

Эффективная реализация медианного фильтра на FPGA

Байрак С.А.

Кафедра электронных вычислительных машин
Белорусский государственный университет информатики и радиоэлектроники \\
Минск, Республика Беларусь
bairak@bsuir.by

Аннотация—В статье представлена эффективная реализация медианного фильтра на микросхеме FPGA. Ее особенностью является использование информации, полученной при обработке пиксела на предыдущей итерации. Это позволяет сократить аппаратные затраты, а также увеличить скорость обработки. Предлагаемая архитектура протестирована на микросхемах класса Spartan 6 компании Xilinx и может работать на частотах до 100 МГц.

Ключевые слова: медианный фильтр; нелинейные фильтры; импульсный шум; FPGA

I. ВВЕДЕНИЕ

Фильтрация шумов одна из самых часто выполняемых операций предварительной обработки изображений. Не редко на изображениях преобладает импульсный шум, удаление которого линейными фильтрами приводит к достаточно сильным искажениям изображения, а точнее его "размытию" и удалению мелких деталей. В этой ситуации эффективно использовать вместо линейных фильтров – нелинейные, наиболее популярным из которых является медианный фильтр. Отличительной особенностью медианных фильтров является удаление импульсного шума без существенного искажения исходного изображения [1].

В случае использования в качестве входных данных видеопотока, зачастую требуется его обработка в режиме реального времени. Это приводит к необходимости аппаратной реализации фильтра, так как в случае его программной реализации скорость обработки данных оказывается значительно ниже требуемой. Наиболее гибким и эффективным решением с точки зрения критерия цена/производительность в этом случае является использование микросхем программируемой логики, а точнее одной из их разновидностей - микросхем FPGA.

Большинство пространственных фильтров, к которым также относится и медианный фильтр, работают в определенной окрестности центрального пиксела, часто называемой окном обработки. Для выполнения медианной фильтрации всего изображения это окно должно "скользить" по нему в направлении обработки каждого пиксела. В случае, если входные данные представляют собой видеопоток, они поступают последовательно, строка за строкой, пиксел за пикселом. Для реализации медианной фильтрации становится необходимым обеспечить в определенный момент времени доступ ко всему окну обработки. Типовая аппаратная архитектура,

используемая для получения такого "скользящего" окна обработки представлена на рисунке 1 [2].

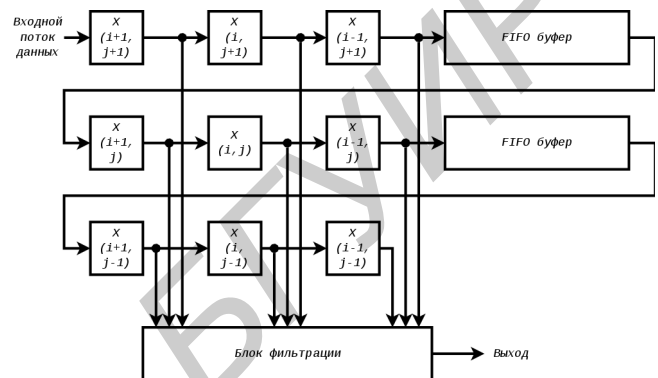


Рис. 1. Типовая аппаратная архитектура для реализации "скользящего" окна обработки размером 3x3.

На рисунке 1 скользящее окно обработки представлено совокупностью блоков X, чаще всего реализуемых с помощью регистров. FIFO буферы предназначены для временного хранения промежуточных строк. Их глубина зависит от ширины изображения и может быть рассчитана как "ширина изображения - 3", а их количество - на единицу меньше высоты скользящего окна. Блок фильтрации непосредственно реализует алгоритм медианной фильтрации, а на его входы подаются сразу все значения, сформированные "скользящим" окном.

Блок фильтрации можно реализовать несколькими способами [3], но наиболее популярным является использование так называемой сети сортировки. Она строится с использованием базовых элементов, на входы которых подаются значения пикселей, а на выходах формируются, как минимум, наибольшее и наименьшее значения. Классическая сеть сортировки для девяти пикселей и с использованием двухвходовых базовых элементов требует 41 такой элемент. Если выполнить определенную оптимизацию, их количество можно сократить до 27, и даже до 19 [4]. Однако во всех этих случаях при обработке очередного пиксела не используется информация, полученная на предыдущем этапе, что могло бы привести к уменьшению аппаратных затрат и увеличению скорости обработки.

II. ПРЕДЛАГАЕМАЯ АРХИТЕКТУРА МЕДИАННОГО ФИЛЬТРА

На рисунке 2 представлена часть обрабатываемого массива для скользящего окна размером 3x3.

