

УДК 004.932.4

АППАРАТНАЯ РЕАЛИЗАЦИЯ СЖАТИЯ ДИНАМИЧЕСКОГО ДИАПАЗОНА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ НА ПЛИС XILINX

Д. А. Гаврилов ¹, А. В. Павлов ², Д. Н. Щелкунов ¹

¹ Федеральное государственное автономное образовательное учреждение высшего образования "Московский физико-технический институт (государственный университет)", 141700, Долгопрудный, Институтский переулок, дом 9

² ОАО «Институт точной механики и вычислительной техники имени С.А. Лебедева РАН», 119991, Москва, Ленинский проспект, 51

Статья поступила в редакцию 28 сентября 2018 г.

Аннотация. Постановка задачи: технологии компьютерного зрения находят широкое применение во многих областях науки и промышленности. Задачами систем технического зрения являются получение, передача, воспроизведения и обработка визуальной информации без потери информативности. Алгоритмы и фильтры, используемые для предобработки, имеют целью подавление шумов, извлечение, а также сохранение информации необходимой для дальнейшей обработки. Цель работы: разработка модуля обработки цифровых изображений с аппаратной реализацией на программной логике на кристалле XC7Z020-2CLG400I семейства ZYNQ-7000 от Xilinx, позволяющей осуществлять сжатие HDR-изображения без потери детализации изображения и с высокой скоростью обработки. Результаты: описана аппаратная реализация алгоритма сжатия цифровых изображений. Выполнен анализ анизотропных сглаживающих фильтров для обработки изображений, рассмотрены недостатки и преимущества билатерального и управляемого фильтров, на основе чего осуществлен выбор фильтра для задач шумоподавления и сжатия динамического диапазона. Представлен алгоритм выбранного управляемого фильтра. Выполнена аппаратная реализация выбранного фильтра. Авторами проведена эффективная оптимизация аппаратной реализации управляемого фильтра. Практическая значимость: разработанное IP ядро использовано в модулях шумоподавления и сжатия динамического диапазона HDR-

изображения без потери детализации изображения телевизионной и тепловизионной камер.

Ключевые слова: обработка изображения, билатеральный фильтр, управляемый фильтр, HDR-изображение, сжатие HDR-изображения, интегральное представление изображения, оптимизация ресурсов.

Abstract. Introduction: technologies of computer vision are widely used in many fields of science and industry. The tasks of vision systems are the acquisition, transmission, reproduction and processing of visual information without loss of information. Algorithms and filters used for preprocessing are designed to suppress noise, retrieve, and store information necessary for further processing. **Purpose:** the development of a digital image processing module with hardware implementation on the Xilinx XC7Z020-2CLG400I crystal XC7Z020-2CLG400I chip Xilinx, which allows the HDR image to be compressed without loss of image detail and high processing speed. **Results:** In the presented work, the image processing system based on computer vision technology was developed. The analysis of filters for image processing is carried out, drawbacks and advantages of the bilateral and guided filters are considered, on the basis of which the filter is selected. The algorithm of the selected controlled filter is presented. The hardware implementation of the selected filter on the programmable logic is performed. The authors conducted an effective method for optimizing the guided filter using the integrated image representation and reducing its bit depth, which allowed to increase the efficiency of the filter and to compress the HDR image without loss of quality and with high processing speed. **Practical relevance:** The developed algorithm provides noise suppression and storage of information necessary for further processing.

Keywords: image processing, bilateral filter, guided filter, HDR image, HDR image compression, integrated image representation, resource optimization.

Введение

В настоящее время во многих сферах деятельности человека используется технологии компьютерного зрения. Компьютерное или техническое зрение находит широкое применение в различных областях науки и промышленности,

в том числе в медицине, космических исследованиях, компьютерной графике, новых военных технологиях и разработках. Задачами систем технического зрения являются получение, передача, воспроизведения и обработка визуальной информации без потери информативности, в том числе в ряде случаев и интерпретация информации.

Получаемое с сенсора цифровое изображение неминуемо содержит шумы различной природы: дробовой шум, присущий исходному регистрируемому световому сигналу, несовершенство светочувствительной матрицы, появление наводок от источников электромагнитных полей, неблагоприятные условия съемки и т.д. Подобные проблемы возникают как при применении фотоприемных матриц видимого диапазона, так и при использовании детекторов инфракрасного излучения. Таким образом, остро встает вопрос необходимости устранения шумов.

Современные устройства захвата цифрового изображения, как правило, позволяют получать снимки в широком динамическом диапазоне яркостей/интенсивностей (high dynamic range — HDR), тем не менее, стандартные устройства вывода ограничивают исходные изображения и работают с изображениями низкого динамического диапазона яркостей (low dynamic range — LDR). Вследствие этого выводимое LDR-изображение может достаточно сильно отличаться от исходного HDR-изображения. Такие изменения можно объяснить потерей информации изображения в процессе сжатия градаций его интенсивностей до достаточного для устройства вывода уровня.

Решение обозначенных проблем предполагает предварительную обработку изображений. Алгоритмы и фильтры, используемые для предобработки, имеют целью подавление шумов, извлечение, а также сохранение информации необходимой для дальнейшей обработки.

Специализированные камеры, используемые в системах технического зрения, как правило, формируют изображения с разрядностью 12 бит. При этом стандартом качества для высококачественных цифровых изображений является

разрядность 8 бит. Использование камер, обеспечивающих получение изображений 12 битной разрядности, дает возможность осуществлять преобразования яркости без потери информативности. При преобразовании 12-битного изображения в 8-ми битное происходит отбрасывание избыточной и не значимой для системы распознавания информации.

