующим требованиям:

- чем медленнее движение объектов, тем больше должен быть  $k$ ;
- значение  $k$  должно быть таким, чтобы ОД производилась с допустимой точностью. Другими словами, неприемлема ситуация, когда вектора движения для некоторых объектов вычисляются неверно, вследствие слишком большого значения  $k$ . Данное требование обязательно, поскольку неверно найденные вектора движения являются причиной некорректного результата временной сегментации.

Очевидно, эти требования противоречивы. Основная сложность при вычислении  $k$  заключается в том, чтобы найти такое значение  $k$ , при котором разница между движением объектов и фона является достаточной для успешной сегментации и в то же время не является слишком большой, когда метод ОД перестает правильно работать. Поскольку характер движения может меняться во времени, временной шаг ОД должен вычисляться адаптивно. В разработанном методе вычисление временного шага производится итеративно по кадрам: значение  $k$  вычисляется для каждого кадра сразу после выполнения ОД, затем новое значение  $k$  используется в ОД при обработке следующего кадра. При вычислении  $k$  последовательно проверяются несколько условий понижения и повышения шага, после чего производится изменение шага в нужном направлении. Ниже приводится обоснование используемых условий. Общая схема алгоритма представлена в подразделе 2.1.

Логично предположить, что шаг  $k$  должен зависеть от некоего параметра  $p$ , который характеризует скорость движения в текущей видео сцене. Например,  $p$  можно вычислять как среднюю длину векторов движения всех блоков кадра. Однако такое решение имеет существенный недостаток. Среднее значение длины векторов может быть значительно меньше длины векторов движения какого-либо локального объекта в кадре. Поэтому увеличение временного шага может привести к тому, что метод ОД не сможет корректно находить вектора движения для этого объекта вследствие слишком большой разницы между его позициями на кадрах  $t$  и  $t-k$ .

В разработанном методе  $p$  вычисляется как  $N$ -я процентиль от длин векторов движения (т.е. такая длина, что  $N\%$  векторов движения блоков имеют меньшую длину). Таким образом, значение  $p$  характеризует скорость движения наиболее быстро двигающегося объекта, размер которого не меньше  $100-N\%$  площади кадра. Порог в  $100-N\%$  позволяет не учитывать неверно найденные вектора движения («шумные» вектора), которые могут быть слишком длинными. С учетом вышесказанного:

$$p = \max \{x : F(x) > T\}, \quad (1)$$

$$F(x) = \sum_{B: |v(B)| \geq x} 1, \quad x \in X, \quad (2)$$

где  $T$  – порог, используемый для борьбы с шумом ( $2\%$  от общего числа блоков);

$x$  – аргумент функции (длина вектора движения);

$X = \{0, \dots, v_{\max}\}$  – возможные значения длины вектора движения;

$v(B)$  – вектор движения блока  $B$ .

Значение  $p$  используется в условии уменьшения временного шага и учитывается следующим образом. Если  $p$  больше порога  $M$ , то значение временного шага вычисляется по формуле:

$$k_{i+1} = \frac{k_i}{p/M}, \quad (3)$$

где  $k_i$  и  $k_{i+1}$  – старое (для текущего кадра) и новое (для следующего кадра) значения временного шага соответственно; при этом  $k_0 = 2$ .

Значение порога  $M$ , выбирается в зависимости от возможностей используемого метода ОД. Значение этого порога следует сделать равным максимальному смещению между кадрами (выраженному в пикселях), при котором используемый метод ОД работает с допустимой точностью.

