

Модификация метода нелокального усреднения для подавления шума в видео

Сергей Путилин, Алексей Лукин
факультет Вычислительной Математики и Кибернетики,
МГУ им. М.В. Ломоносова, Москва, Россия

Аннотация

В статье рассматривается адаптация популярного алгоритма нелокального усреднения к задаче шумоподавления в видеосигнале. Изучается обобщение области поиска блоков на несколько кадров видео, использование компенсации движения и предобработка кадров путем подавления коэффициентов ДПФ.

Ключевые слова: нелокальное усреднение, NLM, шумоподавление в видео.

1. ВВЕДЕНИЕ

Сравнительно недавно появился новый эффективный алгоритм шумоподавления для изображений, основанный на нелокальном усреднении блоков (non-local means, далее – NLM) [1]. Традиционные алгоритмы шумоподавления усредняют пиксели в пределах локального окна [2]. Пиксели с близкими значениями участвуют в усреднении с большими весами. Алгоритм NLM отличается тем, что оценивается не близость значений отдельных пикселей, а похожесть структуры (содержимого) их двумерных окрестностей. Благодаря этой особенности алгоритм NLM способен хорошо сохранять текстуры и подавлять на них шум.

Вычислительная сложность оригинального алгоритма NLM [1] очень высока, и поэтому была предложена оптимизация, позволяющая пересчитывать веса усреднения блоков не в каждом пикселе, а на разреженной решетке [3]. Это позволило реализовать метод NLM для обработки видео.

2. АДАПТАЦИЯ МЕТОДА NLM К ВИДЕО

2.1 Оригинальный алгоритм NLM

Обозначим $x[i, j, t]$ – пиксель в позиции $[i, j]$ в кадре номер t , A – квадрантная область с центром в точке $[0, 0]$, в дальнейшем именуемая окном, по которой собственно и будет идти усреднение. Тогда результатом работы оригинального NLM будет кадр с пикселями $x'[i, j, t]$:

$$x'[i, j, t] = \frac{1}{s} \sum_{[k, m] \in A} w[k, m, t] x[i+k, j+m, t],$$

где s – сумма коэффициентов w , а коэффициенты $w[h, v, t]$ вычисляются по формуле

$$w[k, m, t] = \exp\left(-\frac{SSD(B[i, j, t], B[i+k, j+m, t])}{h^2}\right),$$

где $B[i, j, t]$ – блок с центром в пикселе $x[i, j, t]$, SSD – сумма квадратов разностей значений соответствующих пикселей двух блоков.

2.2 Поблочный NLM

Алгоритм NLM требует достаточно много вычислений. Поэтому для его ускорения была предложена следующая техника. Рассмотрим блок $P[i, j, t]$ размера $m \times m$ с центром в пикселе $x[i, j, t]$. Будем применять обычный NLM, но с одинаковыми коэффициентами $w[h, v, t]$ для всех пикселей блока $P[i, j, t]$. Рассчитаем их по формуле:

$$w[h, v, t] = \exp\left(-\frac{SSD(B[i, j, t], B[i+k, j+m, t])}{h^2}\right),$$

где $[k, m] \in A$. Тем самым можно отметить, что полученный алгоритм оперирует вместо отдельных пикселей блоками размера $m \times m$.

Заметим, что в общем случае m может быть равно 1. При этом получится оригинальный алгоритм NLM. Обозначим за n сторону блоков, используемых для вычисления SSD. Не обязательно $m < n$ строго, случай равенства этих величин также рассматривался авторами и при этом алгоритм давал приемлемые результаты.

2.3 NLM в нескольких кадрах

В отличие от изображений, для которых изначально был придуман NLM, в видео есть возможность нахождения похожих областей в соседних кадрах. Причем в большинстве случаев именно в соседних кадрах и будут найдены блоки с наилучшим соответствием. Поэтому предлагается расширить область поиска подобных блоков на один или несколько соседних кадров:

$$x'(i, j, t) = \frac{1}{s} \sum_{l \in [-T, T]} \sum_{[k, m] \in A} w[k, m, t] x[i+k, j+m, t+l],$$

где

$$w[h, v, t] = \exp\left(-\frac{SSD(B[i, j, t], B[i+k, j+m, t+l])}{h^2}\right),$$

Сравнение показало преимущество алгоритма, использующего несколько кадров перед алгоритмом, который работает по одному кадру. Уровень шумоподавления в обоих случаях примерно одинаков, если учитывать, что в случае одного кадра для адекватного сравнения (при равной вычислительной сложности) необходимо брать окно поиска блоков большего размера. Однако в случае алгоритма, работающего по одному кадру, остается неприятный эффект временного шума

– однородные области кадра имеют постоянно меняют оттенки, что приводит к мельканию и снижает соотношение сигнал/шум. В случае же алгоритма, использующего несколько кадров, данный эффект пропадает.

