

Технические науки

УДК 004.421

Молодяков Сергей Александрович

доктор технических наук, доцент,
профессор Санкт-Петербургского
политехнического университета Петра Великого

Molodyakov Sergey

doctor of engineering sciences, professor
Peter the Great Saint-Petersburg
Polytechnic University

Тышкевич Антон Игоревич

кандидат технических наук, доцент,
доцент Санкт-Петербургского
политехнического университета Петра Великого

Tyishkevich Anton

doctor of philosophy, docent
Peter the Great Saint-Petersburg
Polytechnic University

**ПРИНЦИПЫ ВЫДЕЛЕНИЯ ПАРАЛЛЕЛЬНЫХ ПОТОКОВ
КОМАНД ОБРАБОТКИ ВИДЕОИЗОБРАЖЕНИЙ В SMART-
ВИДЕОКАМЕРАХ**

**PRINCIPLES FOR THE ALLOCATION OF PARALLEL
INSTRUCTION STREAM VIDEO PROCESSING IN SMART- VIDEO
CAMERAS**

Аннотация: Рассматривается методология применения методов параллельной обработки данных в smart- видеокамерах. Предложен способ разбиения алгоритмов обработки видео данных на потоки команд, которые могут выполняться параллельно. Предложенный способ позволяет существенно сократить объем буферной памяти, необходимой для работы программ. Рассмотрены схемы организации буферов памяти при многопоточной обработке кадров с видеокамеры.

Ключевые слова: видеокамера, Smart-камера, видеоизображение, поток команд.

Summary: We consider the methodology of the use of parallel processing techniques in smart- cameras. A method for splitting video processing algorithms for data streams of commands that can be executed in parallel. The proposed method can significantly reduce the amount of buffer memory required to run programs. The schemes of the organization of the memory buffer in multi-threaded processing of frames from a video camera.

Key words: video camera, smart- video camera, instruction stream, video.

Введение

В настоящее время цифровая видеокамера становится неотъемлемым элементом повседневной жизни человека. Каждый ноутбук, каждый смартфон оснащен цифровой камерой [1]. Кроме того, видеокамеры применяют в разных специальных областях так, например, в оптоэлектронных приемных системах радиотелескопов [2, 3]. Основными направлениями совершенствования видеокамер являются: повышение их разрешения и расширение встраиваемых функций по обработке видео. Отсюда появляются требования по повышению производительности цифрового процессора видеокамер. Достигнут высокой производительности можно за счет применения методов параллельного выполнения команд (конвейер команд, векторизация, суперскалярность) и параллельного выполнения потоков команд (многопоточность). Особенно перспективным представляется применение многопоточности, которую можно реализовать в программируемой логике, на ядрах сопроцессоров и в многоядерных процессорах. Применение многопоточности связано с решением вопросов разбиения программы обработки видеокадров на потоки команд и с выделением буферов памяти для каждого из потоков [5].

Актуальность повышения производительности видеокамер связана и с широким внедрением smart-камер, которые могут соперничать с компьютером с точки зрения вычислительной мощности и

функциональности. Smart-камеры применяют в ряде проектов, например «умный дом», «стерео», «smart видеосеть» и др. Особенностью smart-камер является возможность загрузки собственных программ для специальных применений. В результате smart-камеру можно рассматривать в качестве специализированной камеры [5], функциональность которой за счет смены программного обеспечения можно настраивать на решаемые задачи.

В статье рассмотрены принципы выделения параллельных потоков команд обработки видеоизображений в smart-видеокамерах. Показана возможность использования параллельных потоков при минимальном объеме видеобуферов. Сделан первый шаг проектирования многопоточкового программного обеспечения видеокамер на стадии кадрового обеспечения.

Алгоритмы обработки в smart-видеокамерах

Smart-видеокамеру (интеллектуальную камеру) можно рассматривать как компактную архитектуру системы технического зрения (рис. 1). Smart-камеры, как правило, состоят из нескольких (но не обязательно все) компонент: фотоприемник (матрица или линейный, ПЗС или КМОП), кодек для оцифровки изображения, видеопроцессор, интерфейсы связи (RS-232, Ethernet, Camera Link, др.), встраиваемое и внешнее программное обеспечение [5].

