

Разработка алгоритмов цифровой обработки изображений на основе метода Винограда в общем виде и анализ их вычислительной сложности

П.А. Ляхов^{1,2}, Н.Н. Нагорнов¹, Н.Ф. Семёнова¹, А.Ш. Абдулсаламова²

¹ Северо-Кавказский федеральный университет, 355017, Россия, г. Ставрополь, ул. Пушкина, д. 1;

² Северо-Кавказский центр математических исследований, 355017, Россия, г. Ставрополь, ул. Пушкина, д. 1

Аннотация

Стремительный рост количественных и качественных характеристик цифровых визуальных данных приводит к необходимости улучшения эксплуатационных показателей современных устройств обработки изображений. В данной работе предложены новые алгоритмы цифровой обработки двумерных изображений на основе метода Винограда в общем виде. Анализ полученных результатов показал, что использование метода Винограда сокращает вычислительную сложность обработки изображений до 84% по сравнению с традиционным прямым методом цифровой фильтрации в зависимости от параметров фильтра и фрагментов изображения, не влияя при этом на качество обработки изображения. Составленные матрицы преобразования метода Винограда и разработанные алгоритмы могут быть использованы в системах обработки изображений для улучшения эксплуатационных характеристик современных микроэлектронных устройств, осуществляющих очистку от шума и сжатие изображений, а также распознавание образов. Перспективным направлением дальнейших исследований является аппаратная реализация разработанных алгоритмов на современных устройствах вычислительной техники, таких как программируемые пользователем вентильные матрицы и интегральные схемы специального назначения, разработка алгоритмов цифровой обработки изображений, используемой в сверточных нейронных сетях, на основе метода Винограда в общем виде для одномерных вейвлет-фильтров с децимацией и для свертки с шагом.

Ключевые слова: цифровая обработка изображений, цифровая фильтрация, метод Винограда, вычислительная сложность.

Цитирование: Ляхов, П.А. Разработка алгоритмов цифровой обработки изображений на основе метода Винограда в общем виде и анализ их вычислительной сложности / П.А. Ляхов, Н.Н. Нагорнов, Н.Ф. Семёнова, А.Ш. Абдулсаламова // Компьютерная оптика. – 2023. – Т. 47, № 1. – С. 68-78. – DOI: 10.18287/2412-6179-CO-1146.

Citation: Lyakhov PA, Nagornov NN, Semyonova NF, Abdulsalyamova AS. Development of digital image processing algorithms based on the Winograd method in general form and analysis of their computational complexity. Computer Optics 2023; 47(1): 68-78. DOI: 10.18287/2412-6179-CO-1146.

Введение

Цифровая обработка изображений широко используется в различных областях современной науки и техники [1]. Основой многих методов обработки изображений является многократное использование цифровой фильтрации, имеющей высокую вычислительную сложность. Устройства обработки изображений не поспевают за стремительным ростом количественных и качественных характеристик цифровых визуальных данных [2, 3]. Одной из самых активно решаемых задач современной науки является разработка и развитие методов и средств для улучшения качественных и эксплуатационных показателей устройств цифровой обработки изображений. Проработано множество подходов, в числе которых параллеливание вычислений [4], использование различных аппаратных ускорителей [5] и так далее.

В данной статье рассмотрены методы цифровой фильтрации изображений. Различают фильтры нечетно-

го и четного порядков, содержащие нечетное и четное количество коэффициентов соответственно. Цифровая фильтрация прямым методом при использовании фильтра нечетного порядка представима в виде

$$I_2(x, y) = \sum_{i=-(r-1)/2}^{(r-1)/2} \sum_{j=-(r-1)/2}^{(r-1)/2} I_1(x+i, y+j)R(i, j),$$

где I_1 и I_2 – исходное и обработанное двумерные изображения соответственно; x и y – номера строки и столбца пикселя, обрабатываемого фильтром с маской R размера $r \times r$. Цифровая фильтрация прямым методом при использовании фильтра четного порядка представима в виде

$$I_2(x, y) = \sum_{i=-r/2+1}^{r/2} \sum_{j=-r/2+1}^{r/2} I_1(x+i, y+j)R(i, j).$$

Примерами фильтров нечетного и четного порядков являются фильтры сверточных слоев нейронной сети

[6] и тензорные вейвлеты [7] соответственно. Схема цифровой фильтрации прямым методом представлена на рис. 1, где фильтр с маской R размера 3×3 используется для получения фрагмента M размера 1×1 из фрагмента N размера 3×3 . Размер маски фильтра при использовании прямого метода совпадает с размером фрагментов, на которые делится исходное изображение для обработки.

