

УДК 621.397

DOI: 10.15587/2706-5448.2021.225257

КОНВЕЙЕРИЗОВАННАЯ РЕАЛИЗАЦИЯ ASWM ФИЛЬТРА ИЗОБРАЖЕНИЙ НА ПЛИС

Васильченков О. Г., Либерг И. Г., Можяев М. А., Сальников Д. В.

Объектом исследования является алгоритм адаптивного переключающегося взвешенного медианного фильтра изображений (ASWM). Данный алгоритм является одним из наиболее эффективных в области подавления импульсных шумов. Вычислительная сложность и алгоритмические особенности этого адаптивного нелинейного фильтра не позволяют реализовать фильтр, работающий в режиме реального времени на современных микросхемах ПЛИС.

Наиболее проблемными местами алгоритма являются цикл оценки весовых коэффициентов, который не имеет ограничения количества итераций и содержит большое количество операций деления. Это не позволяет реализовать фильтр на ПЛИС достаточно эффективным методом.

В ходе исследования использовалась программная модель фильтра на языке Python. Оценка качества работы алгоритма проводилась при помощи метрик Peak Signal-to-Noise Ratio (PSNR) и Structural Similarity Index Measure (SSIM).

Моделирование позволило выяснить эмпирически количество итераций цикла оценки весовых коэффициентов при различных уровнях плотности шума и оценить влияние искусственного ограничения максимального количества итераций на качество работы фильтра. В независимости от интенсивности зашумляющего воздействия алгоритм выполняет менее 40 итераций цикла оценки. Также было проведено моделирование работы алгоритма с различными вариантами реализации модуля деления. В работе рассмотрены основные из них и предложен наиболее оптимальный с точки зрения соотношения точность/аппаратные затраты на реализацию. Таким образом был предложен модифицированный алгоритм, не имеющий указанных недостатков.

Благодаря модификациям алгоритма обеспечена возможность реализовать конвейеризованный ASWM фильтр изображений на современных ПЛИС. Фильтр синтезирован для основных семейств ПЛИС компании Intel. Реализация, не уступающая по метрикам SSIM и PSNR оригинальному алгоритму, требует менее 65000 логических ячеек ПЛИС и позволяет осуществлять фильтрацию монохромных изображений разрешением FullHD со скоростью 48 кадров/с при тактовой частоте 100 МГц.

Ключевые слова: *адаптивный фильтр, нелинейный фильтр, медианный фильтр, импульсный шум, Peak Signal-to-Noise Ratio, Structural Similarity Index Measure.*

1. Введение

Процесс получения графических данных с различных датчиков и их передачи в подсистемы обработки изображений всегда сопровождается повреждением данных или даже их потерей.

Электромагнитный шум или ошибки передачи часто приводят к появлению в выходном изображении пикселей со случайными значениями. Такой тип шума называется импульсным шумом. Его можно увидеть на полутоновых изображениях как случайные белые и черные пиксели или группы пикселей, искажающие основную информацию изображения.

Медианный фильтр в настоящее время является одним из самых эффективных при фильтрации импульсного шума. Фильтр применяется для всех без исключения пикселей. Во время медианной фильтрации часть информации теряется, что приводит к сглаживанию границ и размытию [1].

Для устранения этого недостатка был предложен фильтр взвешенной медианы WM [2], который позволяет сохранить больше исходных деталей изображения благодаря весам для каждого пикселя внутри окна фильтрации.

Идеальный фильтр не должен изменять пиксели, неповрежденные шумом. В [3] рассмотрены популярные типы фильтров, которые были предложены для борьбы с потерей деталей изображения вследствие сглаживания. В частности, предложены:

- детекторы шума для классификации и обнаружения зашумленных пикселей [4, 5];
- гибридные фильтры [6, 7];
- адаптивные «переключающиеся» фильтры [8–10], в которых не применялась медианная фильтрация для пикселей без шума.

Наиболее эффективным представляется адаптивной переключающийся медианный фильтр (ASWM) [11]. Тем не менее, его алгоритм сложно реализовать на ПЛИС и современных процессорах, что затрудняет его использование в задачах реального времени и подтверждает актуальность данного исследования.

Таким образом, *объектом исследования* выбран алгоритм адаптивного переключающегося взвешенного медианного фильтра изображений (ASWM).

Цель исследования – поиск методов реализации ASWM, позволяющих реализовать конвейеризованный фильтр на большинстве доступных на рынке программируемых логических интегральных схем.

2. Методика проведения исследования

Фильтр изображений ASWM использует детектор шума, основанный на аппарате порядковых статистик.

Алгоритм работы представлен на рис. 1.

