УДК 004.932.4 doi 10.26089/NumMet.v17r318

ПРИМЕНЕНИЕ ПОЛИНОМИАЛЬНЫХ ПРЕОБРАЗОВАНИЙ ДЛЯ БЫСТРОГО ВЫЧИСЛЕНИЯ ДВУМЕРНЫХ СВЕРТОК

И. А. Калиновский[1], **В. Г. Спицын**[2]

Рассмотрен быстрый алгоритм вычисления двумерных сверток, основанный на полиномиальных преобразованиях Нуссбаумера. Предложена его эффективная программная реализация с использованием набора SIMD-инструкций Intel AVX. Показано, что для ограниченного диапазона размеров ядер достигается 50% увеличение производительности вычислений по сравнению с прямым алгоритмом и методом быстрой свертки на основе быстрого преобразования Фурье, реализованных в библиотеке Intel IPP.

Ключевые слова: двумерная свертка, полиномиальные преобразования, быстрые алгоритмы.

1. Введение. Вычислительная сложность является важной характеристикой алгоритмов, от которой зависит время, необходимое для обработки некоторого объема данных на имеющемся вычислительном оборудовании. Требование обработки данных в реальном масштабе времени накладывает ограничения на быстродействие применяемых алгоритмов. В ряде случаев изменение структуры алгоритма может привести к существенному сокращению количества арифметических операций, необходимых для его вычисления с приемлемой точностью. Такая модифицированная версия алгоритма называется быстрой. Наиболее распространенным примером быстрого алгоритма является быстрое преобразование Фурье (БПФ), имеющее асимптотическую сложность $O\big(N\log(N)\big)$, в то время как сложность прямого алгоритма $O\big(N^2\big)$ [1].

Первый алгоритм БПФ по основанию 2 с прореживанием по времени, предложенный Кули–Тьюки в 1965 г., для одномерного сигнала требовал $\dfrac{1}{2}N\log_2 N$ комплексных умножений и $N\log_2 N$ комплексных сложений. Так как произведение комплексных чисел выполняется за 6 операций (3 умножения и 3 сложения) [2], то общее число вещественных арифметических операций в этом алгоритме составляет

$$\mathrm{FLOP} = 5N\log_2 N. \tag{1}$$

Простейшим способом вычисления двумерного дискретного преобразования Фурье (ДПФ) является строчно-столбцевой метод, в котором сначала вычисляются M N-точечных ДПФ-строк, а затем N M-точечных ДПФ-столбцов. Следовательно, при использовании БПФ Кули–Тьюки для этого понадобится $5MN\log_2(MN)$ операций. Более сложные методы, имеющие бóльшую эффективность при вычислении двумерного ДПФ, можно найти в [2].

Одной из базовых операций в обработке изображений и сигналов является пространственная фильтрация, используемая для сглаживания, удаления шумов, выделения признаков и др. Пространственная фильтрация осуществляется путем свертки изображения X размера $M \times N$ с некоторой прямоугольной матрицей H (ядром свертки) размера $U \times V$, состоящей из коэффициентов фильтра:

$$Y = X * H \,:\, y_{mn} = \sum_{u=0}^{U-1}\sum_{v=0}^{V-1} x_{m-u,n-v}\,h_{uv}, \quad m = 0,1,\ldots,M-1, \quad n = 0,1,\ldots,N-1. \tag{2}$$

Как можно заметить, сложность вычисления свертки по формуле (2) составляет $O(MNUV)$, т.е. эта операция не является вычислительно эффективной, особенно для изображений и фильтров большого размера. Однако с помощью теоремы о свертке [1]

$$F(X * H) = F(X)F(H) \tag{3}$$

и алгоритма БПФ фильтрация изображений может быть осуществлена с существенно меньшей асимптотической сложностью $O\big(MN\log_2(MN)\big)$ при условии, что ДПФ для ядра H выполнено заранее.

