

## КРАТКИЕ СООБЩЕНИЯ

УДК 004.931; 004.932

А.С. Поляков, В.Е. Самсонов

**ВОЗМОЖНОСТИ АППАРАТНОЙ РЕАЛИЗАЦИИ АЛГОРИТМОВ ФИЛЬТРАЦИИ ВИДЕОИЗОБРАЖЕНИЙ В РЕАЛЬНОМ МАСШТАБЕ ВРЕМЕНИ**

*Анализируются результаты разработки и моделирования аппаратной реализации алгоритмов улучшения качества видеоизображений в автоматизированных системах видеонаблюдения. Приводятся оценки производительности аппаратной реализации алгоритмов фильтрации видеоизображений.*

**Введение**

Развитие современных информационных технологий стимулирует исследования в области обработки изображений и распознавания образов. Одной из актуальных задач при этом является разработка методов, алгоритмов, способов и средств, обеспечивающих существенное повышение скорости обработки данных. Особенно важно повышение скорости обработки в системах автоматизированного видеонаблюдения, где требуется решать задачи распознавания и сопровождения подвижных объектов. В таких системах используются специальные многоуровневые алгоритмы [1], в том числе алгоритмы улучшения качества изображения или подчеркивания определенных признаков. На нижнем уровне применяются алгоритмы предварительной обработки для улучшения качества изображения, на последующих уровнях используются алгоритмы быстрого вычисления оптического потока, а также распознавания и интеллектуального сопровождения объектов.

Для применяемого в вычислении оптического потока алгоритма на основе расчета корреляции адаптивного окна необходима большая вычислительная мощность [2, 3], обеспечивающая режим реального времени слежения за объектами. В связи с этим для повышения эффективности системы видеонаблюдения за подвижными объектами существенным является предоставление функциям верхнего уровня максимальных вычислительных ресурсов.

В задачах нижнего уровня необходимо выполнять значительное количество однотипных базовых операций над большими массивами данных, что требует существенных временных затрат. Для решения этой проблемы предлагается множество вариантов параллельных высокопроизводительных вычислительных систем – от универсальных многопроцессорных систем до специализированных устройств.

Общепризнано, что одним из наиболее эффективных способов решения указанной выше проблемы является аппаратная реализация алгоритмов на основе современных микроэлектронных средств. Аппаратная реализация в большинстве случаев оказывается очень эффективной, поскольку отличительной особенностью многих задач из рассматриваемой области является возможность параллельного выполнения базовых операций над большими массивами однотипных данных. К сожалению, сведений о возможностях аппаратной реализации алгоритмов обработки изображений в отечественной литературе практически нет.

В настоящей статье предпринята попытка в какой-то мере устранить отмеченный недостаток и представить сведения, позволяющие оценить возможности аппаратной реализации алгоритмов фильтрации изображений, которые предназначены для улучшения качества видеопоследовательностей, получаемых с видеокамер в режиме реального времени.

**1. Структура устройства, назначение и функции основных блоков**

Для аппаратной реализации алгоритмов обработки изображений выбрана элементная база FPGA фирмы XILINX.

Устройство обработки изображений (рис. 1) состоит из следующих компонентов:

1. FPGA фирмы XILINX, микросхема xc4vlx80 семейства Virtex-4 [4, 5]. В FPGA загружается конфигурационная последовательность, которая обеспечивает реализацию алгоритмов фильтрации видеоизображений, управляющей логики и интерфейсов с другими компонентами устройства.

2. Видеодекодер. На вход поступает аналоговый сигнал видеокамеры, на выходе выдаются оцифрованные данные.

3. Двухпортовое ОЗУ типа IDT объемом 1 Мб. В памяти хранятся оцифрованные видеодекодером кадры изображения, которые по запросам FPGA поступают на фильтрацию.

4. Fifo типа IDT объемом 1 Мб. В Fifo записываются отфильтрованные кадры, которые в дальнейшем передаются в ПЭВМ для последующей обработки программными средствами верхнего уровня.