Шаг 1 – Инициализация. Веса $w_{k,l}$ инициализируются значениями 1.0.

В каждом окне изображения вычисляется средневзвешенное значение для пикселя с координатами (i, j) в соответствии с уравнением:

$$M_w(i, j) = \frac{\sum_{k,l} w_{k,l} X_{i+k, j+l}}{\sum_{k,l} w_{k,l}}, \quad (1)$$

где $w_{k,l}$ – весовые коэффициенты, в соответствии с уравнением (2); $X_{k,l}$ – элементы окна фильтра.



Рис. 1. Алгоритм работы ASWM фильтра

Шаг 2 – Оценка весов.

$$w_{k,l} = \frac{1}{|X_{i+k,j+l} - M_w(i,j)| + \delta}, \quad (2)$$

где $X_{k,l}$ – элементы окна фильтра; M_w – средневзвешенное значение; δ – заданное малое значение, для избегания деления на ноль.

Затем вычисляется новое значение $M_w(i,j)$ с новыми вычисленными весами.

Шаг 3 – Сохранение результата. Если $|M_w(i,j)^t - M_w(i,j)^{t-1}| < \varepsilon$, где ε – заданное пороговое значение, цикл завершается, в противном случае переходите к шагу 1.

Затем стандартное отклонение рассчитывается в соответствии с уравнением:

$$\delta_w(i, j) = \sqrt{\frac{\sum_{k,l} W_{k,l} (X_{i+k, j+l} - M_w(i, j))^2}{\sum_{k,l} W_{k,l}}}, \quad (3)$$

где $W_{k,l}$ – весовые коэффициенты, рассчитанные согласно уравнению (2); $X_{k,l}$ – элементы окна фильтра; M_w – средневзвешенное значение согласно уравнению (1).

Вывод о зашумленности пикселей делается на основании произведения среднеквадратичного отклонения и порогового значения.

Наконец, фильтр ASWM можно описать в виде уравнения:

$$Y_{i,j} = \begin{cases} m_{i,j}, & \text{если } |X_{i,j} - M_w(i, j)| > \alpha \times \delta_w(i, j) \\ X_{i,j}, & \text{в противном случае} \end{cases}, \quad (4)$$

где $m_{i,j}$ – медианное значение в текущем окне; $X_{i,j}$ – начальное значение пикселя; $Y_{i,j}$ – значение отфильтрованного пикселя; $M_w(i, j)$ – средневзвешенное значение окна; α – заранее определенный порог; $\delta_w(i, j)$ – стандартное отклонение.

Как видно из алгоритма, процесс нахождения коэффициентов является итерационным. Количество итераций не известно заранее. Зашумленность текущего пикселя определяется по значению среднеквадратичного отклонения и величине отклонения значения пикселя от взвешенного среднего. Это позволяет осуществлять фильтрацию с высокими показателями шумоподавления в широком диапазоне уровней шума.

Недостатки этого фильтра можно резюмировать следующим образом:

- большое количество операций деления;
- наличие цикла недетерминированной длины;
- чувствительность к уменьшению разрядности вычислений с фиксированной точкой;
- отсутствие возможности распараллелить алгоритм ввиду описанных выше недостатков.

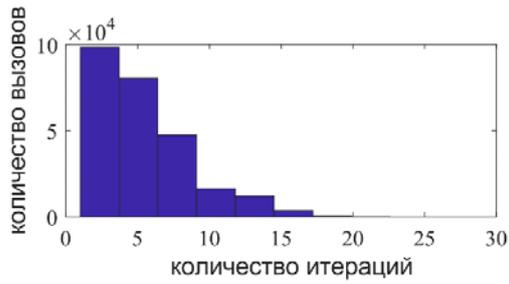
Данные недостатки не позволяют реализовать конвейерный алгоритм фильтрации ASWM на современных ПЛИС и векторных процессорах что делает практически невозможной задачу обработки изображений в реальном времени.

3. Результаты исследования и их обсуждение

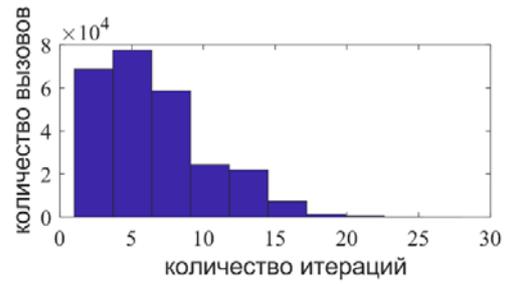
3.1. Разработка модифицированного ASWM фильтра

Как было указано выше, недостатком является сложность подпрограммы оценки весов. Основой программного модуля оценки весов является цикл с неопределенным количеством итераций. На основе построенной программной модели было проведено исследование работы фильтра при различном уровне зашумления импульсным шумом.