Целью настоящей работы является разработка модуля обработки цифровых изображений с аппаратной реализацией на программной логике на кристалле XC7Z020-2CLG400I семейства ZYNQ-7000 от Xilinx, позволяющей осуществлять сжатие HDR-изображения без потери детализации изображения и с высокой скоростью обработки.

Постановка задачи

Рассмотрим модуль сжатия динамического диапазона на примере камеры видимого диапазона:

- фотоприемная матрица формирует 12-ти битный сигнал, который зависит линейно от интенсивности падающего света;
- разрабатываемое устройство должно формировать 8-ми битное изображение в режиме цветовой модели RGB.

Для достижения поставленной цели необходимо решить следующие задачи:

- выбор фильтра для обработки изображения;
- решение задачи оптимизации выбранного фильтра;
- аппаратная реализация выбранного фильтра на программированной системе на кристалле XC7Z020-2CLG400I семейства ZYNQ-7000 от Xilinx;
- оценка эффективности работы фильтра.

На первом этапе работы необходимо разработать алгоритм сжатия динамического диапазона изображения (с 12-ти бит до 8 бит), позволяющий сохранять детали и границы объектов изображения, а также обеспечивающий отсутствие артефактов и минимизацию потери детальности изображения.

На втором этапе работы осуществляется реализация управляемого фильтра в программируемой логике и дальнейшая его оптимизация для увеличения скорости работы алгоритма и уменьшения расхода ресурсов модуля на кристалле семейства Zynq-7000 (XC7Z020-2CLG400I) для осуществления возможности обработки изображения в режиме реального времени.

Выбор фильтра для обработки изображения

В пространственных фильтрах для обработки изображений значения пикселей конечного изображения, как правило, вычисляется путем выполнения ряда линейных или нелинейных операций по заданной окрестности, называемой фильтром, ядром, маской или окном [1]. Достаточно распространённым из них для решения задач компрессии являются билатеральный фильтр (bilateral filter – BF) и управляемый фильтр (guided filter – GF) [2].

Билатеральный (bilateral) фильтр

Билатеральный фильтр был впервые предложен в 1998 году Tomasi и Manduchi [3] Этот фильтр представляет собой семейство наиболее простых и популярных среди пространственных нелинейных средневзвешенных фильтров. Интенсивность каждого пикселя для отфильтрованного изображения в билатеральном фильтре рассчитывается, как взвешенное среднее соседних пикселей в пределах заданной прямоугольной окрестности (ядро фильтра или окно) по исходному изображению. Вес пикселей определяется двумя гауссовыми функциями. Первая функция G_{σ_s} (пространственный вес) обеспечивает локальность фильтра и дает большой вес соседним пикселям, определяется как: $G_{\sigma_s}(x) = e^{-x^2/2\sigma_s^2}$. Вторая G_{σ_r} (ранговый вес) – дает большой вес пикселям близкой интенсивности и определяется как: $G_{\sigma_r}(x) = e^{-x^2/2\sigma_r^2}$. Схема реализации билатерального фильтра представлена на рисунке 1(а).

Имея входное изображение p , в качестве результата билатеральной фильтрации получим выходное изображение q , в котором каждый пиксель i (рисунок 1(а)), согласно [4], будет определяться формулой:

$$q_i = \sum_{j \in S} W_{ij}(p) p_j, \quad (1)$$

$$W_{ij} = \frac{1}{K_i} \sum_{j \in S} G_{\sigma_s}(x_{p_i} - x_{p_j}) G_{\sigma_r}(L_{p_i} - L_{p_j}),$$

где: W_{ij} – ядро фильтра;

S – количество пикселей, покрываемых маской фильтра входного изображения p , размер которого определяется параметром σ_s ;

i – центральный пиксель в маске фильтра, $i \in S$ изображения p ;

j – соседние пиксели к пикселю i в маске фильтра, $j \in S$ входного изображения p ;

L_{p_i}, L_{p_j} – интенсивности пикселей i и j в окрестности S входного изображения p ;

x – координаты пикселей i и j в окрестности S изображения p (пиксель с координатами (0,0) расположен в верхнем левом углу изображений q и p);

K_i – сумма произведения всех весов для нормализации значения интенсивности по изображению ($K_i = \sum_{j \in S} G_{\sigma_s}(x_{p_i} - x_{p_j}) G_{\sigma_r}(L_{p_i} - L_{p_j})$).

Обобщением билатерального фильтра является совместный билатеральный фильтр (joint/cross bilateral filter — CBF) [4]. В таком фильтре пространственные и ранговые веса вычисляются не из входного изображения p , которое необходимо отфильтровать, а из другого опорного изображения I . Этот подход актуален, когда входное изображение p сильно зашумлено и не является надежным для вычисления и выделения границ. Схема реализации совместного билатерального фильтра представлена на рисунке 1 (б).