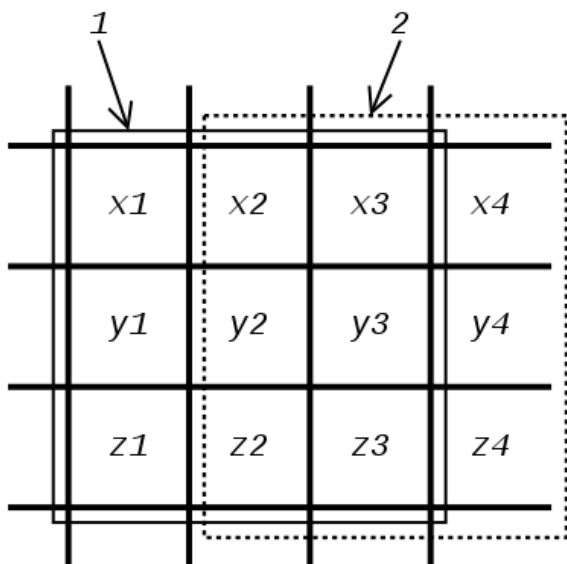


Рис. 2. Пример обработки данных для "скользящего" окна размером 3x3.

Здесь строки обозначены буквами X, Y и Z. Элементы $x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3$ представляют собой текущее состояние скользящего окна, которое обведено сплошной линией и обозначено цифрой 1. Последующее состояние скользящего окна обведено пунктиром и обозначено цифрой 2. На рисунке видно, что обозначенные окна имеют общие элементы. Это означает что можно использовать часть результатов вычислений текущей итерации для расчета последующей, что приведет к упрощению аппаратной части и увеличению скорости обработки. При этом стоит заметить, что с увеличением размеров окна обработки, количество общих элементов в процентном отношении к общему числу элементов будет расти.

Для вычисления медианы можно воспользоваться следующим алгоритмом: сначала сортируем столбцы "скользящего" окна, затем строки, а затем главную диагональ. Медианой окна обработки будет центральный элемент главной диагонали. Так как часть столбцов для текущей и предыдущей итерации одинаковы, то их сортировку этих можно не выполнять, а полученные значения "задержать" до следующей итерации. Таким образом количество операций, требуемых для поиска медианы, снижается, а значит экономятся аппаратные ресурсы и время обработки.

Обобщенная схема предлагаемой архитектуры для размера окна 3x3 представлена на рисунке 3.

Входные данные через блок формирования "скользящего" окна поступают на блок сортировки столбца. На выходе этого блока формируются отсортированные значения столбцов, которые далее сохраняются в буферах для последующих итераций. Схемотехнически буферы реализуются на базе регистров. Блок формирования главной диагонали фактически выполняет операцию нахождения максимально, среднего и минимального элементов

трех строк, в результате чего и происходит формирование главной диагонали. Последний этап - блок формирования медианы, на котором выполняется сортировка главной диагонали. Центральный элемент результата этой операции - медиана окна обработки.

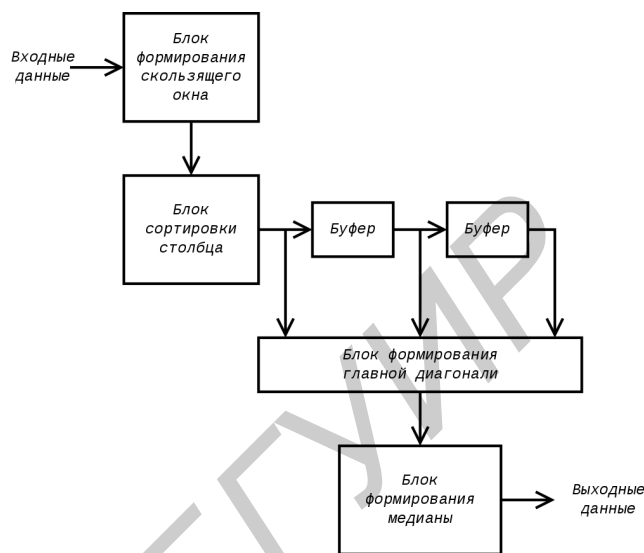


Рис. 3. Обобщенная схема архитектуры медианного фильтра для окна обработки размером 3x3.

Центральным элементов для реализации всех перечисленных блоков является блок сортировки, имеющий для указанного размера окна три входа и три выхода. На входы подаются значения элементов, а на выходе формируется их упорядоченная последовательность. Блок сортировки реализован с помощью компараторов и представляет собой тривиальную комбинационную схему.

III. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Предложенная архитектура была реализована на микросхеме FPGA Spartan 6 компании Xilinx, являющейся ядром платы SP605. Для разрядности данных 8 бит и окна обработки размером 3x3 частота работы полученного устройства равнялась 100 МГц, при этом кадр изображения размером 1024x1024 пиксела обрабатывался за время равное 15 мс. Это с запасом обеспечивает обработку данных с частотой 30 кадров в секунду в реальном масштабе времени.

- [1] S. Marshall, Logic-based Nonlinear Image Processing. Bellingham, Washington, 2007, pp. 57-71.
- [2] Ch.Ravi Kumar, S.K. Srivathsa, "EHW architecture for design of adaptive median filter for noise reduction", European Journal of Scientific Research, Vol.36 No.3 (2009), pp 473-479.
- [3] Zdenek Vasicek, Lukas Sekanina, "Novel hardware implementation of adaptive media filters", Proc. of 2008 IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop, 2008, p. 110-115.
- [4] Miguel A. Vega-Rodríguez, Juan M. Sánchez-Pérez, Juan A. Gómez-Pulido, "An FPGA-based implementation for median filter meeting the real-time requirements of automated visual inspection system", Proceedings of the MED2002, 2002.