

# Пространственно-временное подавление шума в ВИДЕО ПОСЛЕДОВАТЕЛЬНОСТЯХ

Путилин С. Ю.

МГУ им. Ломоносова, факультет Вычислительной Математики и Кибернетики

Шумоподавление в видео актуально для многих задач, например для восстановления старых фильмов, телевизионных систем, телеконференций, систем видео наблюдения и кодирования видео.

Целью нашего проекта было создание быстрого алгоритма шумоподавления для обработки видеопоследовательностей перед кодированием. Для отладки алгоритма был создан фильтр для программы VirtualDub (<http://www.virtualdub.org/>), который свободно доступен на сайте, посвященном сжатию видео (<http://www.compression.ru/video/denoising/index.html>).

Разработанный алгоритм удаляет шум сначала во временной, а затем в пространственной области. Такой подход позволяет получить результат лучше, чем при удалении шума только в пространственной области или только во временной области.

Быстрые алгоритмы временного шумоподавления в большинстве своем осуществляют один из нижеследующих вариантов смешения кадров во временной области:

$$tdf(i, j, k) = \frac{tdf(i, j, k-1) + f(i, j, k) + (i, j, k+1)}{3},$$

$$tdf(i, j, k) = Median(tdf(i, j, k-1), f(i, j, k), (i, j, k+1)),$$

$$tdf(i, j, k) = k * f(i, j, k) + (1-k) * tdf(i, j, k-1),$$

$$tdf(i, j, k) = \frac{1}{N} \sum_{k=0}^{N-1} f(i, j, k),$$

где  $tdf(i, j, k)$  – пиксель кадра номер  $k$  в позиции  $i, j$  после шумоподавления,  $f(i, j, k)$  – пиксель кадра номер  $k$  в позиции  $i, j$  до шумоподавления,  $Median$  – медиана (функция, которая выбирает среднее значение из трех предоставленных),  $k$  – коэффициент шумоподавления в рекурсивном методе,  $N$  – количество кадров без движения.

Однако при просто смешивании кадров во временной области, без адаптации к движению, могут возникнуть артефакты. Они представляют из себя ореол границ быстро движущегося объекта. Пример можно видеть на Рисунке 1.



**Рисунок 1. В центре оригинальное изображение, слева после временного шумоподавления, не учитывающего движение, справа после нашего варианта временного шумоподавления (увеличенный фрагмент кадра)**

Наш алгоритм временного шумоподавления адаптируется к наличию движения в отдельных областях кадра и не производит в них смещения, чтобы избежать появления вышеописанных артефактов. Для этого вводится параметр фильтра  $T$  силы временного шумоподавления и формула выглядит следующим образом:

$$tdf(i, j, k) = \begin{cases} f(i, j, k) & \text{if } MDF > T \\ 0,5 \times \left(1 + \left(\frac{MDF}{T}\right)^\gamma\right) \times f(i, j, k) + 0,5 \times \left(1 - \left(\frac{MDF}{T}\right)^\gamma\right) \times tdf(i, j, k-1), & \text{иначе} \end{cases}$$

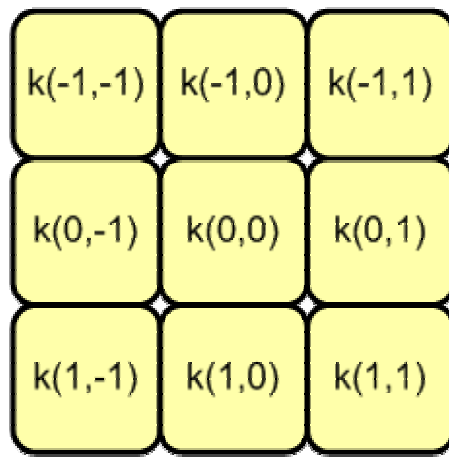
где  $tdf(i, j, k)$  – пиксель  $k$ -го кадра видео последовательности после шумоподавления во времени,  $f(i, j, k)$  – пиксель  $k$ -го кадра входной видео последовательности,  $MDF$  – функция, определяющая движение в локальной области пикселя  $f(i, j, k)$ ,  $\gamma$  – экспериментально подобранная константа,  $T$  – параметр фильтра.

Простые и быстрые алгоритмы пространственного шумоподавления используют смещение обрабатываемого пикселя с его соседями. При этом основное их отличие друг от друга в том, какие пиксели они считают соседями и с какими весами их берут. Одним из самых распространенных вариантов пространственного шумоподавления является Гауссово размытие. Оно осуществляется со следующими коэффициентами:

$$k(i, j) = \alpha \times \exp\left(-\frac{(i^2 + j^2)}{2\sigma^2}\right),$$

$$\alpha = 1 / \sum_{\text{all } i, j} k(i, j),$$

где  $i, j$  – координаты соседа (см. Рисунок 2 для пояснений),  $\sigma$  – коэффициент, определяющий степень размытия.



**Рисунок 2. Текущий обрабатываемый пиксель имеет координаты (0,0)**

Однако у этого алгоритма тоже есть свои недостатки. Он размывает мелкие детали, как показано на Рисунке 3.