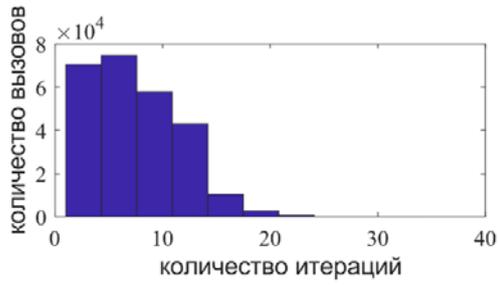
В результате были получены гистограммы распределения количества итераций цикла (рис. 2) при различной плотности шума. Как видно, подавляющее большинство вызовов выполняют менее 25 итераций, т. е. для реализации фильтра без значительных потерь производительности достаточно 25 итераций цикла.



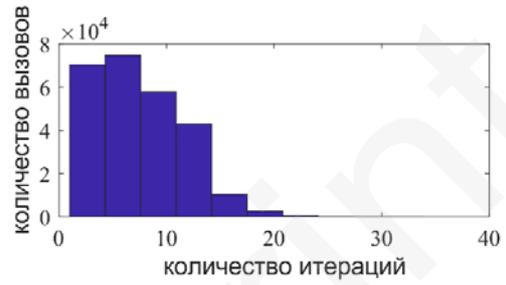
a



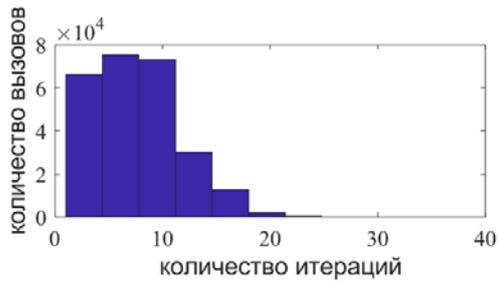
б



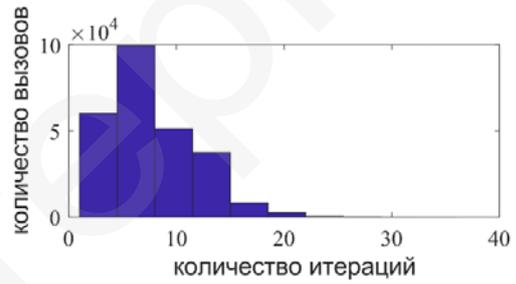
в



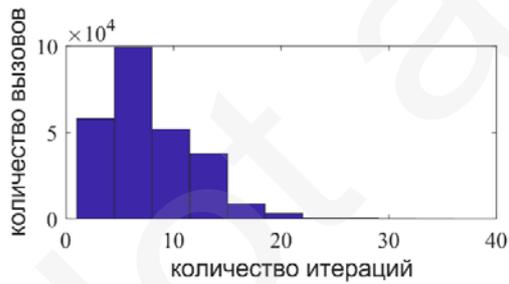
г



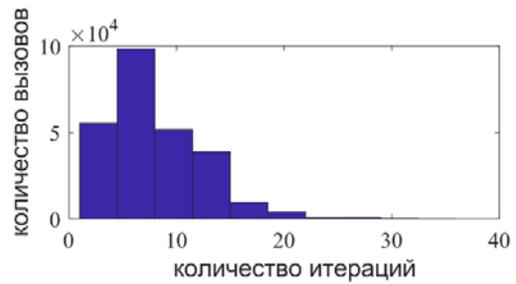
д



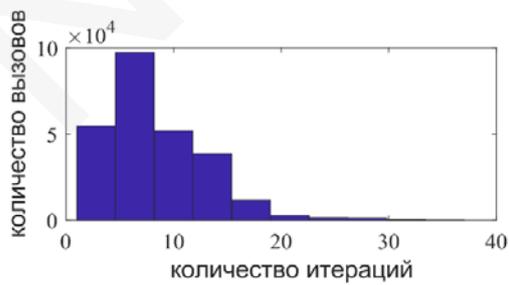
е



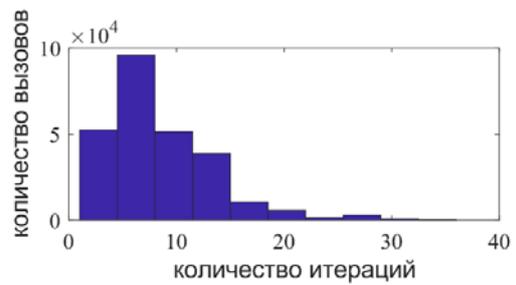
ё



ж



з



и

Рис. 2. Распределение количества итераций алгоритма при различных уровнях плотности шума: *a* – 5 %; *б* – 15 %; *в* – 20 %; *г* – 30 %; *д* – 35 %; *е* – 45 %; *ё* – 50 %; *ж* – 60 %; *з* – 65 %; *и* – 75 %