[1] Томский политехнический университет, Институт кибернетики, просп. Ленина, 30, 634050, Томск; аспирант, e-mail: kua_21@mail.ru

[2] Томский политехнический университет, Институт кибернетики, просп. Ленина, 30, 634050, Томск; профессор, e-mail: spvg@tpu.ru

Таким образом, $MN\big(10\log_2(MN)+1\big)$ вещественных операций требуется для вычисления двумерной свертки с использованием БПФ Кули–Тьюки. Для прямого вычисления свертки необходимо $2MNUV$ операций, поэтому быстрый алгоритм эффективен при $UV > 5\log_2(MN)$. Например, для фильтрации изображения с разрешением 640×480 пикселей с помощью квадратного фильтра следует применять БПФ, если размер ядра превышает 9×9 отсчетов. Однако к настоящему времени разработано множество более эффективных алгоритмов БПФ, стремящихся уменьшить постоянный множитель в (1). Так, для БПФ со смешанным основанием этот множитель равен 4 [3], а для модифицированной версии БПФ, предложенной в 2007 г. Джонсоном и Фриггом, составляет $34/9$ [3], при этом точная нижняя граница требуемого числа операций неизвестна. Однако в этих алгоритмах используются сложные схемы организации вычислений, поэтому реальная граница эффективности того или иного метода зависит от его программной реализации, особенностей аппаратной платформы и объема обрабатываемых данных.

2. Пространственная фильтрация изображений с использованием циклических сверток. Одним из недостатков БПФ является использование комплексной арифметики и относительно большой расход памяти. В то же время существуют и другие подходы к уменьшению числа операций при вычислении двумерных сверток. Один из них, предложенный Нуссбаумером, основан на использовании циклической свертки и полиномиальном преобразовании входных данных. Далее будут приведены основные идеи этого метода, подробное описание которого можно найти в [2].

Введем понятие циклической (или круговой) свертки. В случае циклической свертки предполагается, что дискретные сигналы X и H — периодические с одинаковым периодом. Если X и H представлены двумерными массивами $N\times N$, то циклическую свертку можно записать в виде

$$y_{mn} = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} x_{m-u,n-v}\,h_{uv}, \quad m,n = 0,1,\ldots,N-1. \tag{4}$$

Можно заметить, что если дополнить нулями ядро линейной свертки до размера изображения и вычислить циклическую свертку, то отсчеты циклической свертки совпадут с отсчетами линейной свертки в $(M-U+1)\times(N-V+1)$ точках. При этом за счет использования быстрых алгоритмов вычисления циклических сверток можно добиться существенного сокращения количества арифметических операций.

Рассмотрим одномерную дискретную свертку сигналов X и H длины M и N соответственно:

$$y_m = \sum_{n=0}^{N-1} x_{m-n}\,h_n, \quad m = 0,1,\ldots,M+N-2, \tag{5}$$

где x_m и h_n — вещественные числа.

Если представить дискретные сигналы X и H в виде полиномов по переменной Z: $X(Z) = \sum\limits_{m=0}^{M-1} x_m\,Z^m$ и $H(Z) = \sum\limits_{n=0}^{N-1} h_n\,Z^n$, то отсчеты свертки (5) можно вычислить как произведение этих полиномов

$$Y(Z) = X(Z)H(Z) = \sum_{m=0}^{M+N-2} y_m\,Z^m.$$

Введем понятие остаточных полиномов. Полином $P(Z)$ является делителем полинома $X(Z)$, если можно найти такой полином $D(Z)$, для которого выполняется соотношение $X(Z) = P(Z)D(Z)$. Полином $X(Z)$ называется неприводимым, если только его единственными делителями являются полиномы $P(Z)$, степени которых равны 0. Если $P(Z)$ не является делителем $X(Z)$, то при делении $X(Z)$ на $P(Z)$ образуется остаток $R(Z)$:

$$X(Z) = P(Z)D(Z) + R(Z).$$

Это представление единственно. Все полиномы, которые при делении на $P(Z)$ дают один и тот же остаток $R(Z)$, называются сравнимыми по модулю $P(Z)$: $R(Z) = X(Z) \bmod P(Z)$.

Непосредственно из определения сравнимых по модулю полиномов следует, что если $X_k(Z)$ и $R_k(Z)$ попарно сравнимы по модулю $P(Z)$, то и их суммы $\sum\limits_k X_k(Z)$, $\sum\limits_k R_k(Z)$ и произведения $\prod\limits_k X_k(Z)$, $\prod\limits_k R_k(Z)$ тоже сравнимы по модулю $P(Z)$.

С остаточными полиномами связана китайская теорема об остатках [4]. Предположим, что $P(Z)$ есть произведение d полиномов без общих множителей: $P(Z) \equiv \prod\limits_{i=1}^{d} P_i(Z)$.