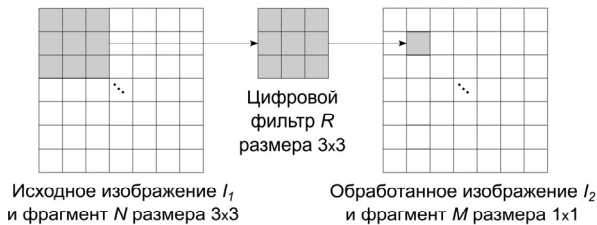


Рис. 1. Схема цифровой фильтрации прямым методом

В качестве современной альтернативы классическому прямому методу фильтрации используется метод Винограда [8], основанный на матричном умножении и позволяющий значительно снизить вычислительную сложность обработки изображений за счет групповой обработки пикселей. Обработанное изображение собирается не из совокупности отдельных пикселей, а из фрагментов некоторого размера, что позволяет уменьшить количество вычислительно сложных операций умножения за счет увеличения количества операций сложения. При этом качество обработки изображения остается неизменным [8]. Схема цифровой фильтрации методом Винограда представлена на рис. 2, где фильтр с маской R размера $r \times r$ используется для получения фрагмента M размера $m \times m$ из фрагмента N размера $n \times n$, где $n = m + r - 1$. Метод Винограда для фильтра R и фрагментов M принято обозначать как $F(m \times m, r \times r)$.

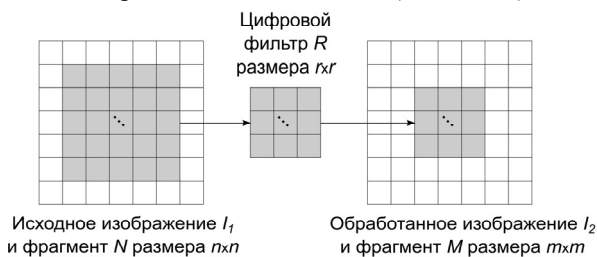


Рис. 2. Схема цифровой фильтрации методом Винограда $F(m \times m, r \times r)$

Метод Винограда активно используется для снижения вычислительной сложности и повышения скорости работы различных алгоритмов нейросетевой обработки изображений [5]. В статье [9] предложенные алгоритмы цифровой фильтрации на основе метода Винограда для сверточных слоев нейронных сетей показали превосходство над быстрым преобразованием Фурье по скорости работы глубокой нейронной сети при обработке больших массивов визуальных данных. Данный подход был расширен и обобщен на случаи обработки одномерных, двумерных и

трехмерных сигналов сверточной нейронной сетью [10]. На базе данных исследований разработаны различные архитектуры [11] и аппаратные ускорители [12–14] для высокопроизводительной реализации алгоритмов нейросетевой обработки изображений на основе метода Винограда. В работе [15] предложена архитектура цифрового фильтра с использованием метода Винограда в системе остаточных классов, аппаратная реализация которого позволила повысить скорость обработки изображений за счет повышения аппаратных и энергетических затрат. Несмотря на активное применение, в каждой из изученных работ рассматриваются только один или несколько частных случаев использования метода Винограда $F(m \times m, r \times r)$ для конкретных размера $r \times r$ цифрового фильтра и размера $m \times m$ фрагментов изображения. Алгоритмы обработки изображений на основе метода Винограда в общем виде еще не разработаны.

Целью данной работы является разработка алгоритмов обработки двумерных изображений на основе метода Винограда $F(m \times m, r \times r)$ в общем виде для любых значений размера $r \times r$ маски фильтра R и размера $m \times m$ фрагментов M обработанного изображения, а также анализ вычислительной сложности этих алгоритмов и сравнение с прямым методом при использовании двумерных фильтров нечетного и четного порядков.

Дальнейшая часть статьи организована следующим образом. В параграфе 1 представлены разработанные алгоритмы метода Винограда для обработки изображений. В параграфе 2 проведен анализ вычислительной сложности алгоритмов при различных условиях их использования и сравнение с прямым методом. В заключении подведены итоги исследования и сделаны выводы. В приложении А представлены матрицы обработки изображений, полученные при использовании алгоритмов и примененные для анализа за вычислительной сложности метода Винограда.