5. Микросхема CPLD фирмы XILINX семейства XC95. Осуществляет специальные функции управления контроллером PCI для прямого доступа в память.

6. Контроллер PCI фирмы AMCC. Осуществляет в режиме прямого доступа передачу отфильтрованных данных в ПЭВМ и управляющих сигналов в обоих направлениях.

7. Конфигурационное ПЗУ фирмы XILINX. Предназначено для хранения конфигурационной последовательности микросхемы FPGA.

8. Тактовый генератор. Обеспечивает рабочую тактовую частоту до 400 МГц для использования компонентами устройства.

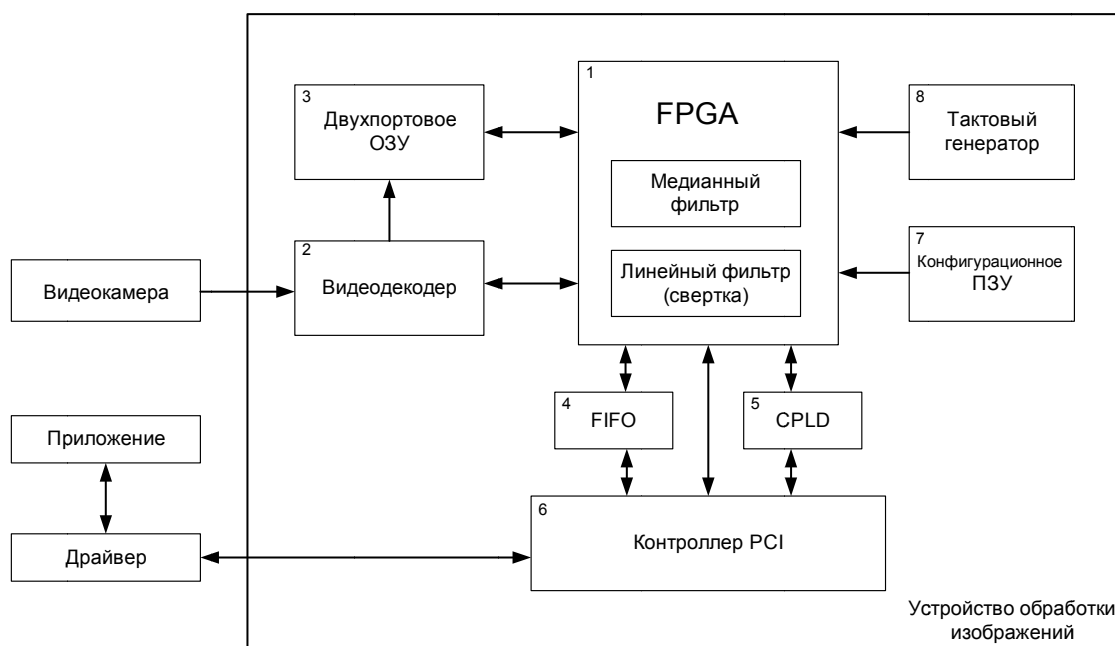


Рис. 1. Структурная схема устройства обработки видеоизображений

Прототипом разрабатываемого устройства является устройство захвата видеопоследовательности, разработанное авторами для исследования возможностей обработки изображений и оценки быстродействия операций по сохранению оцифрованных изображений в памяти ПЭВМ (рис. 2). Элементная база этого прототипа практически аналогична вновь разрабатываемому устройству. В прототипе используется многофункциональный видеодекодер с управлением по интерфейсу I2C, FPGA фирмы XILINX семейства SPARTAN, FIFO типа IDT, контроллер PCI фирмы AMCC. На этом прототипе производилась тестовая проверка алгоритмов фильтрации с ограничением по степени распараллеливания.

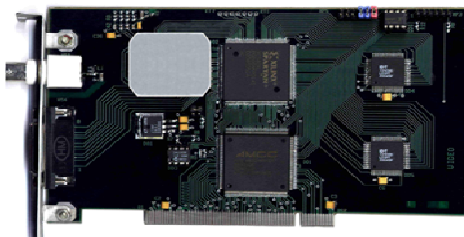


Рис. 2. Устройство захвата видеопоследовательности

## 2. Реализованные алгоритмы фильтрации изображений

В настоящее время реализованы два алгоритма фильтрации изображений: медиана и линейный фильтр.