Согласно китайской теореме об остатках, любой многочлен $X(Z)$ степени не большей $\deg\big(P(Z)\big)$ может быть однозначно представлен в виде последовательности $R = \big(R_1(Z), R_2(Z), \ldots, R_d(Z)\big)$, где $R_i(Z) \equiv X(Z) \bmod P_i(Z)$.

При этом многочлен $X(Z)$ может быть восстановлен по остаткам R_i:

$$X(Z) \equiv \sum_{i=1}^{d} S_i(Z) R_i(Z) \bmod P(Z), \quad \text{где} \quad S_j \equiv \begin{cases} 0 \bmod P_i(Z), & i \neq j, \\ 1 \bmod P_j(Z), & i = j, \end{cases} \quad S_j(Z) \equiv T_j(Z) \prod_{\substack{i=1 \\ i \neq j}}^{d} P_i(Z),$$

где $T_j(Z)$ определяются из формулы $T_j(Z) \prod_{\substack{i=1 \\ i \neq j}}^{d} P_i(Z) \equiv 1 \bmod P(Z)$.

Китайская теорема об остатках играет центральную роль при вычислении сверток, так как она позволяет заменять вычисление произведения $X(Z)H(Z)$ двух больших полиномов по модулю $P(Z)$ на d произведений $X_i(Z)H_i(Z)$ по модулю $P_i(Z)$ значительно меньших полиномов.

Опишем способ вычисления двумерной циклической свертки. Свертка (4) представима в виде одномерной свертки полиномов по модулю $Z^N - 1$:

$$\begin{gathered} Y_n(Z) \equiv \sum_{v=0}^{N-1} X_{n-v}(Z) H_v(Z) \bmod \big(Z^N - 1\big), \\ X_n(Z) = \sum_{m=0}^{N-1} x_{mn} Z^m, \quad n = 0, 1, \ldots, N-1; \quad H_v(Z) = \sum_{u=0}^{N-1} h_{uv} Z^u, \quad v = 0, 1, \ldots, N-1, \end{gathered} \tag{6}$$

где отсчеты y_{mn} получаются из N полиномов $Y_n(Z)$ путем выбора коэффициентов при Z:

$$Y_n(Z) = \sum_{m=0}^{N-1} y_{mn} Z^m, \quad n = 0, 1, \ldots, N-1.$$

Предположим, что $N = q$, где q — нечетное простое число. В этом случае $\big(Z^q - 1\big)$ является произведением двух циклотомических полиномов: $Z^q - 1 = (Z-1)P(Z)$, $P(Z) = Z^{q-1} + Z^{q-2} + \ldots + 1$.

Циклотомические полиномы неприводимы над полем рациональных чисел. Поскольку $Y_n(Z)$ определен по модулю $\big(Z^q - 1\big)$, он может быть вычислен посредством приведения $X_{n-v}(Z)$ и $H_v(Z)$ по модулю $(Z-1)$ и $P(Z)$, множителей $\big(Z^q - 1\big)$, вычисления полиномиальных сверток $Y_{1n}(Z) \equiv Y_n(Z) \bmod P(Z)$ и $Y_{2n}(Z) \equiv Y_n(Z) \bmod (Z-1)$ над приведенными полиномами и последующего восстановления $Y_n(Z)$ в соответствии с китайской теоремой об остатках:

$$\begin{gathered} Y_n(Z) \equiv S_1(Z) Y_{1n}(Z) + S_2(Z) Y_{2n}(Z) \bmod \big(Z^q - 1\big), \\ \text{где} \quad \begin{cases} S_1(Z) \equiv 1, & S_2(Z) \equiv 0 \bmod P(Z), \\ S_1(Z) \equiv 0, & S_2(Z) \equiv 1 \bmod (Z-1), \end{cases} \quad \begin{cases} S_1(Z) = \dfrac{1}{q}\,\big[q - P(z)\big], \\ S_2(Z) = \dfrac{1}{q}\,P(z). \end{cases} \end{gathered}$$

Таким образом, для получения $Y_n(Z)$ требуется выполнить две полиномиальные свертки $Y_{1n}(Z)$ и $Y_{2n}(Z)$. Так как свертка $Y_{2n}(Z)$ определена по модулю $(Z-1)$, то ее вычисление сводится к сверточному произведению скаляров X_{2n} и H_{2v}, полученных подстановкой 1 вместо Z в X_n и H_v:

$$Y_{2n}(Z) = \sum_{v=0}^{q-1} X_{2,n-v} H_{2v}, \quad n = 0, 1, \ldots, q-1; \quad X_{2n} = \sum_{m=0}^{q-1} x_{mn}, \quad H_{2v} = \sum_{u=0}^{q-1} h_{uv}. \tag{7}$$

