



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 126 за 2016 г.



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Держановский А.С., [Соколов С.М.](#)

Обработка зрительных
данных в системе
технического зрения
реального времени с
использованием ПЛИС

Рекомендуемая форма библиографической ссылки: Держановский А.С., Соколов С.М. Обработка зрительных данных в системе технического зрения реального времени с использованием ПЛИС // Препринты ИПМ им. М.В.Келдыша. 2016. № 126. 16 с. doi:[10.20948/prepr-2016-126](https://doi.org/10.20948/prepr-2016-126)
URL: <http://library.keldysh.ru/preprint.asp?id=2016-126>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук**

А.С. Держановский, С.М. Соколов

**Обработка зрительных данных
в системе технического зрения
реального времени
с использованием ПЛИС**

Москва — 2016

УДК 004.932.2

Держановский А.С., Соколов С.М.

Обработка зрительных данных в системе технического зрения реального времени с использованием ПЛИС

Одним из перспективных средств аппаратной поддержки сбора и обработки зрительных данных являются программируемые логические интегральные схемы (ПЛИС). Наибольший объём зрительных данных приходится на этап первичной обработки, до появления первых признаков качества изображения и образов объектов на нём. От качества и своевременности получения данных первичной обработки существенно зависит возможность успешного решения задач системой технического зрения (СТЗ) в масштабе реального времени (РВ). В работе описываются результаты исследований по экспериментальной оценке эффективности использования ПЛИС в составе средств аппаратной поддержки СТЗ реального времени. Дается общее описание этого средства, трудозатрат, связанных с его программированием и использованием в составе СТЗ РВ. Специальное внимание уделено сравнительному анализу временных затрат при реализации алгоритмов обработки изображений с помощью универсального процессора Intel Quad Core i7 2.5GHz и ПЛИС Cyclone IV GX EP4CGX150DF31. Приводятся количественные результаты экспериментов по обработке с помощью ПЛИС изображений различных объектов.

Ключевые слова: система технического зрения, ПЛИС, первичная обработка, обработка зрительных данных в режиме реального времени, программное обеспечение

Derzhanovsky Alexander Sergeevich, Sokolov Sergei Mikhailovich

Image processing in real-time computer vision systems using FPGA

One of perspective means of hardware support of collection and processing of visual data is FPGA. The greatest amount of visual data is the share of a preprocessing stage, before emergence of the first signs of quality of the image and images of objects on it. The possibility of the successful solution of tasks system of computer vision on the scale of real time significantly depends on quality and timeliness of data acquisition of preprocessing. In work results of researches on an experimental efficiency evaluation of use of FPGA as a part of means of hardware support of system of computer vision of real time are described. The general description of this means, the labor costs connected with its programming and use as a part of system of computer vision is given. The special attention is paid to the comparative analysis of time expenditure in case of implementation of algorithms of handling of images by means of the universal Intel Quad Core i7 processor of 2,5 GHz and FPGA Cyclone IV GX EP4CGX150DF31. Quantitative results of experiments on processing by the FPGA images of different objects.

Key words: computer vision system, FPGA, image processing, real-time processing, software

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проекты №№ 16-08-01282, 16-37-00490 и Российского научного фонда, проект № 16-19-10705.

Введение

Современные системы технического зрения требуют обработки всё большего количества зрительных данных. В частности, эти потребности вызваны необходимостью получать данные с большей точностью по пространственному разрешению и использованием цветных изображений и данных с нескольких полей зрения. С другой стороны, для СТЗ реального времени способность обрабатывать зрительные данные в рамках отпущенных временных интервалов является принципиальным требованием.

Процесс обработки зрительных данных можно представить в виде айсберга, вершиной которого является конечный результат обработки. В пути, от порога айсберга (получения изображения) до его вершины (конечного результата), решается множество задач: от сбора зрительных данных, диагностики их качества, выделения низкоуровневых признаков объектов на изображении, до формирования описания/распознавания объектов на изображении и получения информации, необходимой для решения целевой задачи. Можно выделить схему движения к вершине айсберга. Известны алгоритмы решения отдельных задач по обработке зрительных данных на архитектурной базе процессоров общего назначения [7]. Все эти решения имеют ограничение – объем обрабатываемых данных, особенно если рассматривать систему технического зрения с позиций системы реального времени. С учетом этой особенности интересно исследовать и сравнить временные показатели реализаций ряда общеупотребительных алгоритмов обработки зрительных данных с помощью процессоров общего назначения и с помощью ПЛИС. В отечественной [2, 3] и зарубежной [8, 10] литературе освещается тема использования ПЛИС для обработки зрительных данных, однако в них авторы не проводят сравнительный анализ производительности ПЛИС и процессоров общего назначения. При формировании эффективной архитектуры вычислительных средств для решения практических задач робототехники количественные оценки производительности и трудоёмкости реализации тех или иных решений представляются весьма полезными.

Постановка данной задачи принадлежит проф. д.ф.-м.н. С.М. Соколову. Авторы выражают благодарность к.ф.-м.н. А.А. Краснобаеву и д.ф.-м.н. А.А. Богуславскому за консультации и полезные советы при выполнении работы.