1. Алгоритмы построения матриц преобразования для метода Винограда в общем виде

Фильтрация изображения методом Винограда в матричной форме имеет вид [8]

$$M = A^T \left((GRG^T) \odot (B^T N B) \right) A, \quad (1)$$

где M – фрагмент обработанного изображения размера $m \times m$; R – маска фильтра размера $r \times r$; N – фрагмент исходного изображения размера $n \times n$, где $n = m + r - 1$; A^T , G , G^T , B^T , B , A – матрицы преобразования размеров $m \times n$, $n \times r$, $r \times n$, $n \times n$, $n \times m$, $m \times m$ соответственно; \odot – оператор поэлементного умножения матриц. В обозначении метода Винограда $F(m \times m, r \times r)$ указываются размер $m \times m$ фрагментов обработанного изображения и размер $r \times r$ маски фильтра, так как от них зависят размеры всех остальных матриц и фрагментов изображения.

Для составления матриц преобразования нужно выбрать n точек $s_0, s_1, \dots, s_{n-2}, s_{n-1}$, чтобы построить интерполяционный многочлен Лагранжа L [16]. На основе значений точек данного многочлена составляется матрица Вандермонда V [17] размера $n \times n$, имеющая следующий общий вид

$$V = \begin{pmatrix} s_0^0 & s_0^1 & s_0^2 & \dots & s_0^{n-2} & s_0^{n-1} \\ s_1^0 & s_1^1 & s_1^2 & \dots & s_1^{n-2} & s_1^{n-1} \\ s_2^0 & s_2^1 & s_2^2 & \dots & s_2^{n-2} & s_2^{n-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ s_{n-2}^0 & s_{n-2}^1 & s_{n-2}^2 & \dots & s_{n-2}^{n-2} & s_{n-2}^{n-1} \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix},$$

где значения элементов последней строки вычисляются как предел отношения к s_{n-1}^{n-1} при $s_{n-1} \rightarrow \infty$.

Пример 1. Пусть выбраны $n=6$ точек $s_0=0, s_1=1, s_2=-1, s_3=2, s_4=-2, s_5=\infty$ для многочлена L . Тогда матрица V имеет вид

$$V = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 2 & 4 & 8 & 16 & 32 \\ 1 & -2 & 4 & -8 & 16 & -32 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Далее по известным значениям r, m и n , а также по матрице V составляются матрицы преобразования A^T размера $m \times n, G$ размера $n \times r, B^T$ размера $n \times n$. Для получения матрицы A^T выберем из матрицы V^T первые m строк и n столбцов. Заменяя элемент, стоящий в последней строке и последнем столбце на 1, получим матрицу A^T . Процесс составления матрицы A^T представлен в виде алгоритма 1.

Алгоритм 1. Составление матрицы преобразования A^T для фильтрации изображений методом Винограда

Входные данные: матрица Вандермонда V

Выходные данные: матрица преобразования A^T

- 1: **рассчитать** V^T
- 2: $A^T(i, j) = V^T(i, j)$, где $i = \overline{1, m}$ и $j = \overline{1, n}$
- 3: $A^T(m, n) = 1$

Для получения матрицы G выберем из матрицы V первые n строк и r столбцов. Составим диагональную матрицу из наименьших общих кратных знаменателей элементов строк обратной транспонированной матрицы V . Разделим каждую строку матрицы, выбранной из V , на соответствующий отличный от нуля элемент диагональной матрицы. Заменяя в полученной матрице элемент, стоящий в последней строке и последнем столбце, на 1, получим матрицу G . Вычисление знаменателей элементов строк матрицы реализуется с помощью операции

$$den(x) = \begin{cases} 1, & x \in Z, \\ q, & x \in Q \setminus Z, \end{cases}$$

где x – элемент матрицы, q – знаменатель элементов строки матрицы, Z – множество целых чисел, $Q \setminus Z$ – множество рациональных чисел с исключением целых. Вычисление наименьшего общего кратного знаменателей элементов строк матрицы реализуется с помощью операции

$$LCM(x_1, x_2, \dots, x_i) = LCM(LCM(x_1, x_2, \dots, x_{i-1}), x_i).$$

Процесс составления матрицы G представлен в виде алгоритма 2.