Наиболее трудоемким является определение $Y_{1n}(Z)$. Для его упрощения Нуссбаумером было предложено преобразование $\overline{X}_k(Z)$, имеющее такую же структуру, как и ДПФ, но в котором комплексные экспоненты заменены на степени переменной Z, а все операции выполняются по модулю $P(Z)$:

$$\overline{X}_k(Z) \equiv \sum_{n=0}^{q-1} \overline{X}_{1n}(Z) Z^{nk} \bmod P(Z), \quad k = 0, 1, \ldots, q-1; \quad \overline{X}_{1n}(Z) \equiv X_n(Z) \bmod P(Z). \tag{8}$$

Обратное преобразование определяется в виде

$$X_{1n}(Z) \equiv \frac{1}{q} \sum_{k=0}^{q-1} \overline{X}_k(Z) Z^{-nk} \bmod P(Z), \quad Z^{-nk} \equiv Z^{(q-1)nk}, \quad n = 0, 1, \ldots, q-1. \tag{9}$$

Доказано, что полиномиальные преобразования (ПП) обладают свойствами циклической свертки, а выражение (9) является обратным по отношению к (8), т.е. для ПП справедлив аналог теоремы о свертке (3). Используя это свойство, Y_{1n} можно вычислить с помощью трех ПП и q полиномиальных умножений $\overline{X}_{n-v}(Z)\overline{H}_v(Z)$, определенных по модулю $P(Z)$. Более общее определение полиномиальных преобразований, описанное в [2], позволяет аналогичным способом конструировать алгоритмы вычисления для q^c-точечных циклических сверток (q — простое, $c \in \mathbb{N}$).

Схема вычисления циклической свертки через ПП показана на рис. 1. Предполагается, что все вычисления для ядра H выполнены заранее, так как обычно оно известно и фиксировано. Кроме того, отметим, что полиномиальные преобразования выполняются только с помощью операций сложения и простого сдвига коэффициентов в полиноме степени q, а единственные умножения, необходимые для вычисления двумерной свертки (6), соответствуют одной q-точечной свертке (7) и q произведениям полиномов $S_1(Z)\overline{X}_{n-v}(Z)\overline{H}_v(Z)$ по модулю $P(Z)$. В [2] показано, что если операции свертки и произведения полиномов выполняются с минимальным числом умножений, то вычисление $q \times q$-точечной циклической свертки (q — простое) будет выполнено за теоретически минимально возможное количество умножений. Вычисление циклических сверток размера $N = \prod_k q_k$ (т.е. $N \in \mathbb{N}$) с помощью полиномиальных преобразований может быть выполнено с использованием гнездовых алгоритмов [2], которые позволяют преобразовать двумерную $(q_1q_2 \times q_1q_2)$-точечную сверку в четырехмерную $\big([q_1 \times q_1] \times [q_2 \times q_2]\big)$-точечную свертку путем изменения индексации. В таблице приведено количество операций, требуемых для вычисления двумерных циклических сверток через ПП при использовании быстрых алгоритмов произведений полиномов, описанных в [2]. Быстрые алгоритмы вычисления полиномиальных преобразований (БПП), так же как и БПФ, позволяют многократно ускорить вычисление циклических сверток.