Примеры использования ПЛИС в СТЗ

Область применения ПЛИС в системах технического зрения все время расширяется, а сложность выполняемых задач возрастает.

При реализации графовых суперкомпьютеров все более широко используются ПЛИС. Так, например, компания Convey Computers (производитель ускорителей на ПЛИС) создала систему Convey HC-1ex на базе

ПЛИС с уникальной подсистемой памяти, оптимизированной под нерегулярный доступ. При этом многофункциональная реконфигурируемая система Convey HC-1ex – второе поколение встраиваемых в сокет Intel LGA 775 сопроцессоров на базе ПЛИС, выпускаемых компанией Convey Computers. На плате Convey HC-1ex две микросхемы ПЛИС используются для интерфейса с центральным процессором и декодирования команд, четыре – для выполнения пользовательской программы (эти платы также называют Application Engines (AE), остальные восемь реализуют контроллеры памяти. Для перепрограммирования пользователю доступны четыре AE, в которые можно загружать либо собственную прошивку, либо одну из предоставляемых производителем, специализированных под различные области приложений (финансовый анализ, биоинформатика, распознавание речи, графовые задачи и др.).

Еще одним примером использования ПЛИС является его применение в марсоходе «Curiosity», созданном в лаборатории Mars Science Laboratory (MSL) NASA в 2009 году. Здесь задачей ПЛИС является: повышение вычислительной пропускной способности, работа в качестве вычислительного элемента, снижающего уровень шумов, определение границ объектов и предупреждение проскальзывания элементов сцепления марсохода с поверхностью [11]. Передача самых затратных по времени вычислительных функций от процессора к ПЛИС обусловлена его уникальной особенностью ведения независимых параллельных расчетов. В последних разработках лаборатории Jet Propulsion Laboratory (JPL) имеется модуль Mobility Avionics Module, особенность которого заключается в том, что основным вычислительным элементом является ПЛИС Xilinx Virtex-II Pro, в который интегрируется процессор PowerPC 405, максимальная частота тактирования которого составляет 300 МГц. Таким образом, в одном кристалле разработчики получили процессор общего назначения, с возможностью программирования его на языках высокого уровня, и программируемую жесткую логику ПЛИС, которая была использована ими для решения наиболее затратных по времени вычислительных задач. В частности, на ее основе была реализована первичная обработка данных стереосистемы марсохода.

Из двух основных производителей ПЛИС, Xilinx и Altera, первый больше известен в сфере военных технологий и бортовой электроники, а второй – в сфере разработок для широкого коммерческого использования. Полезной характеристикой ПЛИС Altera является интеграция высокоскоростных трансиверов, позволяющих организовывать прием, обработку и передачу информации на скорости до 10 Гбит. Благодаря этому кристаллы ПЛИС от Altera имеют повсеместное применение в высокоскоростных сетевых устройствах ведущих производителей, таких как Cisco, Huawei и т.д.

В работе был использован исследовательский набор DK-DEV-4CGX150N. Данный набор, включает в себя:

- ПЛИС Cyclone IV GX EP4CGX150N;

- 256 MB DDR2 SDRAM;
- 4 MB SSRAM;
- PCI Express;
- 10/100/1000 Ethernet.

Архитектура ПЛИС

ПЛИС, или FPGA (Field-Programmable Gate Arrays) в английской транскрипции, – программируемые вентильные матрицы, состоящие, в свою очередь, из логических блоков (LAB) и коммутирующих связей – программируемых матриц соединений. На рис. 1 представлена обобщенная архитектура ПЛИС.