Поскольку целью работы является реализация наиболее быстродействующего варианта фильтра, было решено реализовать цепочку из N блоков фильтрации, которые можно «обойти» (при помощи сигнала байпасирования), чтобы имитировать выход из цикла при выполнении условия раннего выхода. В случае, если данное условие не достигнуто на протяжении всех блоков фильтрации, будут использованы веса последнего блока (рис. 3).

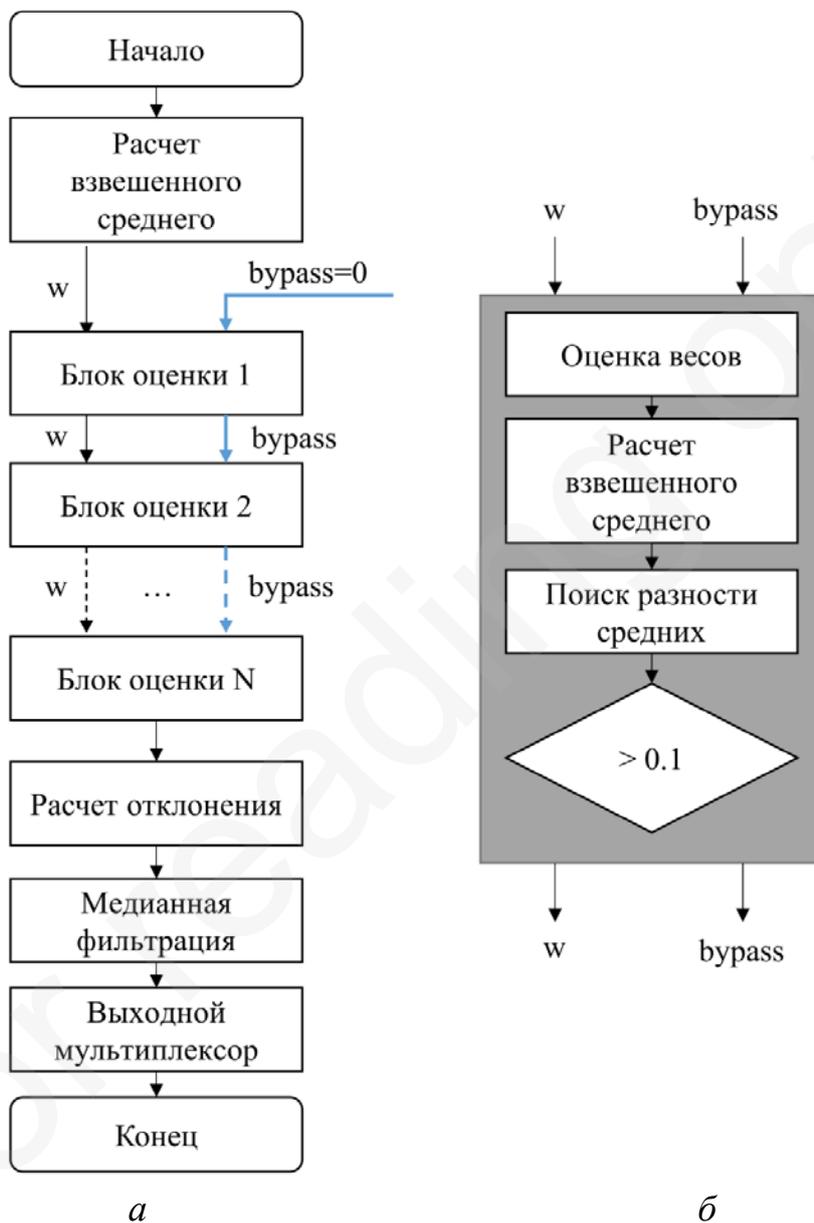


Рис. 3. Модифицированный алгоритм ASWM фильтрации: *a* – алгоритм фильтра; *б* – структура блока оценки весов

3.2. Оптимизация модуля оценки весов

Весь цикл, который оценивает веса для операций нахождения отклонения, может быть разбит на более мелкие подмодули. Блок оценки весов состоит из подблока вычисления весов и логики, которая состоит из операций сравнения и мультиплексора (рис. 4).