*Медианой* дискретной последовательности  $N$  элементов при нечетном  $N$  называется элемент, для которого существует не менее  $(N-1)/2$  элементов, меньших или равных ему, и не менее  $(N-1)/2$  элементов, больших или равных ему по величине. Для нахождения медианы разработано много алгоритмов [6, 7], отличающихся производительностью.

Рассмотрим матрицу  $M$  размером  $3 \times 3$  элемента (рис. 3, а). Для нахождения медианы необходимо упорядочить все элементы матрицы по возрастанию значений. Пятый элемент упорядоченной последовательности элементов  $(m_{1,1}, m_{1,2}, m_{1,3}, m_{2,1}, m_{2,2}, m_{2,3}, m_{3,1}, m_{3,2}, m_{3,3})$  будет ее медианой. При рассмотрении матрицы размером  $3 \times 3$  элемента для нахождения медианы путем простого сравнения элементов требуется произвести 30 операций. При аппаратной реализации поиска медианы можно обойтись меньшим количеством сравнений, если выполнить следующие действия:

1. Упорядочить столбцы матрицы  $M$  по неубыванию значений и получить матрицу  $M_1$  (рис. 3, б), элементы которой находятся в отношениях  $d_1 \leq d_2 \leq d_3, e_1 \leq e_2 \leq e_3, f_1 \leq f_2 \leq f_3$ .

2. Упорядочить строки матрицы  $M_1$  по неубыванию значений и получить матрицу  $M_2$  (рис. 3, в), элементы которой находятся в отношениях  $k_1 \leq k_2 \leq k_3, m_1 \leq m_2 \leq m_3, n_1 \leq n_2 \leq n_3$ .

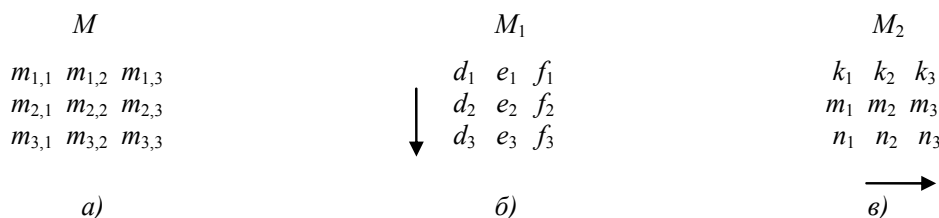


Рис. 3. Схема упорядочивания элементов матрицы (стрелками указаны направления возрастания значений элементов в столбцах и строках)

Поскольку упорядочивание элементов матрицы производилось по неубыванию сверху вниз и слева направо, то это означает, что в число искомых элементов входят только те элементы матрицы, которые расположены на главной диагонали и слева от нее, т. е. элементы  $k_1, k_2, k_3, m_1, m_2, n_1$ , причем элементы  $k_1, k_2$  и  $m_1$  занимают первые три места в упорядоченной по отношению неубывания последовательности элементов матрицы  $M$ . Следовательно, для нахождения медианы необходимо найти средний по значению элемент, расположенный на главной диагонали матрицы  $M_2$ . Значение этого элемента и будет представлять собой медиану матрицы  $M$ .

В качестве *линейного фильтра* используется алгоритм, называемый «сверткой», который обеспечивает преобразование матрицы первоначальных данных в результирующую матрицу с использованием некоторой эвристической матрицы поправочных коэффициентов  $L$ . Элементами этой матрицы являются целые числа со знаками «+» или «-», задаваемые автором производимых преобразований. Рассматриваемые матрицы показаны на рис. 4.