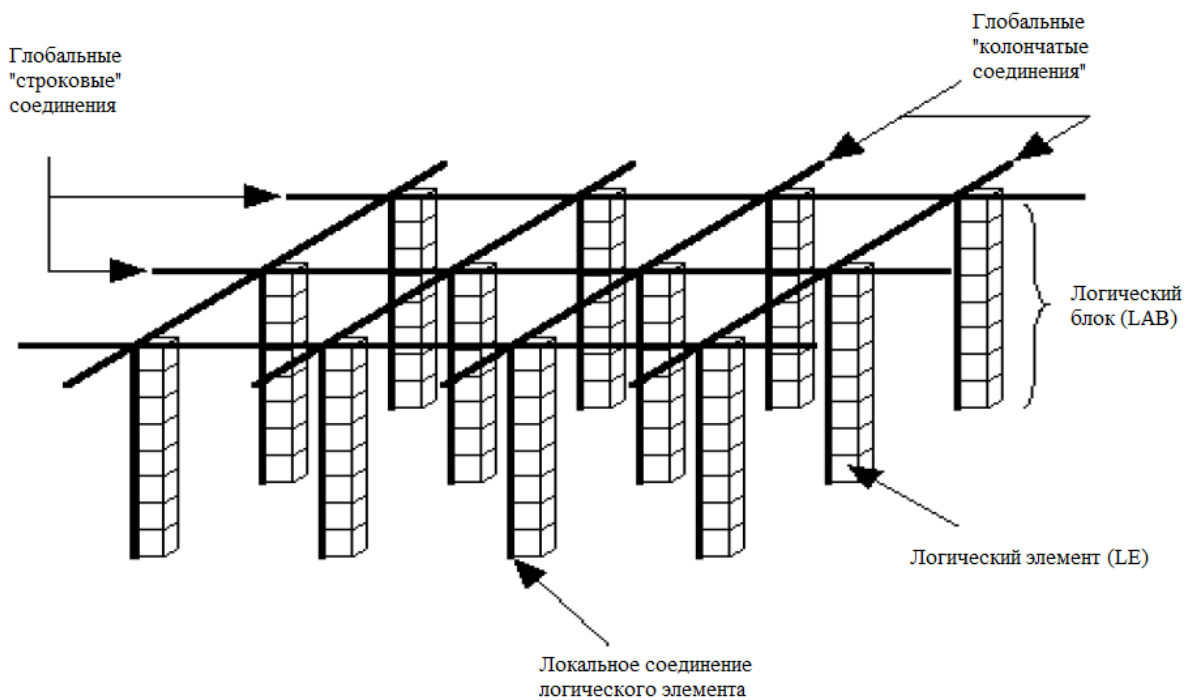


Рис. 1. Обобщенная архитектура ПЛИС

Логические блоки таких ПЛИС состоят из одного или нескольких относительно простых логических элементов, в основе которых лежит таблица перекодировки LUT (look-up table), программируемый мультиплексор, D-триггер, а также цепи управления [12]. Пример логического элемента ПЛИС семейства Cyclone IV Altera представлен на рис. 2.

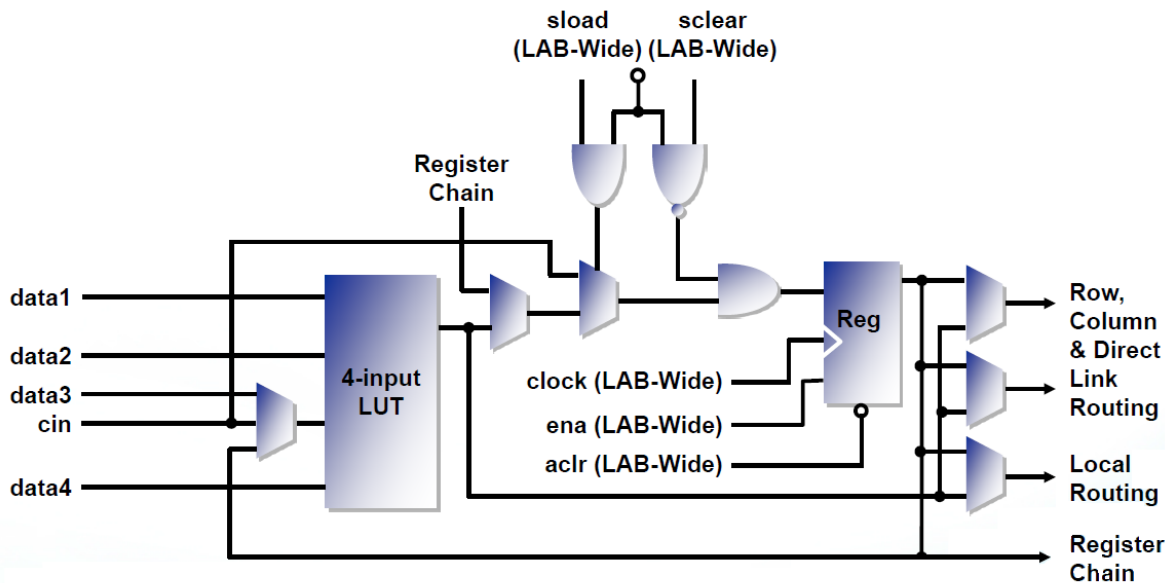


Рис. 2. Логический элемент ПЛИС линейки Cyclone IV производства фирмы Altera

Таких простых элементов может быть достаточно большое количество, у современных ПЛИС емкостью до 1 миллиона вентилей число логических элементов достигает нескольких десятков тысяч. Также некоторые семейства ПЛИС имеют встроенные реконфигурируемые модули памяти, что делает их весьма удобным средством реализации алгоритмов цифровой обработки сигналов, основными операциями в которых являются перемножение, умножение на константу, суммирование, деление, вычитание, задержка сигнала.

Одним из векторов развития ПЛИС является создание так называемой SoC (System on Chip), системы на кристалле. Этот подход заключается в реализации на ПЛИС процессора под управлением операционной системой реального времени. Реализация данного подхода выполнена производителями ПЛИС как Altera, так и Xilinx. Altera предлагает к интеграции в ПЛИС процессор собственной разработки NIOS и/или NIOS II под управлением MicroC/OS-II. Эта операционная система является операционной системой реального времени с поддержкой вытесняющей многозадачности. Таким образом, производитель ПЛИС предоставляет разработчику выбор, как реализовывать стоящие перед ним задачи. Можно описать всю работу на жесткой логике, а можно интегрировать в ПЛИС процессор и решать задачи на удобном в программировании языке C/C++. Оптимальным выбор: исполнять наиболее трудозатратные по времени реализации задачи посредством интеграции процессора, а задачи, требующие большей производительности, выполнить в жесткой логике.