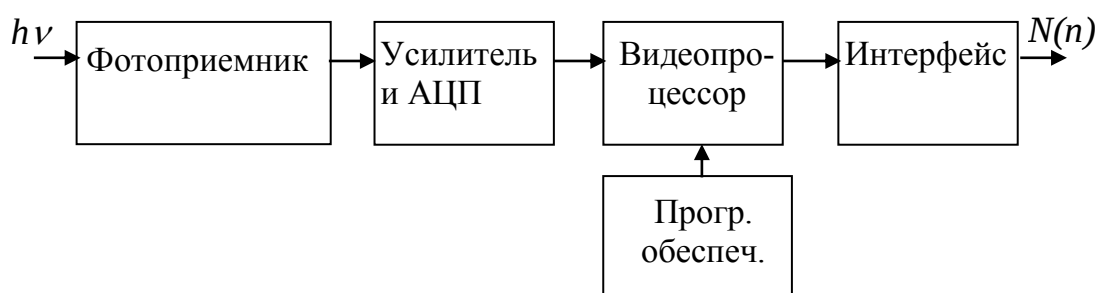


Рис. 1. Схема интеллектуальной камеры (разработка авторов).

Основная цепочка алгоритмов, предназначенных для улучшения качества видеоизображений, представлена в табл. 1. Алгоритмы разделены на стадии применения и могут быть реализованы как в аналоговом, так и

цифровом виде. Разделение на стадии применения является условным и определяется размером того элемента, с которым работает алгоритм. Так при попиксельной обработке вычисления происходят при работе с одним пикселем, а при покадровой обработке в вычислениях используются пиксели всего кадра.

Таблица. 1. Алгоритмы обработки изображений в цифровых камерах.

Стадия применения	Название алгоритмов, используемых в цифровых видеосистемах
Управление фотоприемником	Автоматическая экспозиция (Auto exposure AE). Автобаланс белого (Auto white balance AWB). Детектирование плавления сигнала (Flicker avoid).
Попиксельная обработка	Компенсация черного (Auto black reference ABR). Программирование аналогового усиления. Программирование времени экспозиции. Выбор разрядности АЦП. Контроль насыщения. Преобразование Bayer RGB.
Построковая обработка	Интерполяция цвета. Увеличение/уменьшение. Коррекция дефектов. Коррекция цвета. Гамма коррекция. Компенсация искажений в линзе. Объединение отсчетов (Binning). Подавление шумов.
Покадровая обработка	Контроль резкости (Sharpness). Распознавание. Подчеркивание границ. Изменение формата, сжатие.
Управление режимом	Регистрация одного кадра. Потокное видео. Переключение памяти данных (flash).

([4, с.35])

Разбиение на потоки команд

Разбиение на потоки команд можно провести на основе разделения алгоритмов на стадии или уровни применения (табл. 1). На рис. 2 показан практически полный набор потоков, которые могут исполняться параллельно в фотоприемниках (ФП) [6] и в вычислительных ядрах устройств. В схеме отмечены особенности синхронизации потоков. Причем определены два режима синхронизации: жесткая и мягкая синхронизации. При жесткой синхронизации обработка пикселей должна осуществляться сразу же при их появлении. При мягкой синхронизации

обработка пикселей может осуществляться с задержкой, но время обработки ограничено появлением следующей порции пикселей.



Рис. 2. Схема многоуровневых алгоритмов, потоков команд и их реализации [4, с.176].

Первый самый высокоуровневый поток поддерживает работу с пользователем камеры или с оператором. Этот поток команд осуществляет отображение видео. Может поддерживаться мягкая синхронизация, например, по кадровым сигналам или по часам времени. Поток II осуществляет передачу обработанных кадров на этап отображения. При передаче кадров из буфера обработки в буфер отображения возможно осуществить дополнительную обработку или преобразование формата данных. Поддерживается жесткая внутренняя синхронизация.