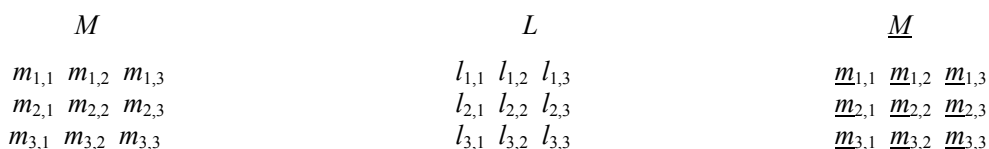


Рис. 4. Матрицы  $M, L$  и  $\underline{M}$

Исходная матрица  $M$  преобразуется в матрицу  $\underline{M}$  с использованием элементов матрицы  $L$  следующим образом:

1) вычисляется значение  $k = l_{1,1} \cdot m_{1,1} + l_{1,2} \cdot m_{1,2} + l_{1,3} \cdot m_{1,3} + l_{2,1} \cdot m_{2,1} + l_{2,2} \cdot m_{2,2} + l_{2,3} \cdot m_{2,3} + l_{3,1} \cdot m_{3,1} + l_{3,2} \cdot m_{3,2} + l_{3,3} \cdot m_{3,3}$ ;

2) в зависимости от величины  $k$  принимается значение  $m_{2,2}$ :  $m_{2,2} = 255$ , если  $k \geq 255$ ;  $m_{2,2} = 0$ , если  $k \leq 0$ ;  $m_{2,2} = k$  в остальных случаях.

### 3. Алгоритм обработки видеокадров

Получаемое с видеокамер изображение после оцифрования и упорядочения четных и нечетных строк записывается в двухпортовое ОЗУ, размер которого позволяет хранить одновременно два кадра изображения следующих форматов:

- minkadr –  $320 \times 240$  пикселей размером 8 бит каждый;
- middlkadr –  $640 \times 480$  пикселей размером 8 бит каждый;
- maxkadr –  $768 \times 576$  пикселей размером 8 бит каждый.

Благодаря наличию памяти для двух видеокадров обеспечивается возможность параллельного выполнения операций обработки текущего кадра изображения и записи следующего. Был разработан проект аппаратной реализации фильтрации видеокадров с использованием алгоритмов «медиана» и «свертка». Аппаратная реализация алгоритмов обработки изображений предоставляет возможность повышения их производительности благодаря организации параллельной обработки различных фрагментов матрицы изображения, а также параллельного выполнения отдельных этапов алгоритма обработки изображения.

Поскольку для перевычисления каждого элемента матрицы в соответствии с указанными выше алгоритмами требуется знание значений соседних с ним элементов матрицы, то идеальным случаем было бы наличие в памяти FPGA значений элементов всего кадра изображения. Вместе с тем это потребовало бы применения дорогих микросхем. С целью оптимизации показателя «стоимость/производительность» были проведены исследования проектов с различными размерами внутренней памяти (регистров) и схем обработки. В результате была выбрана сравнительно недорогая микросхема xc4vlx80 серии Virtex 4 фирмы XILINX.

Основным объектом обработки данных в обоих рассматриваемых алгоритмах фильтрации является матрица размером  $3 \times 3$  элемента, поэтому в качестве базовой была выбрана схема, предназначенная для обработки матрицы указанного размера. Аппаратные возможности указанной выше микросхемы позволили разместить 80 элементарных схем обработки, сосредоточенных в 10 базовых обрабатывающих блоках, в каждом из которых производится обработка 8 матриц размером  $3 \times 3$  элемента. Эти блоки содержат соответствующее количество регистров, необходимых для хранения получаемых из ОЗУ данных. Поскольку для перевычисления значений элементов  $m_{2,1}$ ,  $m_{2,2}$ ,  $m_{2,3}$  необходимо иметь значения не только элементов матрицы  $M$ , но и значения элементов столбцов слева и справа от матрицы, то для обеспечения обработки каждой элементарной схемы в регистры загружаются значения девяти элементов рассматриваемой матрицы, а также значения столбцов, расположенных слева и справа от матрицы. Следовательно, при вычислении значения элемента  $m_{2,3}$  используются значения левого столбца элементарной матрицы, расположенной справа от рассматриваемой, а при вычислении элемента  $m_{2,1}$  – значения правого столбца матрицы, расположенной слева от рассматриваемой.