В завершение необходимо подчеркнуть, что ПЛИС является гибким инструментом для достижения баланса между затратами на разработку и

производительностью результирующей системы. Дополнительного исследования требует вопрос стоимости и удобства программирования соответствующих устройств.

Структура СТЗ с использованием ПЛИС

На рис. 3 изображена функциональная схема СТЗ с использованием ПЛИС для первичной обработки зрительных данных.

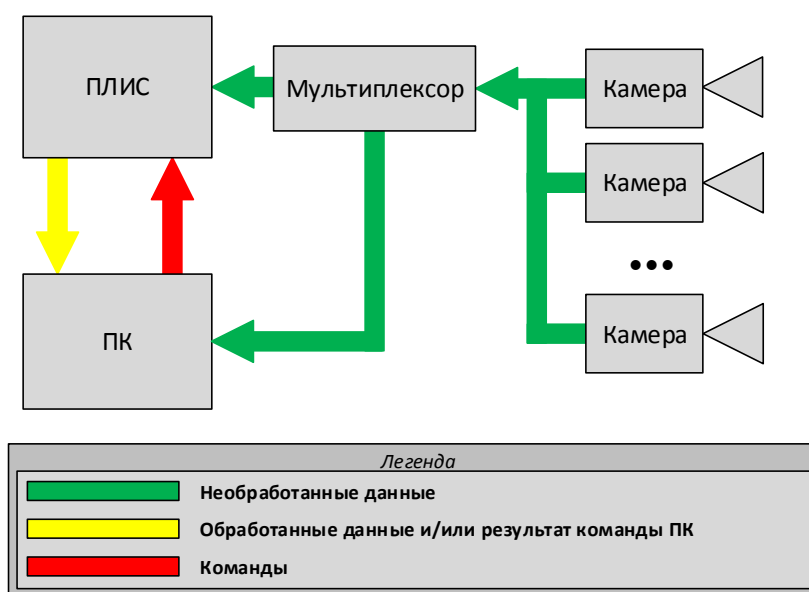


Рис. 3. Используемая в экспериментах схема аппаратной части СТЗ

Данная схема наглядно отражает предназначение программно-аппаратного подхода: разгрузка процессора общего назначения от задач первичной обработки изображения. В предложенной архитектуре блок «ПК», построенный на базе общего процессора, является частью архитектуры СТЗ, в которой реализуются поведенческие алгоритмы, исходной информацией для которых являются результаты обработки зрительных данных. Блок «ПЛИС» – это законченное изделие, имеющее периферийную часть, обеспечивающую ПЛИС оперативным запоминающим устройством (ОЗУ) и интерфейсами, необходимыми для приема/передачи информации. За счет ПЛИС СТЗ приобретает дополнительное вычислительное средство для аппаратной реализации алгоритмов обработки зрительных данных.

Требования к ПО ПЛИС в СТЗ РВ

Ожидается, что наибольший эффект с точки зрения увеличения быстродействия СТЗ может принести использование ПЛИС при обработке объемных массивов данных – на этапе ввода и первичной обработки. Полезным будет применение ПЛИС и на последующих этапах, когда необходимо

оперативно получить результаты компоновки различных вариантов первичных признаков объектов на изображении.

Удобство работы с инструментальными средствами создания программного обеспечения самым существенным образом сказывается на эффективности всего процесса разработки систем технического зрения. Прикладное программное обеспечение ПЛИС в составе СТЗ должно удовлетворять ряду общих требований:

1. модульность

(реализация функциональных блоков ПО в унифицированном интерфейсе и однотипном формате позволяет оперативно настраивать общую последовательность алгоритмической обработки зрительных данных);

2. широкий спектр алгоритмов первичной обработки;

3. универсальность

(ПО должно быть переносимым на ПЛИС различных производителей).

Описание реализации алгоритмов первичной обработки изображений на базе ПЛИС

В качестве первых алгоритмов для сравнения производительности систем обработки зрительных данных были выбраны:

- алгоритм расчета гистограммы черно-белого изображения;
- алгоритм Canny определения элементов границ на изображении.

Первый алгоритм прост в реализации, но одновременно достаточно затратен по времени для процессоров общего назначения, ввиду того что происходит постоянное обращение к памяти и, пока данные из памяти не получены, процессор «простаивает».

Второй алгоритм немного сложнее в реализации и одновременно более затратный по вычислительным ресурсам. Это является следствием как использования вычислений с плавающей точкой, так и большого количества обращений к памяти.

Кроме того, для технологического обеспечения ввода зрительных данных в ПЛИС были реализованы алгоритмы, обеспечивающие сетевой обмен информацией по Ethernet (для получения данных прямо от видеокамеры с цифровым выходом Gigabit Ethernet). На рис. 4 представлена функциональная схема реализации алгоритма на ПЛИС.