Алгоритм 2. Составление матрицы преобразования G для фильтрации изображений методом Винограда

Входные данные: матрица Вандермонда V

Выходные данные: матрица преобразования G

- 1: **рассчитать** V^{-T}
- 2: **для** i от 1 до n
- 3: **для** j от 1 до n
- 4: $y_i = den(V^{-T}(i, j))$
- 5: **конец**
- 6: $z_i = LCM\{y_i\}$
- 7: **конец**
- 8: $G(i, j) = V(i, j) / z_i$, где $i = \overline{1, n}$ и $j = \overline{1, r}$
- 9: $G(n, r) = 1$

Для составления матрицы B^T построим диагональную матрицу из наименьших общих кратных знаменателей элементов строк обратной транспонированной матрицы V . Умножим каждую строку обратной транспонированной матрицы на соответствующий отличный от нуля элемент диагональной матрицы и получим матрицу B^T . Вычисление знаменателей элементов строк матрицы и их наименьшего общего кратного реализуется с помощью операций, описанных в алгоритме 2. Процесс составления матрицы B^T представлен в виде алгоритма 3.

Алгоритм 3. Составление матрицы преобразования B^T для фильтрации изображений методом Винограда

Входные данные: матрица Вандермонда V

Выходные данные: матрица преобразования B^T

- 1: **рассчитать** V^{-T}
- 2: **для** i от 1 до n
- 3: **для** j от 1 до n
- 4: $y_i = den(V^{-T}(i, j))$
- 5: **конец**
- 6: $z_i = LCM\{y_i\}$
- 7: **конец**
- 8: $B^T(i, j) = V^{-T}(i, j) / z_i$, где $i = \overline{1, n}$ и $j = \overline{1, n}$

Далее транспонируем составленные матрицы A^T, G и B^T . Получим матрицы преобразования A, G^T и B соответственно.

Пример 2. Пусть мы имеем фильтр размера 3×3 ($r=3$), а обработанное изображение будет состоять из

фрагментов 2×2 ($m=2$). В этом случае для обработки изображения его понадобится разделять на фрагменты размера 4×4 ($n=m+r-1=2+3-1=4$). При использовании точек $s_0=0, s_1=1, s_2=-1, s_3=\infty$ для построения многочлена L и составления матрицы V , матрицы преобразования A^T, G и B^T , составленные по алгоритмам 1–3, и их транспонированные версии A, G^T и B имеют вид

$$A^T = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 1 \end{pmatrix}, G = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 1/2 \\ 1/2 & -1/2 & 1/2 \\ 0 & 0 & 1 \end{pmatrix},$$

$$B^T = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}, A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -1 \\ 0 & 1 \end{pmatrix},$$

$$G^T = \begin{pmatrix} 1 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & -1/2 & 0 \\ 0 & 1/2 & 1/2 & 1 \end{pmatrix}, B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 \\ -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Матрицы преобразования A^T, G, B^T, A, G^T, B , составленные в примере 2 с использованием алгоритмов 1–3 для случая $F(2 \times 2, 3 \times 3)$, имеют широкое практическое применение, что подтверждается их использованием при проведении исследований в работах [9, 11, 12, 18]. Системы обработки изображений используют множество различных фильтров для обработки изображений разного размера. В результате размеры фрагментов исходного и обработанного изображения могут варьироваться в зависимости от конкретных параметров системы. Элементы матриц преобразования, помимо перечисленного, зависят еще и от выбора точек многочлена Лагранжа. Все указанные факты и закономерности влияют на вычислительную сложность обработки изображений. Далее представлены результаты анализа вычислительной сложности метода Винограда при обработке изображения масками фильтров разного размера, с использованием различных размеров фрагментов исходного и обработанного изображений и нескольких наборов точек многочлена Лагранжа для составления матрицы Вандермонда и матриц преобразования.