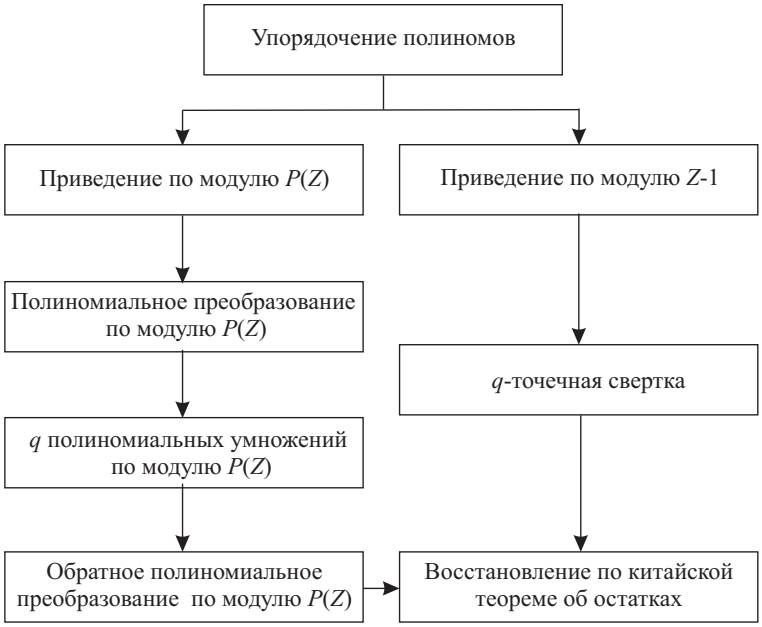

Рис. 1. Схема вычисления циклической свертки с помощью ПП [2]

Для пространственной фильтрации изображений через циклическую свертку применяются методы секционирования. В настоящей работе используется метод перекрытия с отбрасыванием, при котором изображение покрывается циклическими свертками размера N с шагом $N - U + 1$ по каждой оси, где U — размер ядра линейной свертки, при этом оставшиеся $(U-1)^2$ коэффициентов циклической свертки отбрасываются. На рис. 2 приведена зависимость количества арифметических операций, требуемых для свертки изображения с разрешением Full HD (1920×1080 пикс.) с ядрами различного размера при использовании прямого алгоритма (2), БПФ (3) с коэффициентом $C = 34/9$ и циклических сверток с размерами $N = \{12, 36, 72, 120\}$. Исходя из теоретической оценки, применение быстрых циклических сверток для пространственной фильтрации изображений оправдано для широкого диапазона размеров ядер (от 5×5 до 37×37). Этот метод вычисления линейных сверток является точным, описание некоторых приближенных методов можно найти в [5].

3. Вычислительный эксперимент. Несмотря на теоретическое превосходство БПП над БПФ для достаточно большого диапазона размеров ядер, этот метод ускорения вычислений двумерных линейных сверток не получил широкого распространения. Авторам не удалось найти его реализацию в специализированных библиотеках для высокопроизводительных вычислений, например Intel IPP и AMD APP. Тем не менее, теоретическая оценка показывает увеличение производительности пространственной фильтрации

до 2,8 раза, что имеет значение для обработки сигналов в реальном времени. Поэтому была осуществлена реализация метода БПП с целью оценки его эффективности на практике.

Для эксперимента была выбрана циклическая свертка с размером $N = 36$, поскольку она дает лучшее ускорение вычислений для небольших ядер линейной свертки (от 5×5 до 11×11), которые наиболее часто используются в задачах обработки изображений. Алгоритм вычисления свертки для $N = 36$ конструируется как комбинация БПП для сверток размера $N = 2^2$ и $N = 3^2$ с помощью гнездового алгоритма.

Число операций для циклических сверток, вычисляемых с помощью быстрых ПП и гнездовых алгоритмов [2]

Размеры свертки $N \times N$	Прямой метод	Быстрые полиномиальные преобразования			Ускорение
		Число умножений	Число сложений	Всего	
12×12	41 472	286	2 638	2 924	15,7
20×20	320 000	946	14 030	14 976	22,8
30×30	1 620 000	2 236	35 404	37 640	45,8
36×36	3 359 232	4 246	40 286	44 532	83,4
40×40	5 120 000	4 558	80 802	85 360	63,4
60×60	25 920 000	12 298	192 490	204 788	134,7
72×72	53 747 712	20 458	232 514	252 972	231,2
80×80	81 920 000	27 262	345 826	373 088	236,9
120×120	414 720 000	59 254	1 046 278	1 105 532	396,4

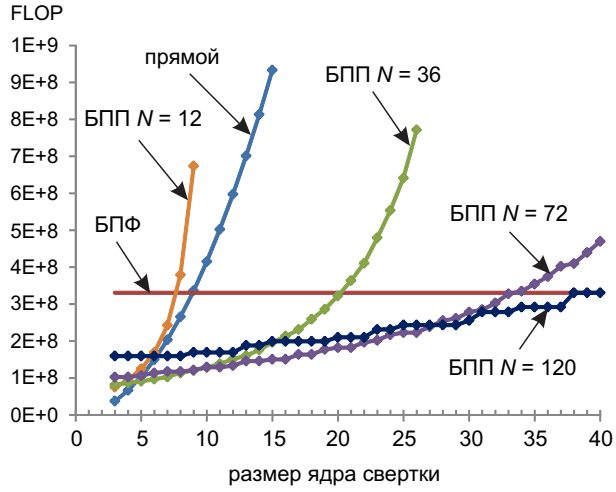

Рис. 2. Количество арифметических операций, требуемых для фильтрации изображения с разрешением Full HD