Команды потока III осуществляют покадровую обработку. Для выполнения алгоритмов необходимо читать данные всего кадра. Начало работы потока осуществляется импульсами периода обрабатываемых сигналов (мягкая синхронизация). Команды потока IV осуществляют

построковую обработку. Алгоритмы работают с одной строкой кадра изображения. Мягкая синхронизация проводится с использованием строковых импульсов. Команды потока V проводят попиксельную обработку, т.е. для их выполнения необходимо иметь буфер на один или несколько ближайших пикселей. Поток синхронизируется тактовым генератором управления ФП. Синхронизацию можно считать условно мягкой. Самый низкоуровневый поток – это поток VI. Здесь осуществляется управление фотоприемником и чтение отсчета. Синхронизация выполнения команд безусловно жесткая, связанная с тактовым генератором ФП. Кроме чтения отсчетов в этом потоке могут выполняться простые алгоритмы обработки: автобаланс белого, ДКВ и др. [7, 8]. Перечисленные потоки команд могут быть выполнены в разных элементах камеры.

Буферы видеоданных

Организация буферов данных при параллельном выполнении потоков команд обработки видео имеет во многом определяющее значение для достижения высоких скоростей обработки. При создании «лишних» буферов или при перекрестной работе с данными при недостаточном количестве буферов значительно возрастают временные затраты. Дополнительные временные затраты могут быть связаны с ненужным переписыванием данных из одного видео буфера в другой или с появлением зависимостей по данным. Зависимость по данным возникает тогда, когда при параллельно работающих потоках команд происходит регулярное обращение к одним и тем же данным из разных потоков. Указанную зависимость можно преодолеть двумя путями. Во-первых, для каждого вычислительного ядра можно создавать свой буфер данных, с которыми происходит работа текущего алгоритма. Во-вторых, можно иметь общий буфер, например, на кадр пикселей, и с ним работать со стороны всех имеющихся ядер, используя при этом команды

синхронизации по аналогии с командами библиотеки OpenMP (команды замков lock). В целом, выделение необходимых буферов можно рассматривать в качестве элемента организации потоков команд.

Рассмотрим вопрос, как размещать и использовать буферы данных. На рис. 3 представлена схема буферов при многопоточной обработке с разделением потоков в соответствии с рис. 2.

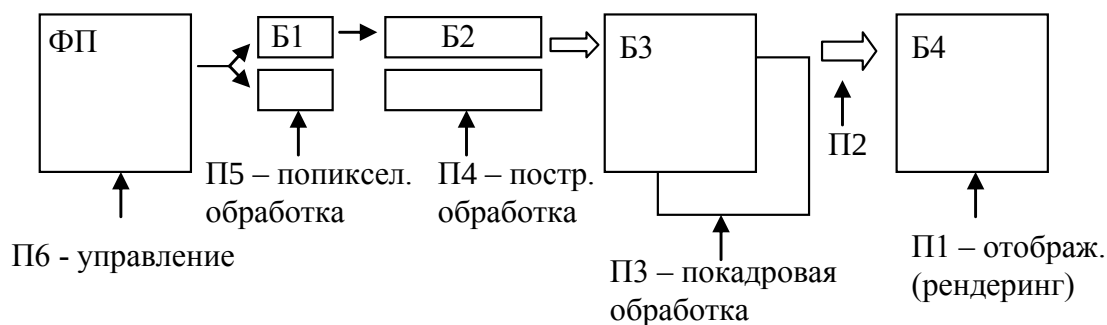


Рис. 3. Схема буферов при многопоточной обработке данных видео камеры (разработка авторов).

В схеме представлены все потоки команд и буферы для их выполнения. Показано, что для попиксельной и для построкой обработки размер буферов минимальный – на несколько пикселей и строку пикселей соответственно. Основные буферы хранения пикселей продублированы. Это необходимо для разнесения операций работы с памятью. В один буфер пишем, например, строку пикселей, а во втором обрабатываем и передаем. Затем происходит переключение буферов. Время выполнения, например, построковых алгоритмов ограничено периодом следования строк. Для увеличения времени, выделенного для алгоритма, количество буферов должно быть соответственно также увеличено.