Как показывает практика, использование только параметра  $p$  для определения временного шага недостаточно, поскольку может привести к чрезмерному росту  $k$  и резкому снижению точности метода оценки движения. Например, это происходит в случае, если после продолжительного медленного движения, некоторый объект в кадре начинает быстро двигаться. В этом случае к моменту начала быстрого движения временной шаг становится настолько большим, что метод ОД не находит верных векторов движения для этого объекта, при этом значение параметра  $p$  может не преодолеть порога  $M$  (поскольку вектора для объекта найдены *неверно* и могут быть небольшими). Для решения этой проблемы используется еще один критерий уменьшения  $k$ . После работы метода ОД вычисляется число  $b$  неверно найденных векторов движения. Если это число превышает некоторый порог  $S$ , то значение временного шага уменьшается на 1. Значение порога  $S$  рассчитывается в зависимости от общего числа блоков кадра. При этом найденный для блока  $B$  вектор движения  $v$  считается верно найденным, если:

$$SAD(B, v, t, t-1) < a \cdot \frac{\sum_{x \in W} SAD(B, x, t, t)}{4}, \quad (4)$$

$$W = \left\{ \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\}, \quad (5)$$

$$SAD(B, v, t_1, t_2) = \sum_{x \in B} |I(x, t_1) - I(x+v, t_2)|, \quad (6)$$

где  $a$  – масштабирующий коэффициент (в разработанном алгоритме равен 1.5);

$I(x, t)$  – яркость пикселя  $x$  на кадре  $t$ .

Значение  $SAD(B, v, t, t-1)$  называется ошибкой компенсации (или просто ошибкой) блока  $B$  с использованием вектора  $v$ . Правая часть формулы **Ошибка! Источник ссылки не найден.** является оценкой «допустимой» ошибки компенсации для соответствующего блока. Достоинство такой оценки заключается в том, что она является адаптивной и зависит от дисперсии блока. Чем выше дисперсия блока, тем больше будет значение оценки в правой части **Ошибка! Источник ссылки не найден.** Такое поведение оценки отражает характер зависимости ошибок компенсации блоков от их дисперсии: как известно, блоки с более высокой дисперсией имеют большие ошибки компенсации. Стоит заметить, что данный метод оценки «допустимой» ошибки компенсации для блоков может быть использован в других алгоритмах. Например, его можно использовать в алгоритмах ОД в качестве критерия правильности нахождения вектора для прекращения поиска.

Описанные критерии уменьшения шага предназначены для контроля точности оценки параметров движения отдельных объектов кадра. Точность информации о движении для всего кадра контролируется дополнительным условием. Временной шаг  $k$  уменьшается вдвое, если:

$$r_i \geq R_i + d, \quad r_i = \frac{E_i^1}{E_i^2}, \quad (7)$$

где  $E_i^1$  и  $E_i^2$  – сумма ошибок компенсации всех блоков кадра, вычисленная для методов ОД с шагом 1 и  $k$  соответственно;

$R_i$  – динамически вычисляемый порог;

$d$  – значение допуска (в разработанном алгоритме  $d$  равно 0.4)

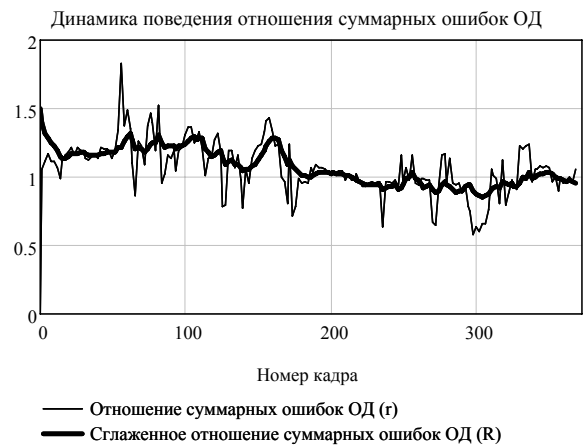
Обновление значения порога  $R_i$  производится при условии, что  $r_i < R_i + d$ , по формуле:

$$R_{i+1} = (1-u) \cdot R_i + u \cdot r_i, \quad (8)$$

где коэффициент  $u$  определяет скорость обновления порога  $R_i$ .