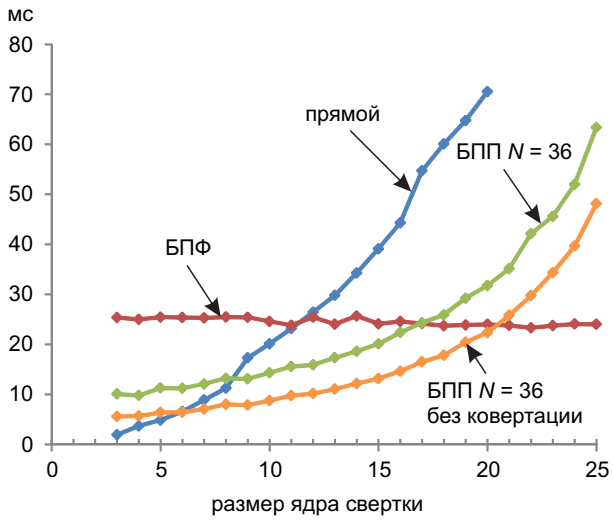

Рис. 3. Время фильтрации изображения с разрешением Full HD

Микроархитектура современных процессоров является суперскалярной, содержит блоки переупорядочения инструкций и переименования регистров, а исполнительные устройства способны выполнять большое количество разнообразных скалярных и векторных операций над данными. В связи с этим производительность алгоритма зависит не только от вычислительной сложности, но и от его структуры (наличие ветвлений, порядок обращения к памяти и др.), типа данных и используемых процессорных инструкций, эффективности векторизации и распараллеливания, а также от способности программной реализации максимально задействовать возможности аппаратной платформы. Для процессоров Intel разработана специализированная библиотека Integrated Performance Primitives (IPP), содержащая множество функций для обработки данных, высоко оптимизированных под соответствующую аппаратную архитектуру. В том числе она содержит реализации прямого алгоритма двумерной свертки и двумерного БПФ.

Очевидно, что наивная реализация быстрого алгоритма не может превзойти по скорости работы оптимизированную векторную реализацию прямого расчета даже при использовании существенно меньшего

количества операций. В связи с этим реализация БПП была осуществлена на языке ассемблера (inline assembler для Intel C++ Compiler) с использованием набора SIMD-инструкций AVX для x86-процессоров Intel. AVX-команды оперируют с 256-битными регистрами и способны выполнять одну операцию одновременно над 8-ю элементами типа float. Отметим, что при использовании опций автоматической векторизации компилятора Intel C++ Compiler 15.0 производительность генерируемого кода была значительно ниже по сравнению с ручной векторизацией.

Структура вычисления полиномиальных преобразований не поддается непосредственной векторизации из-за использования неочевидных алгоритмов перемножения и свертки полиномов, стремящихся минимизировать количество операций. Поэтому векторизация была осуществлена через одновременное вычисление 8-ми последовательно расположенных блоков циклических сверток, которые непрерывно покрывают участок изображения. Однако такой способ векторизации требует предварительной конвертации входных данных в новый формат, в котором пиксели участка изображения, имеющие одинаковые индексы внутри блоков, оказываются расположенными в памяти рядом. Соответственно, после окончания расчета векторного алгоритма циклической свертки требуется выполнить обратную конвертацию полученных отсчетов.

Тестирование реализации быстрого векторного алгоритма вычисления циклической свертки проводилось на процессоре Intel Core i7-3610QM, 3,1 ГГц (микроархитектура Ivy Bridge) в однопоточном режиме. Скорость фильтрации изображения с разрешением Full HD с помощью БПП сравнивалась с оптимизированными реализациями прямого метода вычисления свертки и БПФ из библиотеки Intel IPP 8.2, которые тоже выполнены с использованием набора инструкций AVX. Все вычисления осуществлялись с одинарной точностью. Замеры времени работы алгоритмов усреднены по 10^3 запускам и не включают в себя затраты на вычисление БПФ и БПП для ядер сверки (рис. 3).