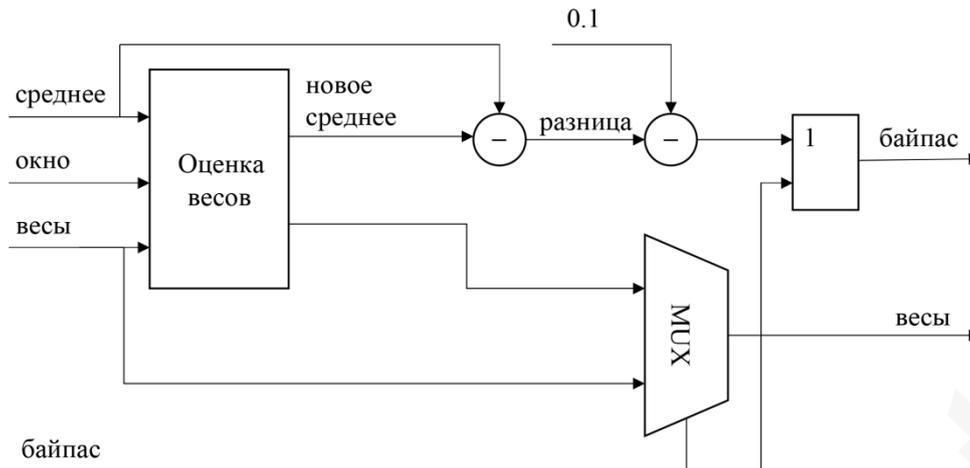


Рис. 4. Блок-схема модуля оценки весов

Такой подход позволяет конвейеризовать фильтр, ограничивая его длину (количество итераций цикла) без значительных потерь качества фильтрации. На рис. 5 и рис. 6 показаны метрики качества работы исходной модели ASWM (максимальное количество итераций – 100) и измененной описанным образом (количество блоков оценки (итераций) – 40, 20, 10 и 5).

Следует отметить, что значительная часть ресурсов, используемых для реализации всего фильтра, потребляется блоком оценки весов, который, как упоминалось выше, содержит девять 32-битных операций деления, каждая из которых потребляет более 1000 LUT (lookup table) ПЛИС.

Как видно из рис. 5 и рис. 6, показатели PSNR (peak signal-to-noise ratio) и SSIM (structural similarity index measure) [12] для фильтров с количеством блоков оценки весов от 100 до 20 практически не уменьшаются. Значительно снижается производительность фильтров с 5–10 единицами.