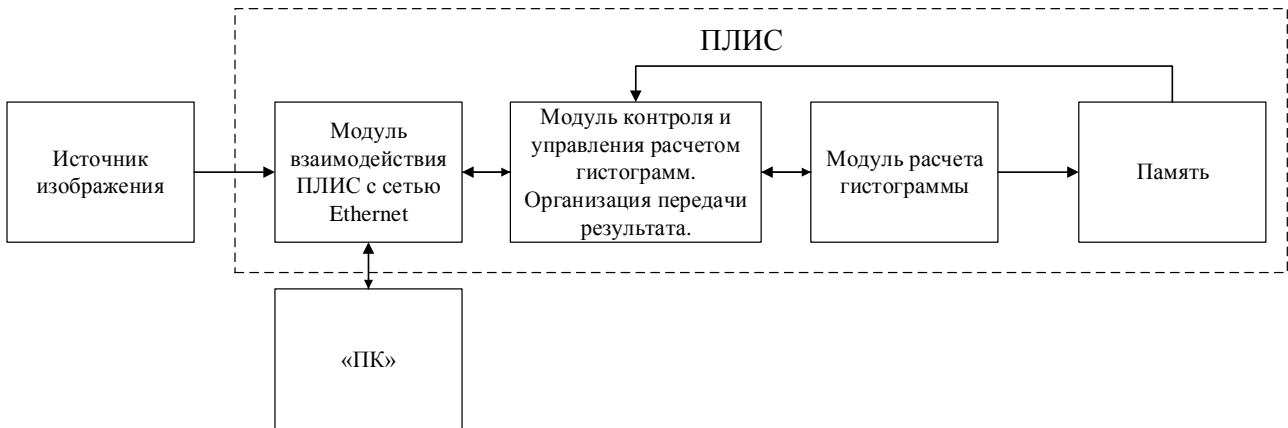


Рис. 4. Функциональная схема реализации алгоритма расчета гистограммы

В функциональной схеме источник изображения передает информацию непосредственно в ПЛИС по шине Gigabit Ethernet. После получения очередного пакета с данными и проверки его на отсутствие ошибок при передаче полученная информация передается в модуль контроля и управления расчетом гистограмм. Функции модулей следующие.

Модуль контроля и управления расчетом:

- определение принадлежности данных к текущему кадру изображения, для которого в данный момент времени ведется расчет;
- управление модулем расчета гистограммы, передача данных в модуль расчета и определение адреса памяти для хранения результата;
- если расчет гистограммы выполнен, организация передачи результата в процессор общего назначения.

Модуль расчета:

- выполнение расчета гистограммы;
- по команде от модуля контроля и управления обнуляет память перед расчетом гистограммы следующего кадра.

Функциональная схема реализации алгоритма определения элементов границ на изображении представлена на рис. 5.

Модуль взаимодействия с сетевыми устройствами по Gigabit Ethernet и модуль контроля и управления алгоритма аналогичны предыдущему алгоритму. Весь алгоритм расчета был разбит на базовые операции (сглаживание, расчет градиентов, расчет нормы градиента). Все расчеты выполняются непрерывно, реализована конвейерная архитектура модулей расчета.

Модуль управления передачей результата необходим для организации передачи результата работы алгоритма из памяти в сеть Ethernet через модуль взаимодействия ПЛИС с сетью Ethernet.

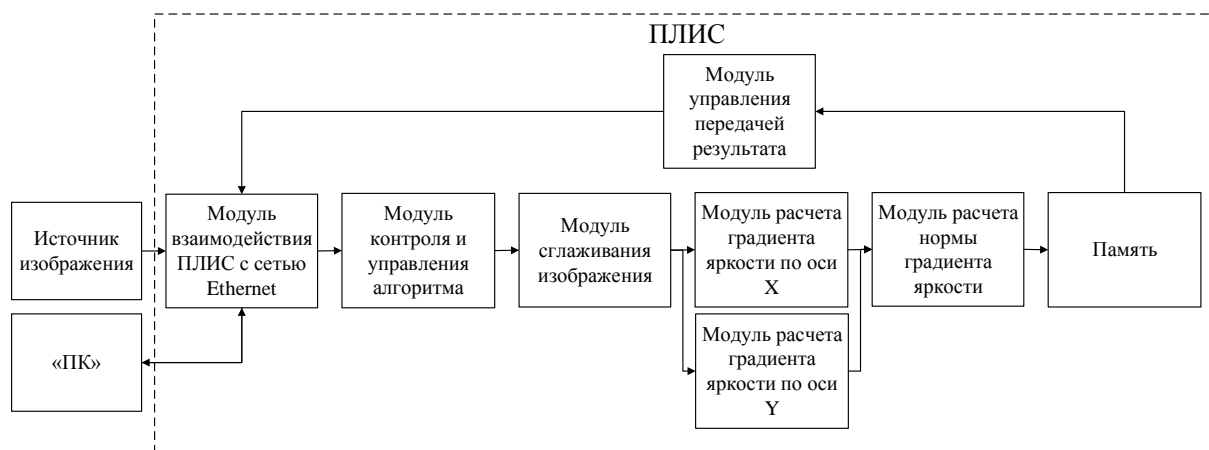


Рис. 5. Функциональная схема реализации алгоритма определения граничных элементов Canny

Для сравнения времени расчета гистограммы и времени определения граничных элементов на изображении СТЗ при реализации на процессоре Intel Quad Core i7 2.5GHz и на ПЛИС Cyclone IV GX EP4CGX150DF31 был проведен эксперимент.