Данное условие позволяет контролировать относительную точность векторных полей, полученных методами ОД с разными значениями  $k$ . При этом коэффициент  $u$  определяет допустимую скорость изменения отношения суммарных ошибок  $E_i^1$  и  $E_i^2$ . Достоинство такого критерия уменьшения шага заключается в том, что он является «самообучающимся» и адаптируется к условиям конкретной видео-сцены. Адаптация заключается в использовании отношения суммарных ошибок блоков. Использование этого отношения позволяет контролировать точность ОД с шагом  $k$ , основываясь на точности векторного поля, полученного ОД с шагом 1. Пример поведения  $r_i$  и  $R_i$  проиллюстрирован на рис. 1.

Стоит заметить, что проверка условия (7) возможна при выполнении двух ОД (с временным шагом 1 и  $k$ ) при обработке каждого кадра. Поскольку это может привести к недопустимой вычислительной сложности всего алгоритма сегментации, этап проверки данного условия можно опустить. При этом качество работы метода не сильно ухудшится за счет использования оставшихся условий.



**Рисунок 1:** Динамика изменения  $r_i$  и  $R_i$  на примере видеопоследовательности «mobile calendar»

Увеличение временного шага  $k$  производится, если ни одно из условий уменьшения шага не выполнено и если  $F(M) < T$ , где  $F$  – функция, определенная в **Ошибка! Источник ссылки не найден.**, а  $M$  и  $T$  – пороги, описанные выше. Это условие выполняется в случае отсутствия объектов, скорость движения которых выше порога  $T$ , определяющего максимальную скорость движения, при которой верно работает ОД.

## 2.1 Общая схема алгоритма

Общий алгоритм получения информации о движении согласно вышеописанным идеям запишется следующим образом. Для каждого кадра перед выполнением сегментации выполняется два этапа ОД с временными шагами 1 и  $k$ . В качестве  $k$  выбирается значение  $k_{i+1}$ , вычисленное при обработке предыдущего кадра. Затем выполняется вычисление  $k_{i+1}$  для текущего кадра (вычисленное значение будет использоваться при обработке следующего кадра). Алгоритм вычисления  $k_{i+1}$  для текущего кадра следующий:

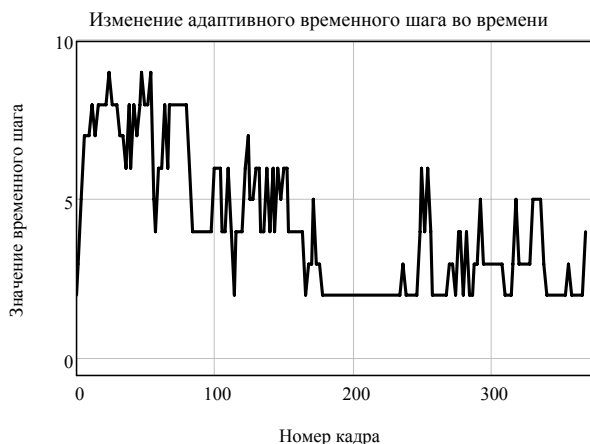
1. если выполнено условие (7), то шаг уменьшается вдвое ( $k_{i+1} = k_i / 2$ ). Переход на шаг 5;
2. если выполнено условие  $p > M$ , то вычисление шага производится по формуле (3). Переход на шаг 5;
3. если выполнено условие  $b > S$ , то значение  $k$  уменьшается на единицу ( $k_{i+1} = k_i - 1$ ). Переход на шаг 5;
4. если выполнено условие  $F(M) < T$ , то значение  $k$  увеличивается на 1;

5. переход к обработке следующего кадра.

### 3. РЕЗУЛЬТАТЫ

Разработанный метод был использован в алгоритме временной сегментации и доказал свою эффективность на практике. На рисунке 2 представлен график изменения временного шага  $k$  для стандартной видеопоследовательности «mobile calendar». Динамика изменения значений  $k$  полностью соответствует изменению характера движения во времени. Начало последовательности характеризуется медленным движением объектов, это обуславливает большие значения  $k$ . Уменьшение значений  $k$  в районе 200 кадра соответствует ускорению движения. В завершающей части видео последовательности, календарь и паровоз движутся немного быстрее, чем в начале.

Результаты временной сегментации до и после использования разработанного метода продемонстрированы на рис. 4, 5. Очевидно, что разработанный метод сыграл решающую роль в получении адекватных результатов сегментации.



