

Применение видеodeкодеров в мониторинге и наблюдении

Витольд КАЦУРБА
(Witold KACZURBA)
witold.kaczurba@analog.com
Перевод: Николай ИСАКОВ
nikolay.isakov@analog.com.ru

Сегодня системы видеонаблюдения получили широкое распространение. Компании предлагают недорогие, с низким разрешением черно-белые камеры CCTV и цветные камеры высокой четкости. Они используются в различных приложениях: от наблюдения за производственным процессом до контроля трафика и распознавания лиц в реальном времени.

Видео изначально несет в себе много лишних данных, которые могут осложнить задачи обработки и хранения информации. Но видеонаблюдение можно упростить за счет удаления бесполезной информации и пропуска только важных частей изображения, что уменьшает потребление как памяти, так и вычислительных ресурсов. На рис. 1 показаны элементы типичной системы.

В статье приведено несколько примеров того, как извлечение полезных данных может свести к минимуму обработку, потребление памяти и использование ресурсов DSP-процессора, а также показано, как особенности видеodeкодеров фирмы Analog Devices могут способствовать упрощению алгоритмов обработки видео и ускорению разработки систем видеонаблюдения.

Пример 1. Подсчет и инспектирование объектов

Представьте себе широкий транспортер, на котором находится много быстро движущихся объектов. Большое количество изделий делает трудным ручной подсчет. Выполняя задачу автоматизации подсчета, камеры могут быть использованы и для контроля качества продукции: это можно сделать путем модификации простого алгоритма подсчета количества предметов для анализа конкретных деталей и артефактов.

Хранение всех видеоданных требует огромного количества памяти, и обработка больших объемов данных будет стоить дорого, если учесть использование аппаратных средств и энергопотребление. Вместо сбора всех изображений в памяти система может находить интересные детали в массиве данных и выбрасывать как можно больше бесполезных данных.

В большинстве случаев изображения в градациях серого несут достаточно информации. Поэтому информация о цветности может быть отброшена путем преобразования RGB-сигнала в Y (только яркость). Таким образом, предмет на транспортере может быть проверен путем

сравнения его монохромного изображения с шаблоном для определения деформаций за счет применения алгоритмов обнаружения краев.

Такие алгоритмы требуют всего лишь несколько строк активной видеoinформации и небольшого объема памяти. С их помощью находят разрывы в яркости смежных пикселей путем вычисления первой и второй производных яркости изображения [1]. Краевые алгоритмы обнаружения могут быть реализованы на практике путем извлечения информации с помощью матричных вычислений, таких как матричный оператор Собеля (Sobel operator описан, например, в энциклопедии Wikipedia). При реализации данного алгоритма на FPGA (программируемых логических интегральных схемах) на уровне пикселей это дает удовлетворительные результаты. Простая реализация алгоритма на FPGA описана в [2]. Шум может быть удален за счет двумерной гауссовской фильтрации, как это описано в статье [3]: здесь приведена успешная реализация детектора, подобного краевому детектору Канни [4].

Существуют другие алгоритмы оптимизации, которые могут повысить качество изображения, но все они требуют значительных ресурсов FPGA. Однако некоторые интегральные схемы (IC) видеodeкодеров уже имеют встроенные средства реализации алгоритмов предварительной обработки или фильтрации сигнала, поэтому выбор одного из них позволит сэкономить ресурсы FPGA. Например, видеodeкодер ADV7802 включает в себя блоки улучшения яркостных переходов (LTI) и улучшения цветных переходов (CTI). Эти блоки, которые улучшают получаемое изображение за счет увеличения крутизны переходов яркости и цветности, используют адаптивные методы обнаружения перехода и различные нелинейные алгоритмы; они не увеличивают уровень шумов, не вводят артефактов и могут быть очень полезны при обнаружении краев. Кроме того, блоки улучшения переходов и другие встроенные фильтры могут удалять высокочастотный шум источника, то есть они обрабатывают только полезный сигнал и игнорируют случайный шум.

Краевой алгоритм предоставляет информацию о границах объекта, а не полную картину объекта. При этом происходит уменьшение требуемой памяти от 24 бит на пиксель (3×8 бит) до всего лишь одного бита на пиксель, и это экономит много памяти:

- 640×480 пикселей = 307 200 бит;
- 800×600 пикселей = 480 000 бит;
- 1024×768 пикселей = 786 432 бита;
- 1280×720 пикселей = 921 600 бит.