Предложенная схема разделения алгоритмов и команд на потоки позволяет существенно сократить объем буферной памяти, необходимой для работы программ. Данную схему можно использовать, например, при проектировании видеокамеры на кристалле.

Разбиение на потоки команд при кадровой обработке

Наиболее широкое распространение получили кадровые видеокамеры, в которых единицей информации, получаемой из камеры, является кадр пикселей [1, 5, 9]. В этом случае схема буферов при многопоточной обработке данных может выглядеть в соответствии с рис. 4. Кадры из камеры один за другим поступают в буфер Б1 с участием процессорного ядра 1. Затем они переписываются в буфер Б2, с которым работает процессорное ядро 2, выполняя поток команд П5. Затем кадр, обработанных на первом этапе пикселей, переписываются в буфер Б3, с которым начинает работать процессорное ядро 3. Процесс перепись – обработка может выполняться дальше. Количество этапов и соответственно буферов зависит от количества имеющихся ядер и используемых алгоритмов обработки. В конце обработки (из 4-го ядра) данные передаются на отображение.

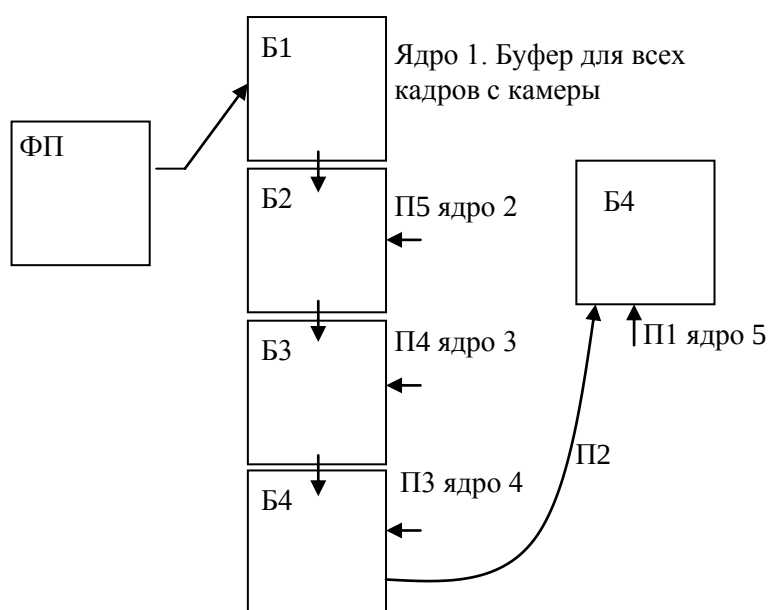


Рис. 4. Схема буферов при многопоточной обработке данных кадровой видеокамеры (разработка авторов).

Представленная схема (рис. 4) является примером реализации конвейерного механизма обработки кадров. Схема может быть применена как с использованием команд OpenMP, так и команд MPI. Ограничением

является необходимость использования операций по переписи кадров из одного буфера в другой, что существенно уменьшает время, выделяемое на выполнение команд алгоритмов обработки. Для сокращения указанного времени можно предложить использование команд строковой передачи, которые можно включить в программу с использованием встроенного ассемблера.

Можно предложить схему (рис. 5) конвейерной обработки видео, которая не требует переписи кадров из буфера в буфер. В этом случае видеокамера должна переключать адреса записи кадров в память. Каждый следующий кадр будет находиться в новом буфере.