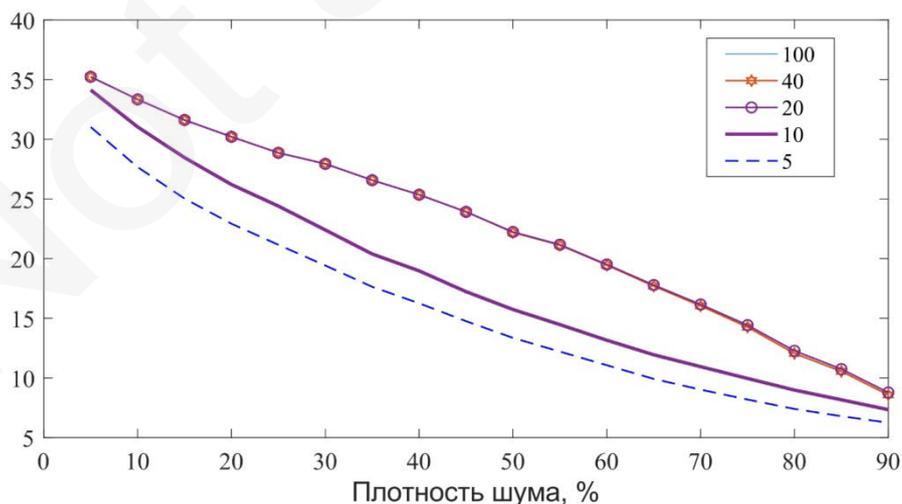


Рис. 5. PSNR фильтра с различным количеством блоков оценки весов

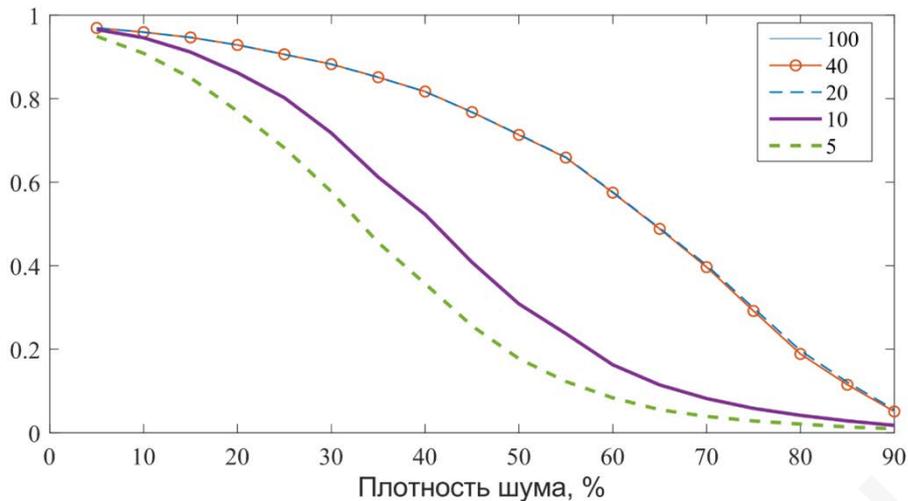


Рис. 6. SSIM фильтра с различным количеством блоков оценки весов

Современные производители ПЛИС предлагают решения [13] с общим количеством LUT достаточным для конвейерной реализации фильтра ASWM с 15 модулями оценки весов. Для обеспечения качества фильтрации необходимо 25.

В модуле оценки весов необходимо использовать 32-х разрядное деление. Реализация деления затратна на ПЛИС.

3.3. Оптимизация модуля деления

Как известно, разрядность числителя и знаменателя влияет на потребление аппаратных ресурсов для реализации блока деления. Исследования показали, что при разрядности деления менее 32 бит не удастся получить требуемые показатели качества фильтрации.

Таким образом, следующим этапом исследования является разработка эффективного модуля деления со сниженным потреблением ресурсов.

Ниже рассмотрены основные методы деления с меньшим потреблением ресурсов.

В результате проведенного моделирования фильтра было определено, что значительное влияние на качество оказывают делители больше значения числа 5. Это упрощает задачу аппроксимации из-за усечения участка с большой нелинейностью.

На программной модели фильтра было проведено исследование нескольких вариантов замены модуля деления.

Одним из вариантов является аппроксимация деления полиномом. Полином 5 степени позволяет реализовать функцию, которая требует 4 такта для получения результата (рис. 7).

$$y = -1.3724e^{-13}x^5 + 1.9381e^{-10}x^4 - 1.0196e^{-7}x^3 + 2.4567e^{-5}x^2 - 0.0027x + 0.1093. \quad (5)$$

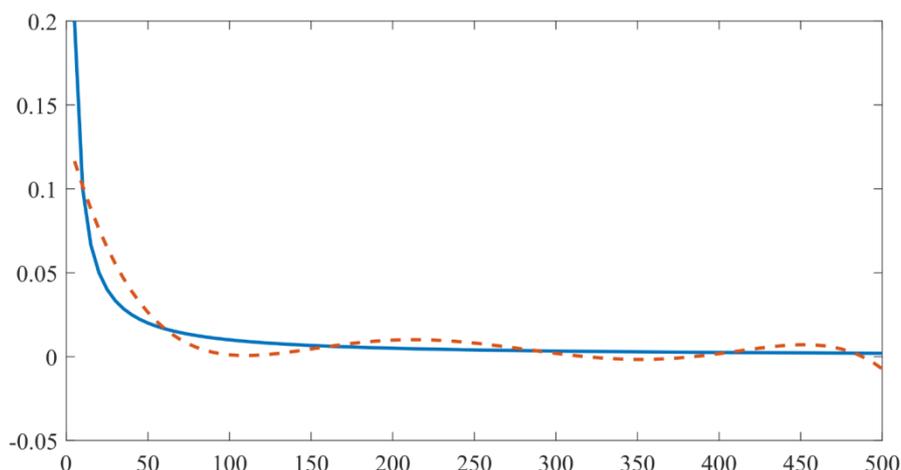


Рис. 7. Сравнение функции деления и аппроксимированной полиномом степени 5

Отклонение не превышает 10 %. Точность работы не удовлетворяет требованиям. Аппроксимация полиномом степени 10 (рис. 8).

$$y = 1.7852e^{-24}x^{10} - 4.7764e^{-21}x^9 + 5.5142e^{-18}x^8 - 3.5953e^{-15}x^7 + 1.4556e^{-12}x^6 - 3.7901e^{-10}x^5 + 6.3589e^{-8}x^4 - 6.7055e^{-6}x^3 + 0.0004x^2 - 0.0142x + 0.2147. \quad (6)$$