Одновременно в микросхему загружаются и обрабатываются  $8 \times 4 \times 10 = 320$  элементов одной строки видеокадра, каждый из которых представляется в виде одного байта. Поскольку элементарная обрабатываемая матрица содержит три строки, то одновременно загружаются и обрабатываются  $320 \times 3 = 960$  элементов видеокадра, т. е. три строки по 320 байтов. При этом надо иметь в виду, что при каждой загрузке указанного количества элементов видеокадра обработке (перевычислению значений) подвергаются только элементы средней строки матрицы, т. е. 320 элементов видеокадра.

Необходимо также учитывать, что обработке не подвергаются элементы первой и последней строк видеокадра, поскольку для перевычисления их значений нет необходимых данных, а именно значений элементов строки сверху – для верхней строки видеокадра и значений элементов строки снизу – для нижней строки видеокадра.

Основная проблема обработки видеокадра в реальном масштабе времени заключается в невозможности производить одновременно фильтрацию всего видеокадра из-за недостаточности аппаратных возможностей используемой микросхемы. Поэтому для полной обработки видеокадра необходимо производить несколько сеансов загрузки данных. Основные потери времени при обработке видеокадра приходятся именно на процедуру загрузки данных.

Количество загрузок данных в микросхему зависит от формата видеокadra и для выбранной микросхемы составляет:

- для формата *minkadr*  $320 \times 238 \times 3 : 960 = 238$ ;
- для формата *middlkadr*  $640 \times 478 \times 3 : 960 = 956$ ;
- для формата *maxkadr*  $768 \times 574 \times 3 : 960 = 1378$ .

Алгоритм загрузки данных из ОЗУ в FPGA достаточно сложен, что обусловлено различными форматами представления данных в ОЗУ и в FPGA, поэтому из-за ограниченности места в статье приводить его не будем.

#### 4. Объемно-временные характеристики аппаратной реализации алгоритмов фильтрации видеозображений

Аппаратная реализация алгоритмов обработки изображений предоставляет возможность повышения производительности не только вследствие более высокой скорости выполнения операций, но и благодаря организации параллельной обработки различных фрагментов матрицы изображения и параллельного выполнения отдельных этапов алгоритма обработки. Так, выше было указано, что операции считывания данных из ОЗУ в FPGA и считывания результатов обработки предыдущей порции исходных данных из Fifo в ПЭВМ производятся одновременно (параллельно) с обработкой следующего фрагмента исходных данных.

Для оценки производительности алгоритмов обработки видеокadров использовался метод логического моделирования. Разработанный в системе проектирования XILINX Ise проект реализации указанных выше алгоритмов фильтрации (медиана и линейный фильтр) был синтезирован, в результате чего установлено, что проект занимает в выбранной микросхеме 86 % ресурсов.

В результате логического моделирования разработанных проектов получены данные о затратах на выполнение отдельных операций алгоритмов и на выполнение в целом всех операций по обработке видеокadров. Анализ результатов моделирования показал, что на выполнение операций в соответствии с рис. 5 необходимо количество тактов, указанное в таблице.

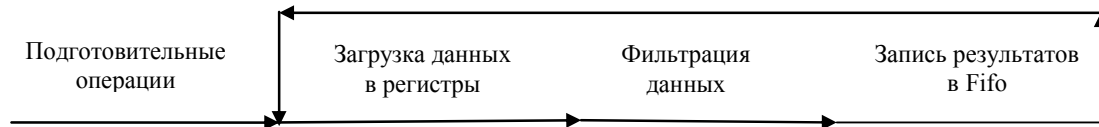


Рис. 5. Укрупненная схема алгоритма обработки данных

Данные о производительности алгоритмов фильтрации, число тактов

Операции алгоритма	Алгоритм, число тактов	
	медиана	линейный фильтр
Подготовительные операции	2	2
Загрузка данных в регистры	288	288
Фильтрация данных	6	57
Запись результатов обработки в Fifo	36	36
Итого (для одного блока данных)	332	383
Количество тактов для обработки кадров:		
<i>minkadr</i> (238 загрузок)	$238 \times 332 = 79\ 016$	$238 \times 383 = 91\ 154$
<i>middlkadr</i> (956 загрузок)	$956 \times 332 = 317\ 392$	$956 \times 383 = 366\ 148$
<i>maxkadr</i> (1378 загрузок)	$1378 \times 332 = 457\ 496$	$1378 \times 383 = 527\ 774$