2. Анализ вычислительной сложности цифровой фильтрации изображений методом Винограда

Вычислительная сложность цифровой фильтрации изображений методом Винограда $F(m \times m, r \times r)$ в основном зависит от значений r и m , определяющих количество элементов матриц, а следовательно, и число необходимых умножений и сложений. Однако еще одним важным фактором является выбор значений точек $s_0, s_1, \dots, s_{n-2}, s_{n-1}$, так как на их основе строятся все используемые матрицы преобразования. В данной

статье рассмотрены случаи обработки изображений с разными размерами $m \times m$ ($m=2, 3, 4, 5$) обрабатываемых фрагментов и масок фильтра $r \times r$ ($r=3, 4$) при использовании трех наборов точек многочлена Лагранжа L для построения матриц Вандермонда V . Наборы точек и матрицы преобразования A^T, G, B^T , построенные с использованием алгоритмов 1–3, представлены в приложении А и могут быть использованы для разработки методов и алгоритмов обработки изображений на основе метода Винограда. На основе полученных матриц преобразования составлена табл. 1, содержащая результаты подсчета количества умножений и сложений, необходимых для цифровой фильтрации изображений прямым методом и методом Винограда для всех рассмотренных случаев. Количество умножений для каждого фрагмента в таблице рассчитано следующим образом.

1. При использовании прямого метода количество умножений, необходимых для получения фрагмента обработанного изображения, равно числу r^2 коэффициентов фильтра, так как размер фрагмента исходного изображения совпадает с размером используемого фильтра. Например, при обработке изображения фильтром размера 3×3 мы используем фрагменты изображения аналогичного размера и умножаем $3^2=9$ значений яркости пикселей на 9 соответствующих коэффициентов фильтра.

2. При использовании метода Винограда количество умножений, необходимых для получения фрагмента обработанного изображения, равно числу пикселей n^2 фрагмента исходного изображения, так как матричное умножение GRG^T в формуле (1) выполняется предварительно при известных значениях коэффициентов фильтров, и результат данного умножения, как и матрицы преобразования A^T, B^T, A, B , хранится в памяти устройства в виде констант, а умножение на константу сводится к масштабированию и сложению. Например, для умножения числа на константу 5, имеющую в двоичной системе счисления вид 101_2 , нужно масштабировать это число на два бита влево и сложить с исходным числом. Масштабирование чисел не требует значимых вычислительных ресурсов для своего выполнения. Таким образом все умножения матриц, кроме поэлементного, реализуются с помощью сложений, а поэлементное умножение выполняется единожды для двух матриц размера $n \times n$. Например, при обработке изображения методом Винограда $F(2 \times 2, 3 \times 3)$ исходное изображение делится на фрагменты размера 4×4 и количество умножений для получения одного фрагмента обработанного изображения равно $4^2=16$.

Количество сложений для каждого фрагмента в таблице рассчитано следующим образом.

1. При использовании прямого метода количество сложений, необходимых для получения фрагмента обработанного изображения, равно числу умножений, уменьшенному на единицу. Например,

при обработке изображения фильтром размера 3×3 мы умножаем $3^2=9$ значений яркости пикселей на 9 соответствующих коэффициентов фильтра и складываем полученный результат с использованием $9-1=8$ сложений. При этом дополнительных сложений не требуется, так что общее число сложений равно числу основных сложений.

2. При использовании метода Винограда количество основных сложений, необходимых для получения фрагмента обработанного изображения, равно сумме числа сложений ненулевых элементов матрицы по строкам, умноженной на сумму числа строк и столбцов этой матрицы по матрицам преобразования A^T и B^T , так как в формуле (1) данные матрицы являются первыми множителями, а их транспонированные версии B и A – вторыми. В примере 2 при обработке изображения методом Винограда $F(2 \times 2, 3 \times 3)$ в каждой из строк матрицы A^T размера 2×4 по три ненулевых элемента и для каждой из этих строк нужно по два суммирования, а в каждой из строк матрицы B^T размера 4×4 по два ненулевых элемента и для каждой из этих строк нужно по одному суммированию, поэтому количество основных сложений равно $(2+2)(2+4)+(1+1+1+1)(4+4)=56$.

3. При использовании метода Винограда количество дополнительных сложений, необходимых для получения фрагмента обработанного изображения, равно количеству единиц элемента матрицы в его двоичной записи, уменьшенному на единицу и умноженному на сумму числа строк и столбцов этой матрицы по каждому ненулевому элементу матриц A^T и B^T , отличному от степени двойки без учета знака. В случае 2 из приложения А, при обработке изображения методом Винограда $F(3 \times 3, 3 \times 3)$ в матрице B^T имеется один ненулевой элемент 3, отличный от степени двойки. В двоичной записи этого числа присутствует две единицы $3=11_2$, следовательно, оно требует одной дополнительной операции сложения при каждом использовании. Данное число будет использоваться $5+5$ раз (сумма числа строк и столбцов матрицы B^T), поэтому количество дополнительных сложений равно $1 \cdot (5+5)=10$.