Имея сильно зашумленное входное изображения p и опорное изображение I , в качестве результата совместной билатеральной фильтрации получим выходное изображение q , в котором каждый пиксель i , (рисунок 1(б)) в соответствии с [4], будет определяться по формуле:

$$q_i = \sum_{j \in S} W_{ij}(I) p_j, \quad (2)$$

$$W_{ij} = \frac{1}{K_i} \sum_{j \in S} G_{\sigma_s}(x_{I_i} - x_{I_j}) G_{\sigma_r}(L_{I_i} - L_{I_j}).$$

Билатеральный фильтр широко используется в задачах шумоподавления [5, 6], удаления текстуры и артефактов с сохранением четких краев [18], сжатия HDR–изображения [7], выделения контуров для распознавания объектов [17, 16], для обнаружения движущихся точечных целей [8], обработке фотографий [9].

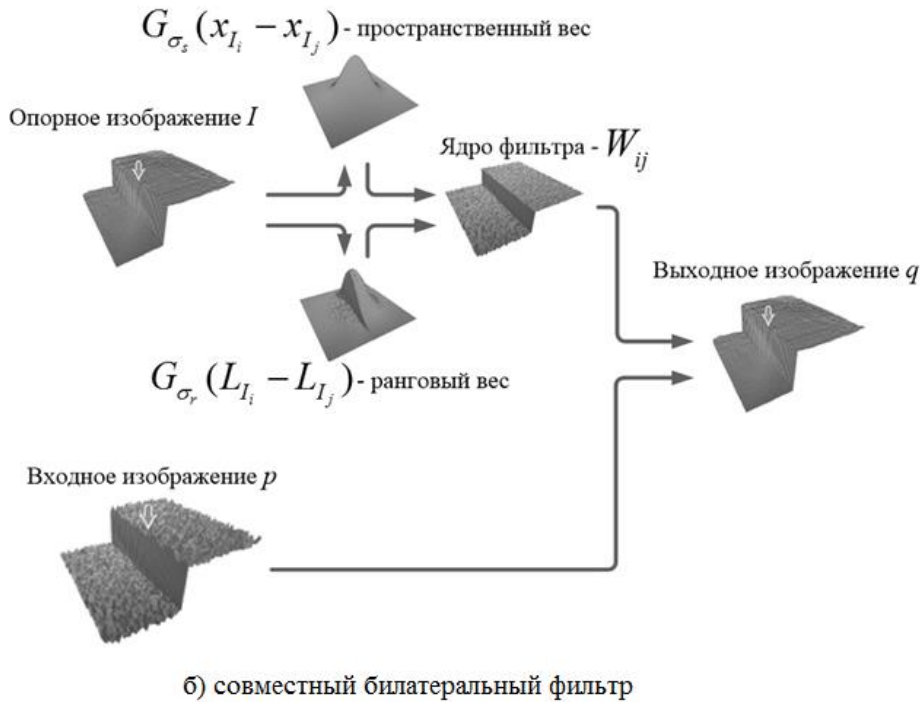
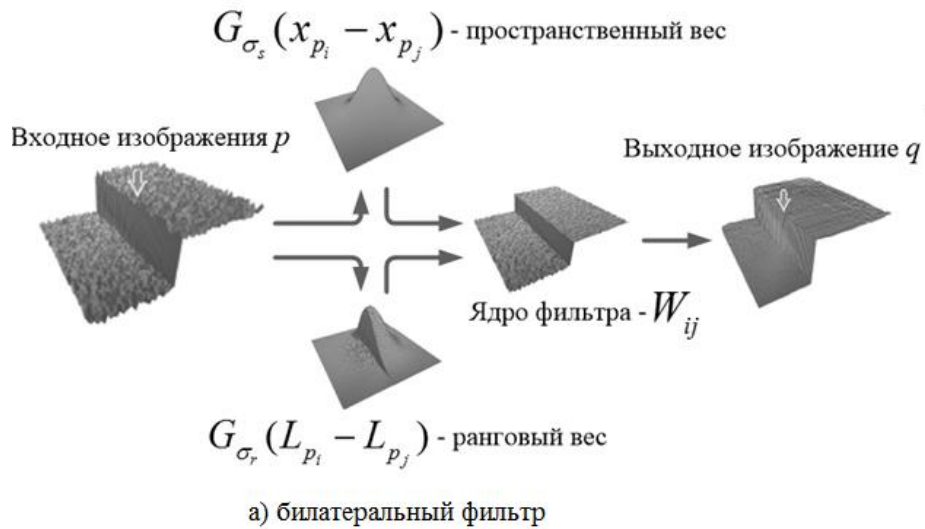


Рисунок 1 — Схемы билатеральных фильтров [2].

Тем не менее, билатеральный фильтр имеет ряд ограничений и недостатков. При повышении детализации, сжатии HDR-изображения или частотном разложении изображения, билатеральный фильтр может вести себя некорректно и порождать артефакты в виде нежелательных профилей по краям границы – “инверсия знака” [11, 7, 12, 2]. Кроме того, билатеральный фильтр требует повышенных вычислительных ресурсов [2].

Сложность алгоритма данного метода составляет $O(Nr^2)$ [13], где N – количество пикселей в изображении, r – радиус ядра фильтра. Упростить сложность алгоритма можно за счет дополнительных улучшений и модификаций фильтра [13], например, используя метод распределённых гистограмм интенсивности – $O(\log r)$ [14] или метод интегральных гистограмм – $O(1)$ [10, 15]. Однако, использование приближения и квантования, применяемых в этих методах, приводит к появлению артефактов и ухудшению качества изображения. Помимо этого, включение дополнительных методов требует больше памяти для работы, чем при использовании обычного билатерального фильтра, что влечет за собой дополнительные сложности при обработке цветных изображений.