**Рисунок 2:** Динамика изменения временного шага  $k$  во времени на примере видеопоследовательности «mobile calendar»

### 4. ЗАКЛЮЧЕНИЕ

Разработан и реализован метод повышения эффективности временной сегментации на сценах с медленным движением. Эффективность разработанного алгоритма подтверждена практикой, о чем свидетельствуют результаты, представленные на графике (рис. 2) и рисунках 4, 5. Дополнительным достоинством метода является его простота и, как следствие, высокая вычислительная эффективность. Разработанный метод позволяет увеличить эффективность любого алгоритма сегментации видео, использующего информацию о движении.

### 5. ЛИТЕРАТУРА

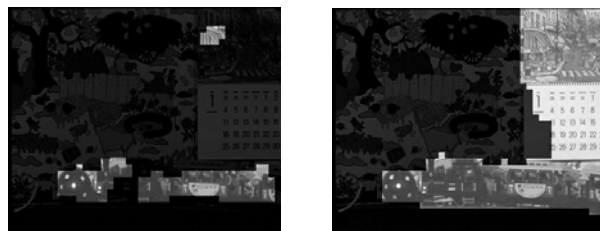
[1] D. Comaniciu, P. Meer, "Robust analysis of feature space: color image segmentation", *Proc. of CVPR'97 (Computer Society Conference on Computer Vision and Pattern Recognition)*, June 1997.

[2] Ju Guo, et al., "New Video Object Segmentation Technique with Color/Motion Information and Boundary Postprocessing", *Applied Intelligence journal*, March 1 1999.

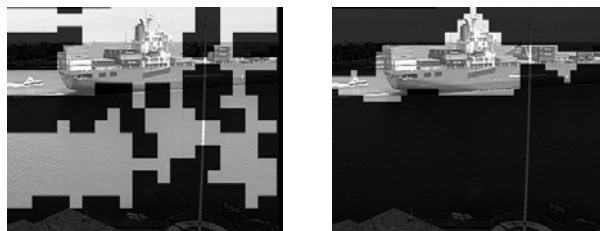
[3] С. Солдатов, К. Стрельников, Д. Ватолин, "Быстрое и надежное определение глобального движения в видеопоследовательностях", *Труды конф. Graphicon*, стр. 430-437, Июль 2006.

[4] Tsaig Y., Averbuch A., "Automatic Segmentation of Moving Objects in Video Sequences: a Region Labeling Approach", *Proc. of IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12(7), pp. 597-612, Jul. 2002.

[5] Ju Guo and J. Kim and C. Kuo, "New Video Object Segmentation Technique with Color/Motion Information and Boundary Postprocessing", *Applied Intelligence Journal*, 1999.



**Рисунок 3:** Результат временной сегментации последовательности «mobile calendar» до (слева) и после (справа) применения разработанного метода



**Рисунок 4:** Результат временной сегментации последовательности «container» до (слева) и после (справа) применения разработанного метода

### Об авторах

Ватолин Дмитрий Сергеевич — специалист в области сжатия изображений и видео с двенадцатилетним стажем. К.ф.-м.н., IEEE Member, старший научный сотрудник лаборатории компьютерной графики ВМиК МГУ, руководитель ВидеоГруппы. Автор книг "Методы сжатия изображений" (Д. Ватолин) и "Методы сжатия данных" (Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин). Основатель серверов «Все о сжатии» <http://compression.ru/> и «Compression Links» <http://www.compression-links.info/>.

E-mail: [dmitriv@graphics.cs.msu.ru](mailto:dmitriv@graphics.cs.msu.ru)

Гришин Сергей Викторович – аспирант, сотрудник лаборатории компьютерной графики ВМиК МГУ.

E-mail: [sgrishin@graphics.cs.msu.ru](mailto:sgrishin@graphics.cs.msu.ru)

Адрес: Москва, 119899, Воробьевы горы, МГУ, 2-й учебный корпус, факультет ВМиК, лаборатория компьютерной графики.