Исходные данные эксперимента:

- 3 черно-белых изображения;
- размер изображений 640x480 пикселей.

Универсальный процессор:

- процессор Intel Quad Core i7 с тактовой частотой 2.4 Гц;
- оперативная память 16 Гб;
- ОС Windows 8.1.

ПЛИС:

- Altera Cyclone IV GX EP4CGX150DF31;
- 150 тыс. логических элементов;
- частота задающего генератора 125 МГц;
- внутренняя память 67,5 КБ.

Методика сравнения:

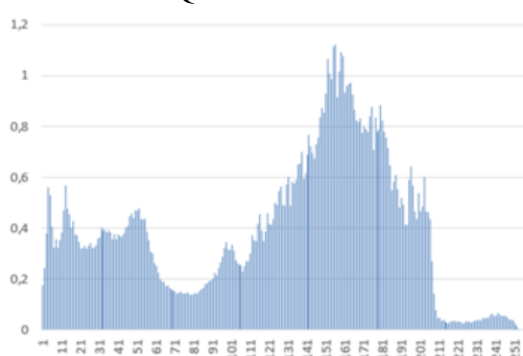
- изображение вводится в СТЗ через интерфейс Gigabit Ethernet. Это может быть изображение с камеры или отправленное по сети из другого источника;
- учитывается только время обработки. Разница времени с момента расположения изображения в памяти и времени появления результата считается временем обработки.

Исходные изображения, используемые в экспериментах, и результаты их обработки представлены на рис. 6, 7 и 8.

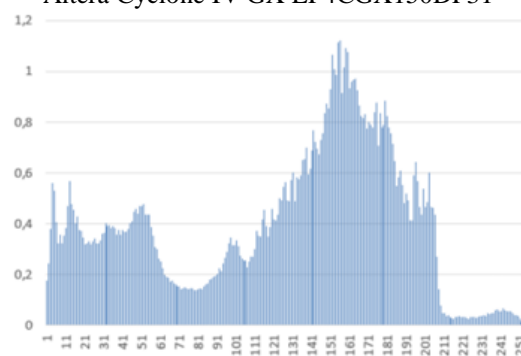
Исходное
изображение



Реализация на универсальном процессоре
Intel Quad Core i7 2.5GHz



Реализация на ПЛИС
Altera Cyclone IV GX EP4CGX150DF31



Результат
расчёта
гистограммы

Результат
выделения
элементов
границы
(алгоритмом
Canny)



Рис. 6. Исходное изображение №1 (сверху) и результаты расчета первого эксперимента (по центру) и второго эксперимента (снизу)

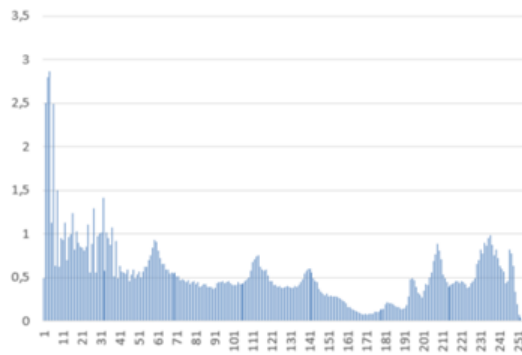
По результатам описанных экспериментов:

- качественно (визуальным контролем) и количественно (вычитанием результирующих изображений одного из другого) результаты работы обеих систем идентичны (~ 0,44% несовпадающих значений);
- результаты сравнительного анализа по среднему времени обработки в двух экспериментах с разными изображениями приведены в таблице 1 (реализация алгоритма расчета гистограммы) и таблице 2 (реализация алгоритма определения элементов границ Canny).

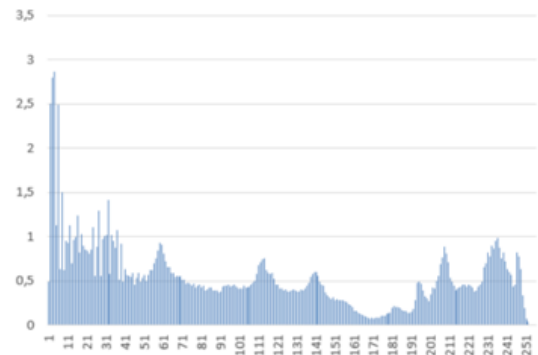
Исходное
изображение



Реализация на универсальном процессоре
Intel Quad Core i7 2.5GHz



Реализация на ПЛИС
Altera Cyclone IV GX EP4CGX150DF31



Результат
расчёта
гистограммы

Результат
выделения
элементов
границы
(алгоритмом
Canny)



Рис. 7. Исходное изображение №2 (сверху) и результаты расчета в первом эксперименте (по центру) и втором эксперименте (снизу)

В ходе выполнения описанных экспериментов были выделены следующие особенности реализации алгоритмов обработки зрительных данных на базе универсального процессора и ПЛИС.