Управляемый (guided) фильтр

Управляемый фильтр использует опорное изображение I аналогично совместному билатеральному фильтру. При этом GF, в отличие от BF и CBF, основан на локальной линейной модели. Согласно [2], управляемый фильтр определяет линейную зависимость каждого пикселя с координатами i выходного изображения q от опорного изображения I в каждом окне ω_k с центральным пикселем k , которые перекрывают данный пиксель ($i \in \omega_k$) (рисунки 2,3), по следующей формуле:

$$q_i = \bar{a}_i I_i + \bar{b}_i = \frac{1}{|\omega|} \sum_{k|i \in \omega_k} (a_k I_i + b_k), \quad (3)$$

где: $|\omega|$ – количество пикселей в заданном окне ω_k (при радиусе окна равным r , $|\omega| = 4r^2$);

\bar{a}_i и \bar{b}_i – средние значения линейных коэффициентов a_k и b_k во всех окнах, которые перекрывают пиксель i ($\bar{a}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} a_k$, $\bar{b}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} b_k$);

a_k и b_k – линейные коэффициенты для каждого окна ω_k , зависящие от входного изображения p и равные соответственно [2]:

$$a_k = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_k} I_i p_i + \mu_k \bar{p}_k}{\sigma_k^2 + \varepsilon}, \quad (4)$$

$$b_k = \frac{1}{|\omega|} \sum_{i \in \omega_k} p_i - a_k \mu_k = \bar{p}_k - a_k \mu_k, \quad (5)$$

где: μ_k и σ_k^2 – среднее значение и дисперсия опорного изображения I в окне ω_k ;

\bar{p}_k – среднее значение входного изображения p в окне ω_k ;

ε – параметр регуляризации.

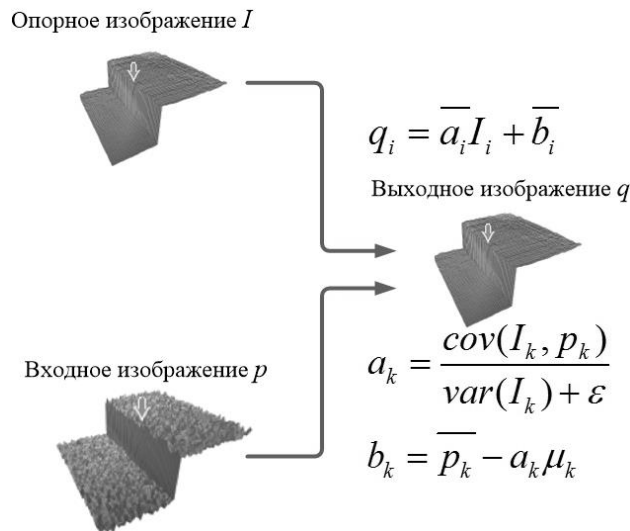


Рисунок 2 — Схема управляемого фильтра, переменные (cov) , (var) означают ковариацию и вариацию, соответственно [2]

Управляемый фильтр может применяться в качестве замены билатеральному фильтру в задачах сглаживания и выделения краев, удаления шумов и артефактов, а также сжатия HRD-изображения и т.д.

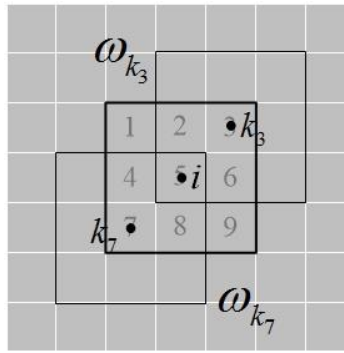


Рисунок 3— Схема работы с окном ω_k : ω_{k_3} и ω_{k_7} – окна с центральными пикселями k_3 и k_7 , перекрывающие пиксель i выходного изображения

Управляемый фильтр обладает всеми преимуществами билатерального фильтра и лишен его недостатков [2]. Сложность его исполнения составляет $O(1)$, отсутствуют проблемы с “инверсией знака” (рисунок 2). Управляемый фильтр является более быстрым алгоритмом с линейным сглаживающим эффектом с сохранением границ, который не зависит от размера ядра и диапазона интенсивности.

Сравнение результатов работы билатерального и управляемого фильтров по краям границ объектов представлено на рисунке 4. Причиной появления артефактов является нестабильность средневзвешенного веса по функции Гаусса расположенного на ребре пикселя, имеющего несколько подобных ему пикселей вокруг.

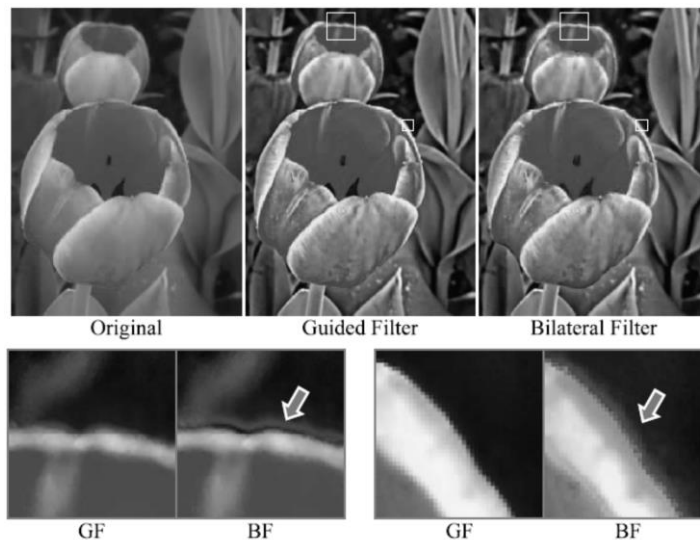


Рисунок 4— Сравнение результатов работ билатерального и управляемого фильтров по краям границ объектов [2]

Согласно [2], уравнения (3-5) определяют управляемый фильтр. Псевдокод выражений (3-5) представлен ниже в Алгоритме №1 Управляемого фильтра.