**Рисунок 3. В центре оригинальное изображение, слева после пространственного шумоподавления, не учитывающего количество деталей, справа после нашего варианта пространственного шумоподавления (увеличенный фрагмент кадра)**

Наш алгоритм пространственного шумоподавления адаптируется к наличию деталей в локальной окрестности обрабатываемого пикселя  $f(i,j,k)$ . Кроме того введен параметр,  $P$  который позволяет контролировать его интенсивность. Сам алгоритм представляет из себя Гауссово размытие со следующими коэффициентами:

$$k(i, j) = wk \times c_{i,j} \times \exp\left(-\frac{\alpha \times i^a + \beta \times j^b}{P} \times LDQ\right),$$

$$wk = \frac{1}{\sum_{\text{all } i,j} c_{i,j} \times \exp\left(-\frac{\alpha \times i^a + \beta \times j^b}{P} \times LDQ\right)},$$

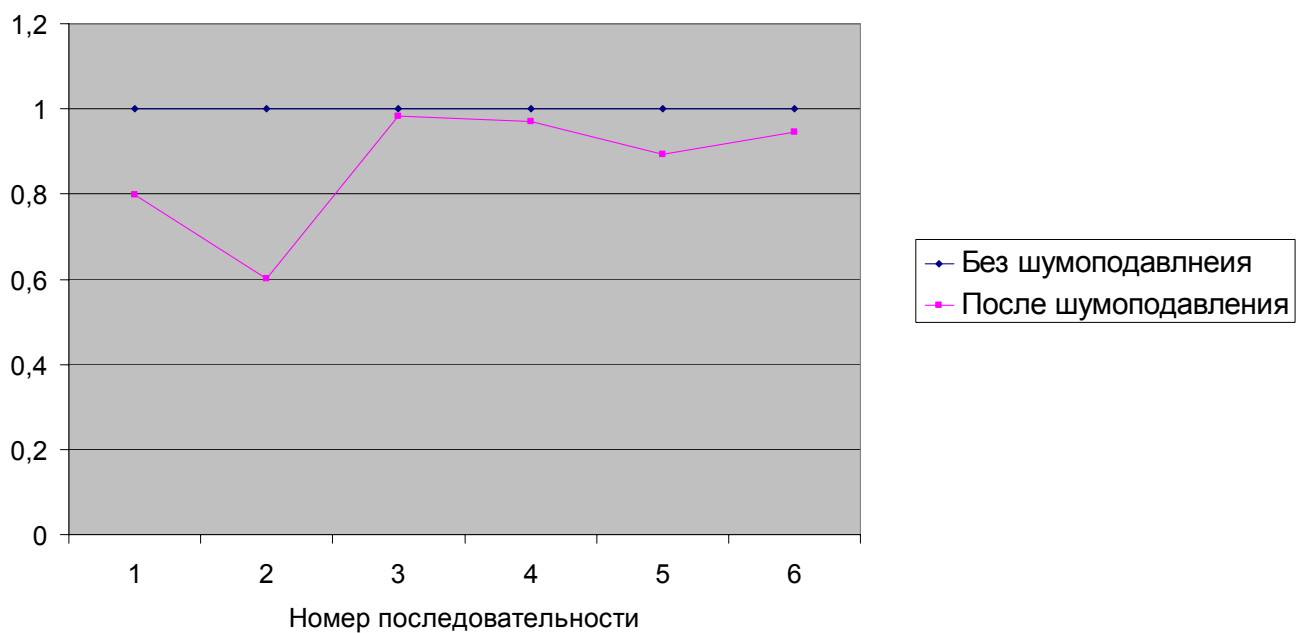
где  $k(i,j)$  – коэффициент Гауссова размытия (текущий обрабатываемый пиксель имеет координаты  $(0,0)$ ),  $c_{i,j}, a, b, c, \alpha, \beta, \gamma$  – экспериментально подобранные константы,  $P$  – параметр фильтра.

Данный фильтр применялся для улучшения степени сжатия кодека без потерь. Режим сжатия, в котором включался в работу фильтр шумоподавления, называется «визуально без потерь». Это означает, что пользователь не должен заметить каких-либо изменений после сжатия на глаз. Поэтому в были подобраны параметры шумоподавления, позволявшие добиться незматного подавления шума с обязательным сохранением всех деталей и предотвращавшие появление артефактов. Результат работы можно увидеть на Рисунке 4. Улучшение степени сжатия проиллюстрировано Таблицей 1.



**Рисунок 4. Слева оригинальное изображение, справа после пространственного шумоподавления (увеличенный фрагмент кадра)**

#### Улучшение сжатия



**Таблица 1. Сравнение степени сжатия в цветовом пространстве RGB (степень сжатия без шумоподавления взята за единицу)**

Данный алгоритм был разработан при участии Ватолина Д.С.

**Литература:**

- [1] G. de Haan, T.G. Kwaaitaal-Spassova, and O.A.Ojo, "Automatic 2-D and 3-D noise filtering for high-quality television", *Proc. of the 7th International Workshop on HDTV, Turin, Italy, 1994, Session 4B, paper No. 2.*
- [2] V. Zlokolica, A. Pizurica and W. Philips, "Video denoising using multiple class averaging with Multiresolution", *International Workshop VLBV03, September 2003, Madrid, Spain - Lecture Notes in Computer Science of Springer Verlag*
- [3] V. Zlokolica, A. Pizurica and W. Philips, "Recursive temporal denoising and motion estimation of video", *International Conference on Image Processing, October 2004, Singapore*