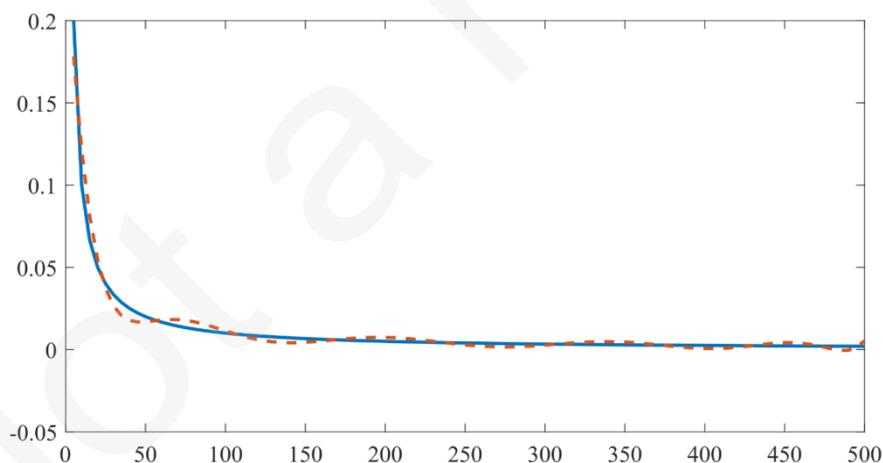


Рис. 8. Сравнение функции деления и аппроксимированной полиномом степени 10

Как видно, увеличение степени полинома существенно улучшает качество алгоритма. Отклонение не превышает 5 %.

Третий этап исследования был связан с применением табличного метода. Поскольку делимое является постоянным, а делитель представляет собой дробное число, не превышающее 255 по абсолютной величине, имеет смысл рассмотреть табличные методы.

Табличный метод с использованием линейной интерполяции для повышения точности может быть представлен следующими уравнениями:

$$tg\alpha = table[n] - table[n+1], \quad (7)$$

$$y = table[n] - tg\alpha \{x\}, \quad (8)$$

где $table[n]$ и $table[n+1]$ – табличные значения деления; $\{x\}$ – дробная часть x ; n – целая часть x .

Отклонение в данном случае не превышает 5 %.

Для данного метода необходимо хранить таблицы, чтобы загрузить текущее и следующее значение.

Дальнейшее моделирование показало, что дробной частью числа в данном случае можно пренебречь, сохранив при этом высокое качество фильтрации.

В соответствии с алгоритмом фильтра, значения X в цикле оценки весов не могут превышать 28. Однако нельзя применять 8-битное деление из-за того, что качество фильтрации значительно снижается.

Пренебрежение дробной частью делителя не вносит существенных изменений (падение показателей PSNR и SSIM в пределах 0,5 %) при сохранении разрядности результата.

Таким образом, наиболее приемлемый вариант – использовать таблицу результатов 32-битного целочисленного деления.

Такая архитектура может быть реализована на ПЛИС с использованием блока постоянного запоминающего устройства с адресной шиной 8 бит и шиной данных 32 бита. Затраты памяти на реализацию блока оценки составляют 8 Кбит, что позволяет реализовать до 85 блоков оценки в современных устройствах.

Ниже показано потребление аппаратных ресурсов для реализации блока оценки весов с модифицированным модулем FPGA.

Описанный подход позволяет реализовать деление без использования дополнительной логики, но с использованием памяти ПЛИС. Потребление памяти позволяет реализовать необходимое количество блоков в современных ПЛИС Altera и Xilinx. Описанное решение можно еще больше минимизировать путем дальнейшего усечения делителя.

3.4. Расчет потребления ресурсов и оценка производительности

Общее потребление ресурсов ПЛИС, необходимых для реализации фильтра, можно оценить как сумму компонентов:

$$V_{sum} = V_{mean} + V_{deviation} + V_{loop} \cdot N, \quad (9)$$

где V_{mean} – ресурсы ПЛИС для единицы взвешенного среднего; $V_{deviation}$ – ресурсы FPGA для блока отклонения; V_{loop} – ресурсы ПЛИС на единицу оценки; N – количество блоков оценки.

Общее количество LUT зависит от количества единиц оценки в фильтре. В табл. 1 показано количество LUT, необходимых для реализации фильтра с N единиц блоков оценки для ПЛИС Intel.

Таблица 1

Потребление ресурсов ПЛИС

Ресурсы	N	Cyclone V	Cyclone IV E	Cyclone 10
Total logic elements	10	5071	29792	29792
	15	5071	38377	38377
	20	5071	46962	46962
	25	5071	55547	55547
	30	5071	64132	64132
Total registers	10	47054	18368	18368
	15	68614	25733	25733
	20	90174	33098	33098
	25	111734	40463	40463
	30	133294	47828	47828
Total memory bits	10	776960	788193	788193
	15	1165440	1182138	1182138
	20	1553920	1576083	1576083
	25	1942400	1970028	1970028
	30	2330880	2363973	2363973
Multiplier 9-bit	–	150	178	66

Длина конвейера фильтра также складывается из длины его составляющих. Длина конвейера для различного количества блоков оценки представлена в табл. 2.