Из таблицы видно, что обработка максимального по размеру кадра изображения (*maxkadr*) наиболее трудоемким из алгоритмов (свертка) производится за 527 774 такта. Используя эти данные, можно вычислить время, затрачиваемое на обработку одного видеокadra. Для получения временных оценок необходимо учесть быстродействие используемой микросхемы, т. е. рабочую тактовую частоту. Выбранная микросхема xc4vlx80 может работать на частоте 400 МГц. Следовательно, за одну секунду производится обработка  $400\ 000\ 000 : 527\ 774 = 760$  видеокadров, т. е. один ви-

деокадр обрабатывается за 1,33 мс. Это значительно меньше, чем реальное время смены кадров изображения в процессе съемки, равное 40 мс.

### Заключение

В результате проведенных исследований установлено, что аппаратная реализация алгоритмов фильтрации изображений позволяет обеспечить обработку видеопоследовательностей в реальном масштабе времени.

Можно варьировать соотношение производительность/стоимость путем выбора различных типов микросхем. Например, при использовании более дешевой (соответственно имеющей меньшие функциональные возможности) микросхемы придется уменьшить размер обрабатываемого фрагмента видеокадра и увеличить количество загрузок фрагментов. Следовательно, можно подобрать микросхему минимальной стоимости, которая будет обеспечивать обработку видеопоследовательностей в реальном масштабе времени.

### Список литературы

1. Программно-аппаратный комплекс слежения в реальном масштабе времени за движущимися объектами / Б.А. Залесский [и др.] // Информационные технологии программы Союзного государства «Триада». Основные результаты и перспективы : сб. науч. тр. – Минск : ОИПИ НАН Беларуси, 2010. – С. 262–269.
2. An Unbiased Second-Order Prior for High-Accuracy Motion Estimation / W. Trobin [et al.] // Pattern Recognition (Proc. DAGM). – Munich, Germany : Springer, LNCS, 2008. – P. 396–405.
3. Schoenemann, T. Globally Optimal Shape-based Tracking in Real-time / T. Schoenemann, D. Cremers // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). – Anchorage, Alaska, 2008. – P. 1–6.
4. Кузелин, М.О. Современные семейства ПЛИС фирмы Xilinx : справочное пособие / М.О. Кузелин, Д.А. Кнышев, В.Ю. Зотов. – М. : Горячая линия – Телеком, 2004. – 440 с.
5. Xilinx, Inc. [Electronic recourse]. – 2011. – Mode of access : <http://www.xilinx.com>. – Date of access : 25.10.2011.
6. Кравчонок, А.И. Алгоритмы медианной фильтрации с окном  $3 \times 3$  на основе неполных сортирующих сетей / А.И. Кравчонок // Информатика. – 2009. – № 1 (21). – С. 91–102.
7. Кравчонок, А.И. Алгоритмы медианной фильтрации с окном  $3 \times 3$  при помощи MMX- и SSE-инструкций процессоров семейства x86 / А.И. Кравчонок // Информатика. – 2010. – № 1 (25). – С. 41–57.

Поступила 23.01.12

*Объединенный институт проблем  
информатики НАН Беларуси,  
Минск, Сурганова, 6  
e-mail: alexpolja@tut.by,  
sveby@mail.ru*

**A.S. Poljakov, V.E. Samsonov**

### POSSIBILITY OF HARDWARE IMPLEMENTATION OF A REAL-TIME VIDEO PROCESSING ALGORITHMS

Hardware implementation and modeling algorithms for video image quality improvement in video monitoring automated systems are discussed. Results of quantitative assessment of hardware-based implementation of image filtration algorithms are also presented.