Алгоритм №1 Управляемого фильтра

Аббревиатуры переменных (*mean*), (*corr*), (*var*), (*cov*) означают: среднее значение, корреляцию, вариацию и ковариацию, соответственно.

Входные данные: входное изображение p , опорное изображение I , радиус окна r , параметр регуляризации ε .

Выходные данные: выходное изображение q .

1 Средние значения:

$$mean_I = f_{mean}(I) \quad mean_p = f_{mean}(p)$$

$$corr_I = f_{mean}(I * I) \quad corr_{Ip} = f_{mean}(I * p)$$

2 Значения вариации и ковариации:

$$var_I = corr_I - mean_I * mean_I \quad cov_{Ip} = corr_{Ip} - mean_I * mean_p$$

3 Коэффициенты a и b :

$$a = cov_{Ip} / (var_I + \varepsilon) \quad b = mean_p - a * mean_I$$

4 Средние значения коэффициентов a и b :

$$mean_a = f_{mean}(a) \quad mean_b = f_{mean}(b)$$

5 Выходное изображение:

$$q = mean_a * I + mean_b,$$

где f_{mean} – фильтр/маска радиуса r для расчета среднего значения.

Основополагающей сложностью управляемого фильтра, является расчет величин (*mean*), (*corr*), (*var*) во всех окнах ω_k .

Интегральное представление изображения

Для реализации управляемого фильтра на аппаратной части, необходимо минимизировать задействованные им ресурсы. Требуется уменьшить количество DSP-блоков выполняющих арифметические расчеты и минимизировать нагрузку на BRAM. Для ускорения расчета суммарной интенсивности заданной области в каждом окне ω_k согласно [1] будем использовать интегральное представление изображения.

Интегральное изображение представляет собой двумерный массив, размерность которого совпадает с размерностью исходного изображения. Элементы этого массива содержат сумму пикселей выше и левее, включая текущий пиксель, и рассчитываются по формуле:

$$LL(x, y) = \sum (L(i, j)), \quad (6)$$

где $(L(i, j))$ – интенсивность пикселя исходного изображения.

Таким образом, каждый элемент интегрального изображения $LL(x, y)$ содержит в себе сумму пикселей изображения в прямоугольнике от $(0, 0)$ до (x, y) .

Интегральное изображение представлено на рисунке 5. Это изображение используется для быстрого вычисления суммы яркостей заданных прямоугольных участков изображения и позволяет найти сумму элементов любого скользящего окна (ядра фильтра) за четыре обращения к памяти и три математических операции:

$$(ABCD) = LL(A) + LL(C) - LL(B) - LL(D). \quad (7)$$

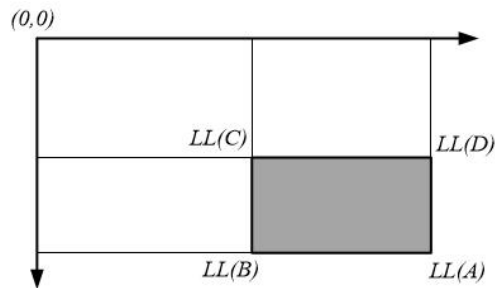


Рисунок 5 — Интегральное представление изображения

Классический метод расчета интегрального представления изображения в пикселе (x, y) представляет собой суммирование значения интегрального изображения строкой выше в пикселе $(x, y-1)$ и суммы значений пикселей в строке y от $(0, y)$ до (x, y) исходного изображения (рисунок 6 (а)). Для расчета суммы яркостей заданного участка исходного изображения, определяемого размером фильтра $(2r+1)$, размер интегрального изображения фильтра будет на один пиксель больше $((2r+1)+1)$ (рис. 6(б)).

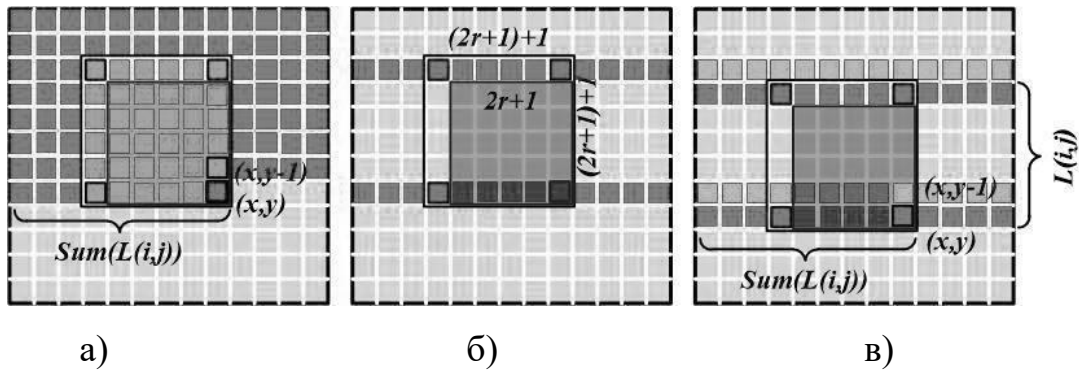


Рис. 6. Алгоритмы расчета интегрального изображения ядра фильтра: а) – классический метод, б), в) – оптимизированный метод

Для исключения хранения полного интегрального изображения в буфере для расчета площади ядра фильтра при каждом сдвиге, можно оптимизировать метод его расчета. В буфере будут храниться следующая информация:

а) Данные об интегральном изображении двух строк, первой и последней, которые обрамляют ядро фильтра (рис. 6б)

б) $((2r+1)+1)$ строк исходного изображения (согласно размеру интегрального изображения фильтра).