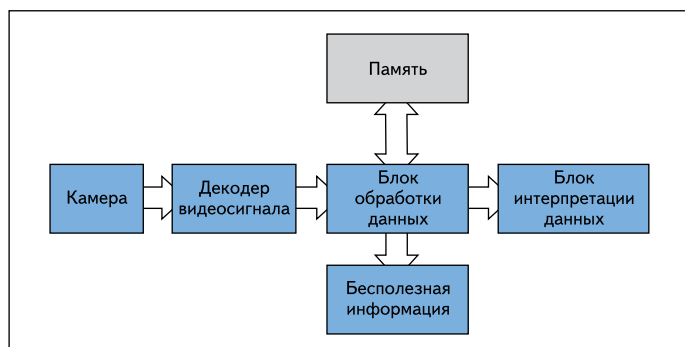


Рис. 1. Упрощенная диаграмма видеонаблюдения

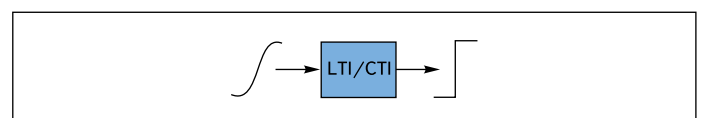


Рис. 2. Работа блока LTI/CTI

Путем преобразования RGB в Y, храня всего несколько строк активного видео в памяти и используя алгоритмы на FPGA, мы можем обнаруживать предметы и видеть их формы. После того как расположение объектов на транспортере стало известно, мы можем оценить их движение и собирать цветность или другую информацию со следующего кадра с гарантией того, что используется минимальный объем памяти. Этот процесс включает:

1. Обнаружение краев.
2. Хранение информации.
3. Предсказание следующей позиции X_{N+1} .
4. Извлечение информации в области, где предсказано расположение объекта.

Пример 2. Обнаружение движения и качества продукции

Робот ищет предметы на определенном расстоянии и в ограниченном пространстве. В некоторых случаях может быть использован ультразвук, но если поверхность предмета поглощает ультразвук или предметы находятся за стеклом, можно использовать видео. Камера фокусируется на близлежащие объекты. В пределах некоторого диапазона расстояний предметы будут иметь резкие края, а те, что находятся за пределами этого диапазона, будут видны нечетко (рис. 3).



Рис. 3. Глубина резкости

Алгоритмы обнаружения краев могут быть использованы для ограничения предметов в пределах целевого диапазона, так как они представляют собой объекты с четкой границей. Предметы на фоне не будут настолько четкими, чтобы краевой алгоритм определил их. В результате обработки получают однобитное растровое изображение, где 1 означает, что край был обнаружен, а 0 — что край не был обнаружен. Положение (x, y) каждого пикселя, указывающего на обнаруженный край, может быть использовано для нахождения середины изолированного объекта:

$$x_{\text{среднее}} = \frac{\sum_{n=1}^N x_n}{N}, \quad y_{\text{среднее}} = \frac{\sum_{n=1}^N y_n}{N}, \quad (1)$$

где x_n — это координата x пикселя с номером n , соответствующего краю предмета; y_n — это координата y пикселя с номером n , соответствующего краю предмета; N — количество выявленных пикселей края.

После того как положение объекта и его края определено, мы можем попытаться проследить его движение. Главное здесь — извлечь только один объект из изображения. Преобразуя его края в контур и вычисляя среднее расстояние в пикселях от середины объекта до его краев, мы можем убедиться, что размер объекта изменяется, и на основе этого сделать заключение, движется ли объект по направлению к камере. Расчет производится по формуле

$$\frac{1}{N} \sum_{n=1}^N \left(\sqrt{(x_n^{\text{FRAME}} - x_{\text{среднее}}^{\text{FRAME}})^2 + (y_n^{\text{FRAME}} - y_{\text{среднее}}^{\text{FRAME}})^2} \right) - \frac{1}{M} \sum_{m=1}^M \left(\sqrt{(x_m^{\text{FRAME-1}} - x_{\text{среднее}}^{\text{FRAME-1}})^2 + (y_m^{\text{FRAME-1}} - y_{\text{среднее}}^{\text{FRAME-1}})^2} \right), \quad (2)$$

где N — это количество «краевых» пикселей в текущем кадре; M — количество «краевых» пикселей в предыдущем кадре (FRAME-1).