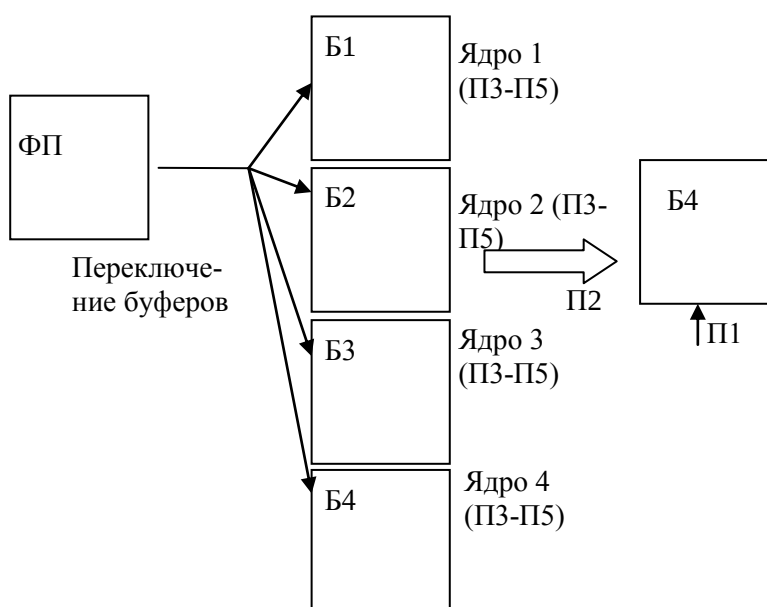


Рис. 5. Схема буферов при многопоточной обработке данных видеокамеры с переключением адреса записи кадров (разработка авторов).

На рис. 5 показано 4-е буфера. Пятый кадр будет записан снова в первый буфер. Таким образом, время на работу совокупного алгоритма обработки будет равно времени следования 4-х кадров. В представленной схеме каждое ядро обработки выполняет полный набор алгоритмов обработки видео. Разбиение на потоки в этих ядрах может отсутствовать. В данной схеме можно ожидать большей производительности и за счет более эффективного использования кэш-памяти в ядрах обработки.

Применение рассмотренных принципов параллельной обработки видео определяется выбранной аппаратной платформой камеры.

Заключение

Успешное проектирование smart-камер связано как с разработкой их аппаратного, так и программного обеспечения. Высокоскоростные, функционально насыщенные smart-камеры требуют применения высокопроизводительных многоядерных процессоров обработки. Предложена методология разбиения алгоритмов обработки видео данных на потоки команд, которые могут выполняться параллельно в вычислительных ядрах. Рассмотрены схемы организации буферов памяти, которые позволяют существенно сократить объем памяти, необходимой для работы программ при многопоточной обработке в видеокамерах или в видеосистемах.

Список литературы

1. Holst G.C., Lomheim T.S. CMOS/CCD Sensors and Camera Systems. SPIE Press, 2007. 376p.
2. Лавров А.П., Молодяков С.А., Саенко И.И. Акустооптические процессоры в радиоастрономических приемниках // Антенны. 2009. № 7. С. 45–55.
3. Лавров А.П. Молодяков С.А. Оптоэлектронный процессор для регистрации радиоизлучения пульсаров // Приборы и техника эксперимента. 2015. № 1. С. 136–145. Lavrov A.P, Molodyakov S.A. An optoelectronic processor for registration of radiation of pulsars // Instruments and Experimental Techniques. 2015. V. 58. № 1. P. 132–140
4. Молодяков С.А. Системное проектирование оптоэлектронных процессоров обработки сигналов. СПб.: Изд-во Политехн. ун-та, 2011.- 226 с., ISBN 978-5-7422-3016-8
5. Молодяков С.А. Проектирование специализированных цифровых видеокамер. / СПб.: Изд-во Политехн. ун-та, 2016 .— 286 с., ISBN 978-

5-7422-5334-1

6. Молодяков С.А. Применение ПЗС-фотоприемников для предварительной обработки сигналов //Датчики и системы. 2014.- № 5 (180).- С. 2-10.
7. Молодяков С.А. Особенности и алгоритмы цифровой обработки сигналов в оптоэлектронных процессорах // Цифровая обработка сигналов. 2013. № 3. С. 61–66.
8. Молодяков С.А. Фотоприемники в системах потоковой обработки сигналов и изображений. СПб.: Изд-во Политехн. ун-та, 2014 .- 134 с. ISBN 978-5-7422-4588-9.
9. Есепкина Н.А., Гаврилов Г.А., Молодяков С.А. и др. Оптоэлектронный процессор на основе матричного ФПЗС с волоконной шайбой. // Письма в ЖТФ. -1992. -т.18. -№3. -с.32-37.