Это позволит при прохождении всех пикселей в строке и сдвиге ядра фильтра на строчку ниже заново произвести расчет интегрального изображения новых первой и последней строк, которые будет охватывать фильтр, записать их в буфер и использовать для дальнейших расчетов (рис. 6в). При этом ядро фильтра можно определить, как скользящее окно, так как оно пробегает каждый пиксель исходного изображения для преобразования и получения пикселей конечного изображения.

Для уменьшения количества потребляемого BRAM в работе используется сокращенная разрядность интегрального изображения. Пусть во входном изображении p , максимальная интенсивность равняется значению max_p и площадь скользящего окна равняется S .

Рассмотрим сумму интенсивностей по скользящему окну:

$$sum_p = LL(A) + LL(C) - LL(B) - LL(D). \quad (8)$$

Возьмем обе части равенства по модулю M , где $M > \max_p * S$, т.е. заранее больше чем максимальное значение суммы по любому окну изображения. Тогда:

$$\begin{aligned} \text{mod}(\text{sum}_p, M) &= \text{sum}_p = \\ &= \text{mod}(LL(A), M) + \text{mod}(LL(C), M) - \text{mod}(LL(B), M) - \text{mod}(LL(D), M) \end{aligned} \quad (9)$$

Следовательно, значения интегрального изображения достаточно знать по модулю M , т.е. максимальная разрядность интегрального изображения составляет

$$\lceil \log_2 \max_p * S \rceil = \text{image_depth} + \lceil \log_2 S \rceil, \quad (10)$$

где image_depth – разрядность входного изображения.

Оптимизация реализации управляемого (guided) фильтра

Выполним упрощение Алгоритма №1 в два этапа. Цель упрощения – уменьшение количества операций вычисления, с использованием техники интегрального представления для расчета площадей, которые используются в оценке средних значений параметров данного алгоритма.

Алгоритм №2. Guided filter

Входные данные: входное изображение p , опорное изображение I (интенсивность которого вычисляется с помощью интегрального представления), $p=I$, радиус скользящего окна равен r , площадь скользящего окна $S=(2r+1)*(2r+1)$, параметр регуляризации ε .

Выходные данные: выходное изображение q .

Этап 1:

При $p = I$, получаем $\text{mean}_I = \text{mean}_p$, $\text{corr}_I = \text{corr}_p = \text{mean}_{II}$, $\text{var}_I = \text{cov}_{Ip}$.

1 Средние значения:

$$\text{mean}_I = \text{sum}_I / S \quad \text{mean}_{II} = \text{sum}_I^2 / S$$

2 Свертка:

$$\text{var}_I = \text{mean}_{II} - \text{mean}_I * \text{mean}_I$$

3 Коэффициенты a и b :

$$a = \text{var}_I / (\text{var}_I + \varepsilon) \quad b = \text{mean}_I - a * \text{mean}_I = \text{mean}_I * \varepsilon / (\text{var}_I + \varepsilon)$$

4 Средние значения коэффициентов a и b :

$$mean_a = sum_a / S \quad mean_b = sum_b / S$$

5 Выходное изображение:

$$q = mean_a * I + mean_b$$

Этап 2:

Интенсивность опорного изображения I в каждом окне ω_k , вычисляем с помощью интегрального представления. Используя константу S , зависящую от размера скользящего окна, переписываем формулы расчета средних значений параметров, коэффициентов a и b , выходного изображения q . При этом, параметр регуляризации определим как $\varepsilon = n * S^2$, где n – константа задаваемая вручную.

1 Средние значения:

$$sum_I \quad sum_I^2$$

2 Коэффициенты a и b :

$$a = (sum_I^2 * S - sum_I * sum_I) / (sum_I^2 * S - sum_I * sum_I + \varepsilon)$$

$$b = ((sum_I * \varepsilon) / S) / (sum_I^2 * S - sum_I * sum_I)$$

3 Средние значения коэффициентов a и b :

$$sum_a \quad sum_b$$

4 Выходное изображение:

$$q = (sum_a * I + sum_b) / S$$

Схематически алгоритм представлен на рисунке 6.

Аппаратная реализация

В данной работе управляемый фильтр применялся для реализации HDR-компрессии, уменьшение разрядности с 12-ти бит до 8-ми для каждого канала RGB изображения и регулирования яркости и контраста выходного изображения. Рассмотрим общий алгоритм:

1 Вычисление логарифма интенсивности для входного изображения:

$$lL_p = \log_e \frac{20 * p_R + 40 * p_G + p_B}{61},$$

где: P_R, P_G, P_B – интенсивности красного, зеленого и синего канала RGB изображения соответственно;

lL_p – логарифм интенсивности входного изображения.