Дальнейшим улучшением данного алгоритма служит использование информации о движении:

$$x'(i, j, t) = \frac{1}{S} \sum_{l \in [-T, T]} \sum_{[k, m] \in A} w[k, m, t] x[i + k + mx(l), j + m + my(l), t + l],$$

где $mx(l)$, $my(l)$ – горизонтальная и вертикальная составляющие вектора движения (сдвига объекта или фона видео) между кадрами t и $t+l$,

$$w[h, v, t] = \exp\left(-\frac{SSD(B[i, j, t], B[i + k + mx(l), j + m + my(l), t + l])}{h^2}\right).$$

Данное улучшение позволяет применять области поиска похожих блоков меньшего размера даже при достаточно быстром панорамировании в видеопоследовательности и получать при этом хороший результат.

Вектора движения можно найти при помощи алгоритмов нахождения движения в видео. Они широко распространены, в качестве примеров можно рассмотреть [4]-[6].

3. ПРЕДОБРАБОТКА С ПОМОЩЬЮ ДПФ

Многие хорошо известные методы шумоподавления основываются на подавлении коэффициентов преобразований изображения. В качестве преобразований часто используются дискретное вейвлет-преобразование (ДВП) или дискретное преобразование Фурье (ДПФ), а также его модификация – дискретное косинусное преобразование (ДКП).

Недостатком стандартного (сепарабельного) ДВП является плохая локализация наклонных и диагональных границ. Сильная сторона ДВП – подавление шумов различного масштаба.

ДПФ одинаково хорошо локализует границы всех направлений, но страдает из-за эффекта Гиббса, порождаемого большим числом осциллирующих базисных функций [3].

Мы предлагаем использовать метод подавления коэффициентов ДПФ совместно с методом NLM для повышения качества шумоподавления.

3.1 Метод DFTT

Опишем используемый нами метод DFTT (Discrete Fourier Transform Thresholding) подавления коэффициентов ДПФ. Изображение разбивается на блоки размером 16×16 пикселей. Для повышения качества и устранения «эффекта блочности» блоки могут выбираться с перекрытием 50% и взвешиваться окном типа Kaiser-Bessel-derived, используемом как при деконпозиции, так и при синтезе. После взвешивания блока изображения он преобразуется с помощью ДПФ. К амплитудному спектру $X[k, m]$ (за исключением нулевого коэффициента) применяются множители $G[k, m]$ для уменьшения амплитуд коэффициентов, близких к шумовому порогу $T[k, m]$:

$$G = \begin{cases} 0.5(X/T)^4, & X < T \\ 1 - 0.5\left(\frac{2T - X}{T}\right)^4, & T \leq X < 2T \\ 1, & X \geq 2T \end{cases}$$

Здесь $X = X[k, m]^2$ и $T = T[k, m]^2$ – мощность сигнала и шума на данной частоте. Для белого шума мощность на всех частотах может полагаться одинаковой, для окрашенных шумов (например, зерна фотопленки) могут использоваться алгоритмы получения спектра шума по образцам.

После умножения амплитудного спектра на множители $G[k, m]$ производится обратное ДПФ, и блоки, повторно взвешенные весовым окном, складываются в результирующее изображение.

3.2 Объединение с методом NLM

С целью повышения качества метода NLM было опробовано 2 варианта соединения NLM и DFTT. В первом варианте метод NLM (многокадровая версия) используется как предобработка перед DFTT. А в методе DFTT производится анализ, как зашумленных блоков, так и обработанных NLM. На основании спектра *обработанных* блоков вычисляются амплитудные множители $G[k, m]$, а взвешиваются ими и участвуют в синтезе – спектральные коэффициенты *зашумленных* блоков. Этот подход напоминает высококачественный алгоритм трехмерной фильтрации блоков [7], но намного проще и применим для обработки видео.

В результате тестирования такого объединения NLM и DFTT было получено незначительное улучшение соотношения сигнал/шум (рис. 1–3) по сравнению с обычным многокадровым методом NLM, однако визуально результат отличался сильно: присутствовал эффект Гиббса, но ровные области обрабатывались достаточно качественно, и результат содержал достаточное количество средне-низких частот [8].

Второй вариант объединения NLM и DFTT осуществляет предобработку методом DFTT всех кадров, подаваемых на многокадровый алгоритм NLM. В результате тестирования этой модификации было замечено снижение количества артефактов по сравнению с отдельным применением методов NLM и DFTT, а также заметное улучшение отношения сигнал/шум (Рис. 1 – Рис. 3).

4. РЕЗУЛЬТАТЫ

Для обоснования использования предложенных идей было проведено объективное тестирование. В не зашумленное видео был добавлен искусственно сгенерированный шум, после чего шум был удален различными вариантами алгоритмов. Далее между обработанным видео и оригиналом было посчитано соотношение сигнал/шум (PSNR) в цветовом пространстве RGB. Результаты для некоторых стандартных видеопоследовательностей показаны на Рис. 1 – Рис. 3. По вертикали отложено улучшение PSNR по сравнению со значением метрики между оригиналом и зашумленной последовательностью. Разные столбцы соответствуют различным вариантам алгоритма. Для каждой версии алгоритма вручную подбирались оптимальное (с точки зрения PSNR) значение силы шумоподавления и пространственный размер окна поиска похожих блоков.