Результаты тестирования показывают, что оптимизированный алгоритм фильтрации изображений на основе циклических сверток и БПП обеспечивает повышение производительности до 50% для размеров ядер свертки из интервала $[9 \times 9, 16 \times 16]$. При этом если данные уже представлены в требуемом формате, то фильтрация может быть ускорена до 2,5 раз для размеров ядер от 6×6 до 20×20. Время вычисления одной векторной циклической свертки составляет в среднем 0,022 мс, что дает производительность в 16,2 GFLOPS (см. таблицу) при теоретической производительности для одного ядра процессора в 24,8 GFLOPS. Отметим, что процедура предварительной конвертации данных может быть исключена за счет использования gather/scatter операций, доступных в наборе инструкций AVX2, которые позволяют упаковывать в векторный регистр элементы, содержащиеся в разных участках памяти. Границы применимости метода БПП для $N = 36$, определенные в результате вычислительного эксперимента (без использования процедуры конвертации), точно совпали с теоретической оценкой. Таким образом, алгоритм фильтрации изображений на основе метода БПП может найти применение в задачах, требующих максимально быстрой обработки данных.

4. Заключение. В настоящей статье впервые показано, что для ограниченного диапазона размеров ядер свертки метод полиномиальных преобразований быстрее, чем самые производительные реализации прямого метода фильтрации и БПФ на CPU. Этот диапазон может быть расширен за счет использования циклических сверток большего размера, например при $N = 72$. Однако существенным недостатком БПП является высокая трудоемкость реализации. Кроме того, остается открытым вопрос о применимости подобных алгоритмов для фильтрации изображений с использованием графических процессоров, которые в настоящее время являются основным вычислительным устройством для задач обработки изображений.

СПИСОК ЛИТЕРАТУРЫ

1. *Айфичер Э., Джервис Б.* Цифровая обработка сигналов. Практический подход. М.: Вильямс, 2004.
2. *Нуссбаумер Г.* Быстрое преобразование Фурье и алгоритмы вычисления сверток. М.: Радио и связь, 1985.
3. *Bernstein D.J.* The tangent FFT // Applied algebra, algebraic algorithms and error-correcting codes. 2007. **4851**. 291–300.
4. *Нестеренко Ю.В.* Теория чисел. М.: Академия, 2008.
5. *Макаров А.О., Старовойтов В.В.* Быстрые алгоритмы вычисления признаков на цифровых изображениях. Препринт № 1. Минск: Объединенный институт проблем информатики Национальной академии наук Беларуси, 2005.

Поступила в редакцию
05.05.2016

Application of Polynomial Transforms for Fast 2D Convolutions

I. A. Kalinovskii [1] **and V. G. Spitsyn** [2]

[1] *Tomsk Polytechnic University, Institute of Cybernetics; ulitsa Sovetskaya 84/3, Tomsk, 634034, Russia; Graduate Student, e-mail: kua_21@mail.ru*

[2] *Tomsk Polytechnic University, Institute of Cybernetics; ulitsa Sovetskaya 84/3, Tomsk, 634034, Russia; Dr. Sci., Professor, e-mail: spvg@tpu.ru*

Received May 5, 2016

Abstract: A fast algorithm for computing 2D convolutions based on the Nussbaumer polynomial transforms is considered. Its efficient implementation is proposed with the use of Intel AVX SIMD instructions. It is shown that, for a limited range of convolution kernels, the performance increases by 50% in comparison with the direct algorithm and with the method of fast convolution based on the fast Fourier transform implemented in the Intel IPP library.

Keywords: 2D convolution, polynomial transform, fast algorithms.

References

1. E. C. Ifeachor and B. W. Jervis, *Digital Signal Processing: A Practical Approach* (Prentice-Hall, Harlow, 2002; Vil'yams, Moscow, 2004).

2. H. J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms* (Springer, Heidelberg, 1982; Radio i Svyaz', Moscow, 1985).

3. D. J. Bernstein, "The Tangent FFT," in *Lecture Notes in Computer Science* (Springer, Heidelberg, 2007), Vol. 4851, pp. 291–300.

4. Yu. V. Nesterenko, *Number Theory* (Akademiya, Moscow, 2008) [in Russian].

5. A. O. Makarov and V. V. Starovoitov, *Fast Algorithms for Computing Features on Digital Images*, Preprint No. 1 (United Institute of Informatics Problems, Minsk, 2005).