Реализация на универсальном процессоре Intel Quad Core i7 2.5GHz:

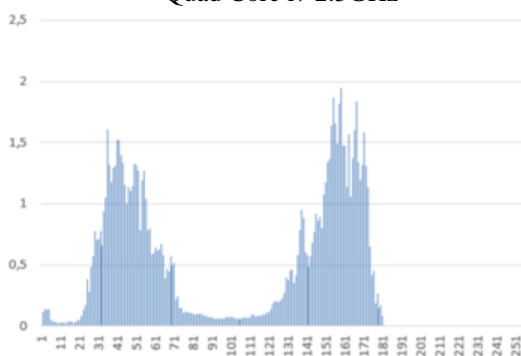
- Простота реализации. Алгоритм описан на языке C++, что обеспечивает ему переносимость на другие универсальные процессоры.

- Трудозатраты реализации. Для реализации данных алгоритмов потребовалось 50 и 300 строк исходного кода на языке C++.
- Возможно использование алгоритмов обработки из большого количества библиотек даже программистами без большого опыта работы.
- Важным фактором, влияющим на производительность обработки, является ОС под управлением которой работает процессор.
- Время обработки прямо пропорционально количеству точек обрабатываемого изображения.

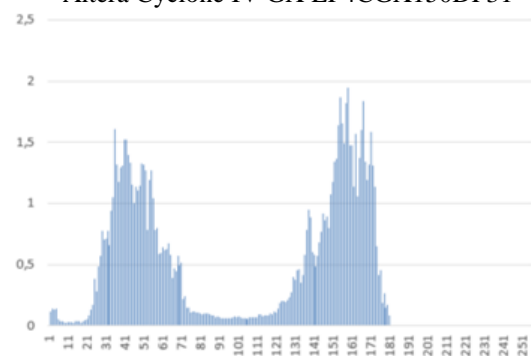
Исходное
изображение



Реализация на универсальном процессоре Intel
Quad Core i7 2.5GHz



Реализация на ПЛИС
Altera Cyclone IV GX EP4CGX150DF31



Результат
расчёта
гистограммы

Результат
выделения
элементов
границы
(алгоритмом
Canny)

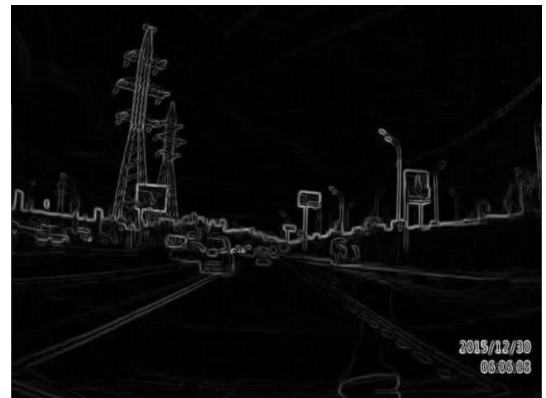


Рис. 8. Исходное изображение №3 (сверху) и результаты расчета в первом эксперименте (по центру) и втором эксперименте (снизу)

Таблица 1

**Сравнительные результаты реализации алгоритма построения
гистограммы**

Изображение	Процессор Intel Quad Core i7 2.5GHz	ПЛИС EP4CGX150DF31
	Время обработки, мкс	
1	328	95
2	293	121
3	329	82
Ср. знач. (по результатам обработки 3 изображений)	317	99

Таблица 2

Сравнительные результаты реализации алгоритма Canny

Изображение	Процессор Intel Quad Core i7 2.5GHz	ПЛИС EP4CGX150DF31
	Время обработки, мкс	
1	923,8	168,5
2	999,2	152,6
3	982,1	183,3
Ср. знач. (по результатам обработки 3 изображений)	968,4	168,1

Реализация на ПЛИС Altera Cyclone IV GX EP4CGX150DF31, частота задающего генератора 125 МГц.

- Время обработки. Сокращается в 3 и более раз.
- Алгоритм описан на языке Verilog, что обеспечивает ему переносимость на другие линейки ПЛИС и на ПЛИС других производителей.
- Обязательное наличие навыков программирования ПЛИС. Желательным является получение дополнительного образования в области проектирования и программирования систем обработки сигналов на базе ПЛИС.
- Высокие трудозатраты для реализации алгоритма. Для реализации данных алгоритмов потребовалось 880 и 2220 строк исходного кода на языке Verilog.

Анализ результатов экспериментов позволяет сделать следующие выводы:

- Реализация алгоритмов первичной обработки на ПЛИС позволяет в разы сократить время первичной обработки информации.

- Переход к реализации СТЗ полностью на базе ПЛИС нецелесообразен. Перспективно объединение многоядерного процессора с ПЛИС на основе быстродействующего интерфейса.
- Невозможен быстрый переход от разработки СТЗ на базе универсального процессора к реализации на ПЛИС ввиду необходимости переобучения специалистов.

Заключение

В работе описан первый шаг на пути интеграции ПЛИС в СТЗ РВ. Получены практические результаты, подтверждающие целесообразность использования ПЛИС для первичной обработки зрительных данных.