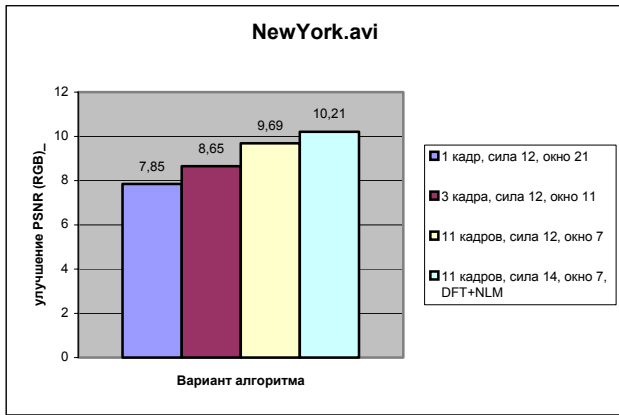


Рис. 1. Результаты замера на NewYork.avi

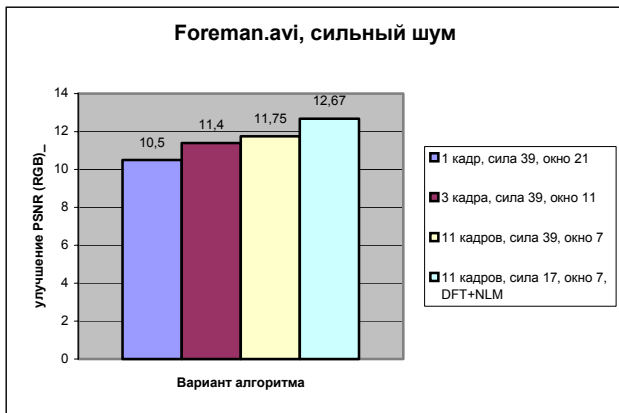


Рис. 2. Результаты замера на Foreman.avi с сильным шумом

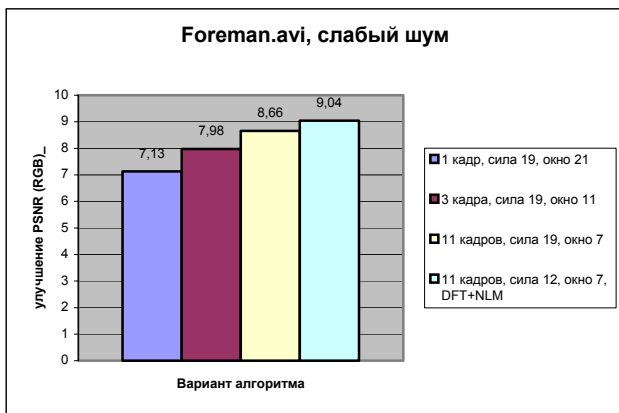


Рис. 3. Результаты замера на Foreman.avi со слабым шумом

Как видно по графикам, наилучшее объективное качество демонстрирует алгоритм DFTT+NLM в многокадровом режиме. Но даже без предобработки DFTT многокадровый режим алгоритма NLM увеличивает значение PSNR.

Также было проведено визуальное сравнение, которое тоже показало обоснованность применения предложенных идей по улучшению качества алгоритма NLM. Изображения с результатами обработки можно найти на веб-страничке [8].

5. БИБЛИОГРАФИЯ

- [1] A. Buades, J.M. Morel "A Non-Local Algorithm for Image Denoising" // *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, vol. 2, 20-26 June 2005, pp. 60-65.
- [2] Carlo Tomasi and Roberto Manduchi, "Bilateral Filtering for Gray and Color Images" // *ICCV 1998*, pp. 839-846.
- [3] Alexey Lukin "A Multiresolution Approach for Improving Quality of Image Denoising Algorithms" // *IEEE International Conference On ACOUSTICS, SPEECH, AND SIGNAL PROCESSING (ICASSP-2006)*. Toulouse, France, 2006.
- [4] 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, No. 3, March 2006.
- [5] 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 on circuits and systems for video technology*, vol. 15, No. 11, November 2005.
- [6] Путилин С.Ю. «Быстрый алгоритм нахождения движения в видеопоследовательностях» // *Труды конференции Graphicon-2006*, стр. 407-410, Новосибирск, Академгородок, Россия, июль 2006.
- [7] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian "Image denoising with block-matching and 3D filtering" // *Electronic Imaging'06, Proc. SPIE 6064*, no. 6064A-30, San Jose, California USA, January 2006.
- [8] Демонстрационная веб-страничка: <http://audio.rightmark.org/lukin/graphics/denoisingvideo.htm>

Об авторах

Сергей Путилин — аспирант лаборатории компьютерной графики и мультимедиа при факультете ВМиК МГУ. E-mail: sputilin@graphics.cs.msu.ru

Алексей Лукин (к.ф.-м.н.) — м.н.с. кафедры мат. физики факультета ВМиК МГУ. E-mail: lukin@graphics.cs.msu.ru

Modifications of a Non-Local Means Denoising for Video

Abstract

In this paper, we propose the adaptation of a popular Non-Local Means image denoising method for video processing. We study extension of blocks search area to multiple frames, use of motion compensation, and pre-processing of frames by DFT thresholding.