Если для простоты ограничиться только горизонтальной осью, уравнение примет следующий вид

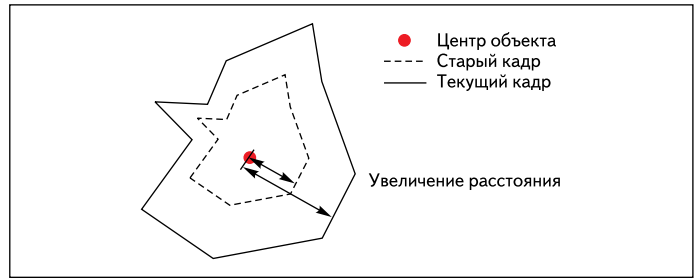


Рис. 4. Изменение контура движущегося объекта

$$\frac{1}{N} \sum_{n=1}^N (x_n^{\text{FRAME}} - x_{\text{среднее}}^{\text{FRAME}}) - \frac{1}{M} \sum_{m=1}^M (x_m^{\text{FRAME-1}} - x_{\text{среднее}}^{\text{FRAME-1}}). \quad (3)$$

Значение этого уравнения будет положительным, если объект приближается к камере (расстояние от середины объекта до краев увеличивается). Отрицательное означает, что объект удаляется от камеры. На рис. 4 показан пример, когда предмет движется к камере.

Заметим, что объект должен находиться в пределах диапазона резкости камеры. Мы можем модифицировать алгоритм так, чтобы динамически изменить фокусное расстояние для сканирования более широкой области. После того как объекты обнаружены, мы можем разделить изображение на сегменты, обрабатывать его и отслеживать перемещение.

Отслеживание объектов становится все более трудным делом по мере увеличения сложности видеоизображения, особенно если объекты покрыты текстурой или теряют четкость из-за быстрого движения. Некоторые алгоритмы слежения показаны в [5]. Как только предметы теряют четкость, алгоритмы обнаружения краев перестают работать. Отслеживание все еще можно реализовать с помощью сложных корреляционных методов, например, таких, как метод совпадения блоков (используемый для оценки движения), или других методов, подробно изложенных в [6].

Благодаря наличию непрерывного потока данных с камеры, объект можно отслеживать, чтобы определить его ускорение и другие параметры. Тем не менее, для получения хороших результатов анализа изображения нужно иметь высококачественное видео. При обнаружении краев путем анализа соседних пикселей разрешающая способность будет выше, если применять прогрессивную развертку вместо некачественных чересстрочных сигналов PAL или NTSC. Видеодекодеры ADV7401 и ADV7403 работают с сигналами различных видеостандартов, в том числе с прогрессивной разверткой. Благодаря возможности оцифровки видеосигналов до 140 МГц, они могут обрабатывать компонентные сигналы SD, ED и HD, а также CVBS и графику. Кроме того, они поддерживают нестандартные режимы видео, что позволяет работать с менее популярными стандартами, такими как STANAG. Многорежимная выходная пиксельная шина работает с форматами 4:2:2, 4:4:4 YCbCr или 4:4:4 RGB. Сигналы нестандартных видеоформатов могут быть интерполированы или децимированы для получения определенного размера по горизонтали, как описано в руководстве по применению AN-0978 [7].

Встроенный преобразователь цветового пространства (CSC), как показано на рис. 5, преобразует цветовое пространство в соответствии с требованиями пользователя (4). Сигналы YPrPb или RGB могут быть преобразованы в другие форматы, с помощью настраиваемых матриц преобразования. Например, преобразование RGB в YCbCr позволяет удалить информацию о цветности (Cb, Cr), чтобы упростить обнаружение края на монохромном изображении

$$\begin{bmatrix} \text{Chanel_A_out} \\ \text{Chanel_B_out} \\ \text{Chanel_C_out} \end{bmatrix} = \begin{bmatrix} A1 & A2 & A3 \\ B3 & B1 & B2 \\ C2 & C3 & C1 \end{bmatrix} \times \begin{bmatrix} \text{Chanel_A_in} \\ \text{Chanel_B_in} \\ \text{Chanel_C_in} \end{bmatrix} + \begin{bmatrix} A4 \\ B4 \\ C4 \end{bmatrix}, \quad (4)$$

где $A1$ – $A4$, $B1$ – $B4$, $C1$ – $C4$ — заданные параметры CSC.