2 Вычисление базового изображения:

$$bl = \text{guided}(lL_p),$$

где: bl – логарифм входного изображения, обработанного с помощью управляемого фильтра;

$\text{guided}(lL_p)$ – аппаратный блок реализующий алгоритм №2 (Этап 2).

3 Вычисление выходных значений интенсивностей каждого канала:

$$q_z = p_z * \exp(bl * (cf - 1) - sb),$$

где: q_z, p_z – каналы входного и выходного RGB изображения;

$$cf = \frac{ic}{bl_{max} - bl_{min}};$$

$$sb = bl_{min} * cf + ib;$$

bl_{max}, bl_{min} – соответственно максимальное и минимальное значение базового изображения;

ic, ib – внешне задаваемые значение контраста и яркости.

В связи с особенностями человеческого восприятия цветов и интенсивности и требованием регуляции контраста и яркости, используется логарифмический масштаб интенсивности изображения для вычисления базового уровня изображения. В рассмотренном выше алгоритме наиболее сложной частью является вычисление базового уровня, поэтому далее будет рассматриваться только реализация управляемого фильтра.

Как указано выше используется логарифмическая шкала, поэтому используются числа с фиксированной точкой. Фотоприемная матрица формирует 12 битный сигнал, поэтому для хранения логарифма максимальной интенсивности без потери информации требуется 4 бита на целую часть и 12 бит на дробную часть. Размер управляемого фильтра (скользящего окна) – 5x5,

$\varepsilon = 0.1$. Общая схема аппаратной реализации алгоритма вычисления базового изображения представлена на рисунке 7.

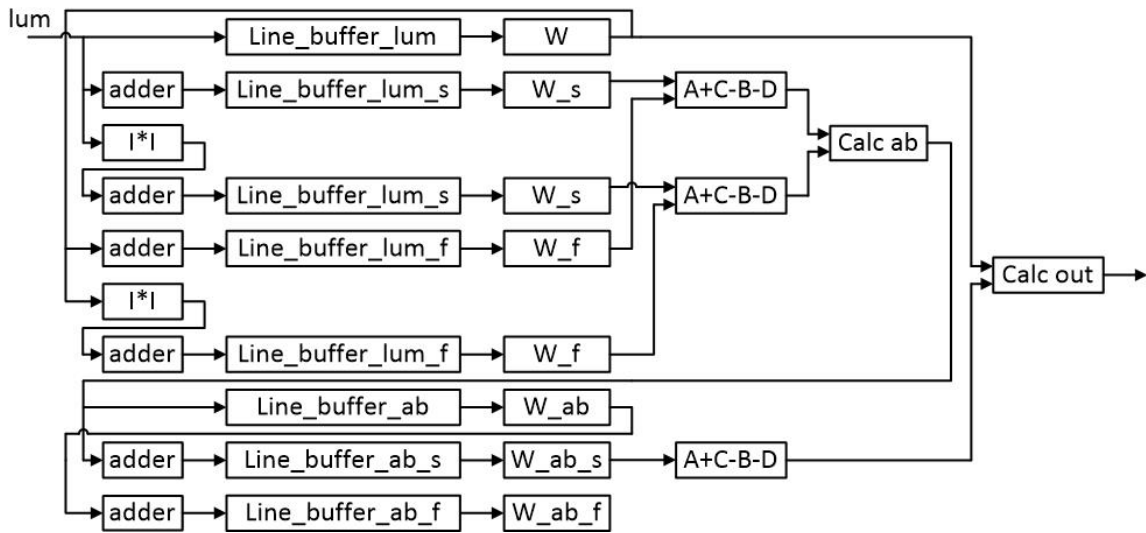


Рис. 7. Общая схема реализации управляемого фильтра

Line_buffer_lum – память BRAM для хранения 5 строк входного изображения;

W – скользящее окно 5x5 для пикселей входного изображения, реализованное на триггерах для одновременного считывания;

Line_buffer_lum_s – память BRAM для хранения нижней обрамляющей фильтр строки интегрального изображения для пикселей и квадратов пикселей входного изображения;

W_s – скользящее окно 5x5 для пикселей и квадратов пикселей входного изображения, реализованное на триггерах, для одновременного считывания;

Line_buffer_lum_f – память BRAM для хранения верхней обрамляющей фильтр строки интегрального изображения для пикселей и квадратов пикселей входного изображения;

W_f – скользящее окно 5x5 для пикселей и квадратов пикселей входного изображения, задержанного на 5 строк, реализованное на триггерах для одновременного считывания;

Line_buffer_ab – память BRAM для хранения 5 строк коэффициентов a и b ;

W_ab – скользящие окна 5×5 для коэффициентов a и b , реализованное на триггерах для одновременного считывания;

Line_buffer_ab_s – память BRAM для хранения нижней обрамляющей фильтр строки интегрального изображения для коэффициентов a и b ;

W_ab_s – скользящие окна 5×5 для коэффициентов a и b , реализованное на триггерах для одновременного считывания;

Line_buffer_lum_f – память BRAM для хранения верхней обрамляющей фильтр строки интегрального изображения для коэффициентов a и b ;

W_f – скользящие окна 5×5 для коэффициентов a и b , задержанного на 5 строк, реализованное на триггерах для одновременного считывания;

adder – сумматор по строке;

$I * I$ – операция возведения в квадрат;

$A + C - B - D$ – операция вычисления суммы пикселей в окне, по интегральному изображению (Алгоритм №2, Этап 2 п.п. 1 и 3);

calc_ab – операция вычисления коэффициентов a и b (Алгоритм №2 Этап 2 п. 2);

calc_out – операция вычисления выходного значения базового изображения (Алгоритм №2, Этап 2 п. 4).