Таблица 2

Длина конвейера ПЛИС в зависимости от количества блоков

N	Total pipeline length
10	80
20	130
25	155
30	180

Описанным методом в существующих микросхемах ПЛИС можно реализовать ASWM фильтр, содержащий до 50 блоков оценки весов, чего достаточно для получения приемлемого качества фильтрации.

Описанная структура на тактовой частоте 100 МГц позволяет фильтровать 48 кадров монохромных изображений формата FullHD в секунду, чего достаточно для работы в режиме реального времени.

4. Выводы

В данной работе была разработана программная модель ASWM фильтра. Проведен анализ работы фильтра с ограниченным количеством итераций подмодуля оценки весов. Рассмотрены методы реализации модуля деления. Разработан алгоритм, который оптимальный с точки зрения отношения аппаратных затрат к точности.

Предложен метод реализации полностью конвейеризованного фильтра изображений ASWM. Рассмотрены использованные методики, позволяющие существенно сократить необходимые для реализации аппаратные ресурсы ПЛИС без критических потерь качества фильтрации. Фильтр синтезирован для основных семейств ПЛИС Intel и требует для реализации наиболее производительной версии менее 65000 логических элементов ПЛИС. В тоже время при тактовой частоте 100 МГц фильтр позволяет обрабатывать до 48 кадров/с.

Литература

1. Gonzalez, R. C. (2003). *Digital Image Processing*. Beijing: Publishing House of Electronics Industry, 123–124.
2. Brownrigg, D. R. K. (1984). The weighted median filter. *Communications of the ACM*, 27 (8), 807–818. doi: <http://doi.org/10.1145/358198.358222>
3. Goyal, P., Chaurasia, V. (2017). Application of median filter in removal of random valued impulse noise from natural images. *2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)*, 1, 125–128. doi: <http://doi.org/10.1109/iceca.2017.8203657>
4. Konieczka, A., Balcerek, J., Dabrowski, A. (2018). Method of adaptive pixel averaging for impulse noise reduction in digital images. *2018 Baltic URSI Symposium (URSI)*. Poznan, 221–224. doi: <http://doi.org/10.23919/ursi.2018.8406738>
5. Dawood, H., Dawood, H., Guo, P. (2015). Removal of random-valued impulse noise by Khalimsky grid. *2015 Asia Pacific Conference on Multimedia and Broadcasting*. doi: <http://doi.org/10.1109/apmediacast.2015.7210268>
6. Darus, M. S., Sulaiman, S. N., Isa, I. S., Hussain, Z., Tahir, N. M., Isa, N. A. M. (2016). Modified hybrid median filter for removal of low density random-valued impulse noise in images. *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. Batu Ferringhi, 528–533. doi: <http://doi.org/10.1109/iccsce.2016.7893633>
7. Zhang, X., Liao, H., Du, X., Xu, B. (2018). A Fast Hybrid Noise Filtering Algorithm Based on Median-Mean. *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*. Changchun, 2120–2125. doi: <http://doi.org/10.1109/icma.2018.8484392>
8. Hsieh, C., Huang, P., Zhao, Q. (2018). Impulse Noise Replacement With Adaptive Neighborhood Median Filtering. *2018 International Conference on*

Machine Learning and Cybernetics (ICMLC). Chengdu, 491–496. doi: <http://doi.org/10.1109/icmlc.2018.8527058>

9. Akkoul, S., Ledee, R., Leconge, R., Harba, R. (2010). A New Adaptive Switching Median Filter. *IEEE Signal Processing Letters*, 17 (6), 587–590. doi: <http://doi.org/10.1109/lsp.2010.2048646>

10. Kang, C.-C., Wang, W.-J. (2009). Modified switching median filter with one more noise detector for impulse noise removal. *AEU – International Journal of Electronics and Communications*, 63 (11), 998–1004. doi: <http://doi.org/10.1016/j.aeue.2008.08.009>

11. Akkoul, S., Ledee, R., Leconge, R., Harba, R. (2010). A New Adaptive Switching Median Filter. *IEEE Signal Processing Letters*, 17 (6), 587–590. doi: <http://doi.org/10.1109/lsp.2010.2048646>

12. Wang, Z., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13 (4), 600–612. doi: <http://doi.org/10.1109/tip.2003.819861>

13. *Intel FPGA Product catalog, Version 20.3*. Available at: <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/sg/product-catalog.pdf>