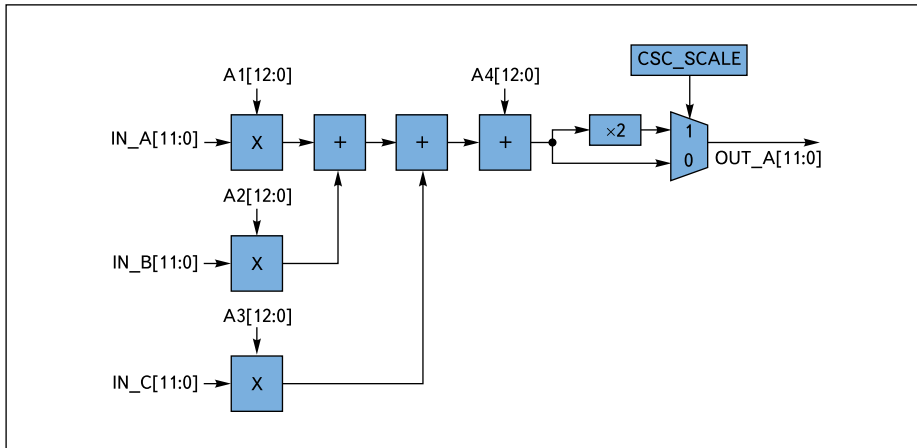


Рис. 5. Блок-схема одного канала CSC (ADV7403)

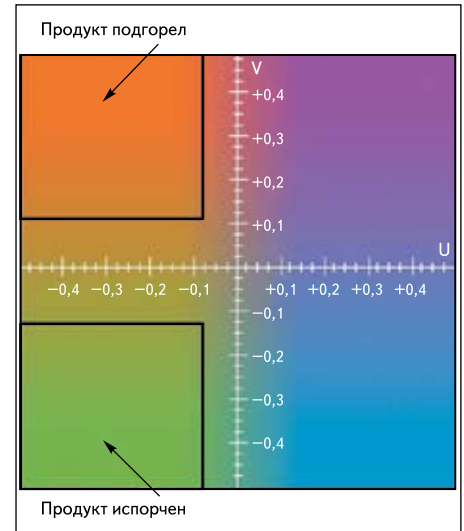


Рис. 6. Пример использования пространства для оценки качества некоего продукта

Преобразователь цветного пространства CSC — очень полезный функциональный блок. Получив входной сигнал RGB или YCbCr, его можно преобразовать в любой формат. На рис. 6 показано цветное пространство YUV, подобное пространству YCbCr.

На рис. 6 показано, как по цвету продукта (по значению YUV) можно определить его качество (например, подгоревший или испорченный). Преобразование цветного пространства необходимо в обработке видео и для взаимодействия с интегральными схемами, в которых используются другие стандарты. В ADV7401/ADV7403 имеется входной мультиплексор, который позволяет легко и быстро переключать источники видеосигнала: это полезно, например, при переключении системы видеонаблюдения с остановленного конвейера на работающий.

Пример 3. Настройка баланса белого и цветности для видеонаблюдения

При разработке видеосистемы, которая выделяет объекты из изображения, необходимы особые усилия, так как небольшие изменения угла падения света или его интенсивности могут влиять на результаты. Инженеры могут использовать имеющиеся в ADV7401/ADV7403 функции настройки коэффициентов усиления и смещения для регулировки яркости и контрастности, если поместят две небольших вспомогательных (реперных) полоски (одну темную, другую — светлую) на ленточный конвейер (рис. 7). Смещение и усиление в ADV7401/ADV7403 корректируются таким образом, чтобы система смогла компенсировать изменения цвета освещения, угла падения света и его интенсивности.

Алгоритм регулировки надлежащего баланса белого может быть очень простым. Во-первых, нужно получить опорные значения RGB (или YCbCr) для каждой из полосок. Затем, чтобы компенсировать изменения освещения, изменить смещение и усиление

в соответствии с величинами, полученными на реперных полосках. Можно использовать такой алгоритм:

1. Получить значение RGB (или YCbCr) темной полосы.
2. Отрегулировать смещение в соответствии с желаемым значением RGB (или YCbCr) темной полосы.
3. Получить значение RGB (или YCbCr) светлой полосы.
4. Отрегулировать усиление в соответствии с желаемым значением RGB (или YCbCr) светлой полосы.
5. Для повышения точности повторите шаги 2 и 4.