Для уменьшения количества используемых BRAM используются сокращение разрядности и интегральное изображение, при этом для вычисления суммы пикселей в скользящем окне хранятся только 2 строки интегрального изображения, обрамляющие фильтр. Благодаря упрощениям требуется хранить 8 строк входного изображения, 2 строки интегрального изображения (10 бит целая часть, 12 бит дробная часть), 2 строки интегрального изображения квадратов (14 бит целая часть, 12 бит дробная часть).

В качестве альтернативного подхода возможно хранение всех строк интегрального изображения без сокращения разрядности, что требует в 2,25 раз большего количества BRAM, для изображения 1024x768. Упрощения из Алгоритм №2, этап 2, п.п. 2 и 3, значительно экономят DSP, в 2 раза, по сравнению с “наивной” реализацией алгоритма. Для вычисления средних значений a и b , используется тот же подход что и для вычисления средних значений изображения.

Заключение

1. Предложен модуль обработки цифровых изображений с аппаратной реализацией на программной логике на кристалле XC7Z020-2CLG400I семейства ZYNQ-7000 от Xilinx;
2. Разработан алгоритм сжатия HDR-изображения (с 12-ти бит до 8 бит), обеспечивающий сохранение мелких деталей исходного изображения и отсутствие артефактов;
3. Предложенное решение позволяет осуществлять предварительную обработку изображений с сохранением полезной информации необходимой для дальнейшей обработки.
4. Для увеличения скорости работы и уменьшения ресурсов проведена оптимизация реализованного в программируемой логике управляемого фильтра;
5. Уменьшенное количество ресурсов и реализация конвейерного подхода при вычислении средних значений позволяют достичь скорости обработки до 200 кадров в секунду для изображений 1024x768.

Литература

1. Гонсалес Р., Вудс Р. Цифровая обработка изображения. М.: Техносфера, 2005.
2. He K., Sun J., Tang X. Guided Image Filtering // *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2013. V.35. N.6. P. 1397-1409.

3. Tomasi C., Manduchi R. Bilateral filtering for gray and color images // *Proc. IEEE International Conference on Computer Vision*. 1998. P. 839-846.
4. Petschnigg G., Agrawala M., Hoppe H., Szeliski R., Cohen M., Toyama K. Digital Photography with Flash and No-Flash Image Pairs // *ACM Transactions on Graphics*. 2004. V.23. N.3. P. 664-672.
5. Wong W.C.K., Chung A.C.S., Yu S.C.H. Trilateral filtering for biomedical images // *Proc. IEEE International Symposium on Biomedical Imaging*. 2004. P. 820-823.
6. Liu C., Freeman W.T., Szeliski R., Kang S.B. Noise estimation from a single image // *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2006, P. 901-908.
7. Durand F., Dorsey J. Fast bilateral filtering for the display of high-dynamic-range images // *ACM Transactions on Graphics*. 2002. V.4. N.3. P. 257-266.
8. Bae T.-W., Sohng K.-I. Small target detection using bilateral filter based on edge component // *Infrared, Millimeter and Terahertz Waves*. 2010. V.31. P. 735-743.
9. Eisemann E., Durand F. Flash Photography Enhancement via Intrinsic Relighting // *ACM Transactions on Graphics*. 2004. V.23. N.3. P. 673-678.
10. Porikli F. Constant time $O(1)$ bilateral filtering // *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008, P. 1-8.
11. Farbman Z., Fattal R., Lischinski D., Szeliski R. Edge-preserving decompositions for multi-scale tone and detail manipulation // *ACM Transactions on Graphics*. 2008. V.27. N.3. P. 67.
12. Bae S., Paris S., Durand F. Two-scale tone management for photographic look // *ACM Transactions on Graphics*. 2006. V.25. N.3. P. 637-645.
13. Крапчатова Т.В., Филиппов М.В. Анализ эффективности алгоритмов билатеральной фильтрации // *Наука и образование [электронный журнал]*. 2012. № 2. URL <http://technomag.bmstu.ru/doc/340957.html>
14. Weiss B. Fast median and bilateral filtering // *ACM Transactions on Graphics*, 2006, V.25. N.3. P. 519-526.

15. Yang Q., Tan K.-H., Ahuja N. Real-time $O(1)$ bilateral filtering // *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2009, P. 557–564.
16. Fleishman S., Drori I., Cohen-Or D. Bilateral mesh denoising// *ACM Transactions on Graphics*. 2003. V.22. N.3. P. 943-949.
17. Jones T.R., Durand F., Desbrun M. Non-iterative, feature-preserving mesh smoothing// *ACM Transactions on Graphics*. 2003. V.22. N.3. P. 950-953.
18. Oh B.M., Chen M., Dorsey J., Durand F. Image-bases modeling and photo editing// *Proc. ACM SIGGRAPH*. 2001. P. 433-442.

Для цитирования:

Д. А. Гаврилов, А. В. Павлов, Д. Н. Щелкунов. Аппаратная реализация сжатия динамического диапазона цифровых изображений на ПЛИС Xilinx. Журнал радиоэлектроники [электронный журнал]. 2018. № 10. Режим доступа: <http://jre.cplire.ru/jre/oct18/6/text.pdf>
DOI 10.30898/1684-1719.2018.10.6