На последующих этапах планируется выполнить показательную задачу по сбору и анализу зрительных данных с помощью СТЗ реального времени на базе гетерогенной аппаратной архитектуры, включающей унифицированный многоядерный процессор и одну или несколько ПЛИС.

Кроме того, планируется наполнить библиотеку алгоритмов обработки зрительных данных на ПЛИС и провести сравнительный анализ эффективности использования в составе средств аппаратной поддержки СТЗ ПЛИС и графических процессоров (GPU) [13].

Библиографический список

1. Богуславский А.А., Соколов С.М. Программно-аппаратные средства ввода зрительных данных в память персонального компьютера // Препринты ИПМ им. М.В.Келдыша. — 2002. — № 52. — 22 с. — URL: <http://library.keldysh.ru/preprint.asp?id=2002-52>
2. Домунян А. А. Программно-аппаратная реализация кортексного сканера // Техническое зрение в системах управления мобильными объектами— 2010: Труды научно-технической конференции-семинара. Вып. 4 / Под ред. Р.Р. Назирова.— М. : КДУ, 2011.— С. 310–316. ISBN 978-5-98227-794-7. — URL : <http://www.iki.rssi.ru/books/2011tz.pdf> (дата обращения: 21.11.2016).
3. Краснобаев А.А. Алгоритмы распознавания штриховых кодов // Препринты ИПМ им. М.В.Келдыша. — 2004. — № 84. — 29 с. — URL: <http://library.keldysh.ru/preprint.asp?id=2004-84>
4. Краснобаев А.А. Метод декомпозиции алгоритмов систем технического зрения на параллельно-конвейерное программно-аппаратное исполнение в архитектуре ПЛИС-ЦСП: Дис. канд. физико-математических наук. – М.: ИПМ им. М.В.Келдыша РАН, 2008. – 160 с.
5. Соколов С.М., Платонов А.К., Богуславский А.А., Трифонов О.В. Система технического зрения в составе биомехатронного тренажёра // Препринты ИПМ им. М.В.Келдыша. — 2014. — № 42. — 19 с. — URL: <http://library.keldysh.ru/preprint.asp?id=2014-42>

6. Leonardo Chang, José Hernández-Palancar, L. Enrique Sucar, Miguel Arias-Estrada. FPGA-based detection of SIFT interest keypoints // Machine Vision and Applications. February 2013, Volume 24, Issue 2, P. 371-392, [doi:10.1007/s00138-012-0430-8](https://doi.org/10.1007/s00138-012-0430-8)
7. OpenCV (Open source computer vision). URL: <http://opencv.org/> (дата обращения: 21.11.2016)
8. Rafael Rodriguez-Gomez, Enrique J. Fernandez-Sanchez, Javier Diaz, Eduardo Ros. Codebook hardware implementation on FPGA for background subtraction // Journal of Real-Time Image Processing March 2015, Volume 10, Issue 1, P. 43-57, [doi:10.1007/s11554-012-0249-6](https://doi.org/10.1007/s11554-012-0249-6)
9. Roberto Marzotto, Paul Zoratti, Daniele Bagni, Andrea Colombari, Vittorio Murino. A real-time versatile roadway path extraction and tracking on an FPGA platform // Volume 114, Issue 11, November 2010, P. 1164–1179, [doi:10.1016/j.cviu.2010.03.015](https://doi.org/10.1016/j.cviu.2010.03.015)
10. Wajdi Elhamzi, Julien Dubois, Johel Miteran, Mohamed Atri. An efficient low-cost FPGA implementation of a configurable motion estimation for H.264 video coding // Special Issue Journal of Real-Time Image Processing. March 2014, Volume 9, Issue 1, P. 19-30, [doi:10.1007/s11554-012-0274-5](https://doi.org/10.1007/s11554-012-0274-5)
11. Larry Matthies, Mark Maimone, Andrew Johnson, Yang Cheng, Reg Willson, Carlos Villalpando, Steve Goldberg, Andres Huertas. Computer Vision on Mars // International Journal of Computer Vision, 2007 Springer Science + Business Media, LLC. Manufactured in the United States, [doi:10.1007/s11263-007-0046-z](https://doi.org/10.1007/s11263-007-0046-z).
12. Стешенко В. Б., ПЛИС фирмы Altera: элементная база, система проектирования и языки описания аппаратуры. — М.: Издательский дом «Додэка-XXI», 2007, 576 с.
13. Васильев А.И. Программное и алгоритмическое обеспечение систем компьютерного видения с несколькими полями зрения: Дис. канд. физико-математических наук. – М.: ИПМ им. М.В.Келдыша РАН, 2013. – 116 с. — URL: <http://keldysh.ru/council/1/vasilev-diss.pdf>

Оглавление

Введение	3
Примеры использования ПЛИС в СТЗ	3
Архитектура ПЛИС	5
Структура СТЗ с использованием ПЛИС	7
Требования к ПО ПЛИС в СТЗ РВ	7
Эксперименты с алгоритмами первичной обработки изображений, реализованными на базе ПЛИС	8
Заключение.....	15
Библиографический список.....	15