Такая процедура особенно полезна при разработке системы, поскольку она обеспечивает правильное смещение (яркость) и усиление (контраст), даже когда свет является слишком сильным или слишком слабым, как это показано на рис. 8. Регистры смещения и усиления доступны посредством шины I²C, что позволяет быстро адаптироваться к изменению внешних условий.

Аналогичным образом цветные опорные полоски могут быть использованы для настройки цвета. Алгоритм этой компенсации аналогичен настройке баланса белого. Когда баланс белого настроен в со-

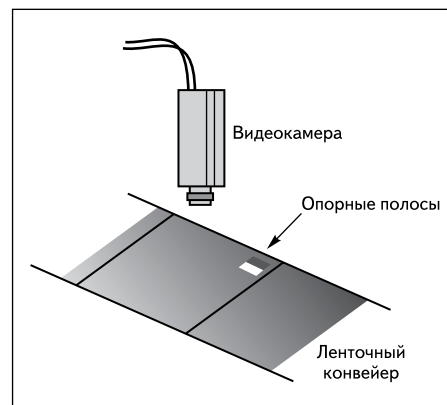


Рис. 7. В поле зрения камеры добавляются небольшие опорные полоски

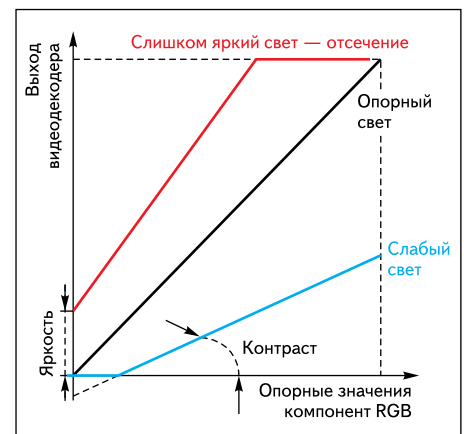


Рис. 8. Смещение и усиление регулируются для компенсации изменений в окружающем свете

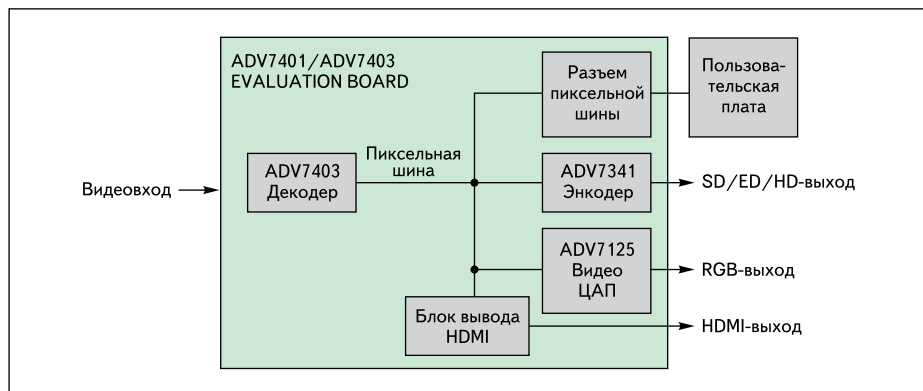


Рис. 9. Пиксельная шина оценочной платы ADV7401/ADV7403

Видеокодер, видео ЦАП и HDMI-передатчик AD9889B подключены к одной и той же пиксельной шине, что позволяет просматривать текущее изображение на втором выходе. Видеодекодеры Analog Devices включают блоки, необходимые для обработки видео, и обеспечивают защиту от сбоев и стабильность изображения.

Литература

1. Jähne B. Digital Image Processing. Springer-Verlag. 1997.
2. <http://www.oldcitypublishing.com/FullText/JAPEDfulltext/JAPED2.4fulltext/Abbasi.pdf>
3. <http://www.uweb.ucsb.edu/~shahnam/HAoEDAoF.pdf>
4. http://en.wikipedia.org/wiki/Canny_edge_detector
5. Shi J., Tomasi C. Good Features to Track. CVPR Proc. 1994.
6. Yao Wang, Ostermann J., Ya-Qin Zhang. Video Processing and Communications. Paperback. 2001.
7. www.analog.com