4. Общее количество сложений равно сумме количества основных и дополнительных сложений.

При использовании метода Винограда мы получаем на выходе не значение яркости одного пикселя, а набор значений яркости группы пикселей, принадлежащих фрагменту размера $m \times m$. Некорректно говорить о количестве умножений и сложений, необходимых для обработки одного отдельно взятого пикселя, так как для расчета значения его яркости нужно обработать весь фрагмент целиком. В то же время фрагменты обработанного изображения при использовании прямого метода всегда имеют размер 1×1 . Для корректного сравнения вычислительной сложности прямого метода и метода Винограда в рамках данной статьи введем понятие удельного веса пикселя обрабо-

танного изображения, вычисляемого делением количества операций (умножений или сложений), необходимых для получения фрагмента обработанного изображения, на количество пикселей m^2 данного фрагмента. В примере 2 при обработке изображения методом Винограда $F(2 \times 2, 3 \times 3)$ количество умножений для получения одного фрагмента обработанного изображения $4^2=16$, а число пикселей фрагмента 2×2 обработанного изображения $2^2=4$, тогда удельный вес пикселя по умножениям равен $16/4=4$.

Для оценки времени работы микроэлектронного устройства, реализующего цифровую обработку изображений, воспользуемся методикой, основанной на подсчете количества срабатываний базовых логических элементов «и», «или» [19]. Время срабатывания одного такого элемента будем считать условной единицей (у.е.). В соответствии с этой методикой время выполнения умножения и сложения двух k -битных чисел на современных устройствах вычислительной техники при использовании для сложений одного из самых быстрых сумматоров, а именно Kogge-Stone adders (KSA) [20], а для умножений бинарного дерева из KSA без ускоряющих техник (например, деревьев Уоллеса и Дадда [21]), равно $8,8 \log_2 k + 5$ и $2 \log_2 k + 4$ [22] соответственно. Сравним время выполнения фильтрации при использовании прямого метода и метода Винограда на основе значений удельных весов пикселя по умножениям и сложениям, представленных в табл. 1. Например, время обработки изображения методом Винограда $F(2 \times 2, 3 \times 3)$ с удельными весами пикселя по умножениям и сложениям 4 и 14 соответственно будет равно $4 \cdot (8,8 \log_2 k + 5) + 14 \cdot (2 \log_2 k + 4) = 63,2 \log_2 k + 76$. При обработке изображений с глубиной цветных каналов, равной 8, цифровым фильтром с 8-битными коэффициентами получим $63,2 \log_2 8 + 76 = 265,6$. Изображения с 8-битными цветовыми каналами являются наиболее распространенными и получаемыми с использованием мобильных устройств и цифровых фотоаппаратов. Результаты расчета времени фильтрации изображений при использовании прямого метода и метода Винограда представлены в табл. 2, получены с допущениями и требуют проверки на современных устройствах вычислительной техники, таких как программируемые пользователем вентильные матрицы (field-programmable gate arrays, FPGAs) и интегральные схемы специального назначения (application-specific integrated circuits, ASICs).

На основе полученных результатов, представленных в табл. 1 и 2, сделаны следующие выводы.

1. Использование метода Винограда позволило уменьшить вычислительную сложность цифровой фильтрации изображений асимптотически от 55,56% до 84,00% по сравнению с прямым методом, в зависимости от размеров фильтра и фрагментов изображения, а также выбранного многочлена Лагранжа. При этом качество обработки изображения осталось неизменным.

Табл. 1. Количество умножений и сложений при обработке изображений прямым методом и методом Винограда для различных фильтров, различных фрагментов изображения и различных наборов точек многочлена Лагранжа

Фильтр	Метод	Количество пикселей фрагмента обработанного изображения	Многочлен Лагранжа	Для каждого фрагмента				Удельный вес пикселя			
				Умножения	Сложения			Умножения	Сложения		
					Основ-ные	Допол-нитель-ные	Всего		Основ-ные	Допол-нитель-ные	Всего
Фильтр нечетного порядка размера 3×3	Прямой	1	–	9	8	0	8	9,00	8,00	0,00	8,00
	$F(2 \times 2, 3 \times 3)$	4	L_1, L_2, L_3	16	56	0	56	4,00	14,00	0,00	14,00
	$F(3 \times 3, 3 \times 3)$	9	L_1, L_2, L_3	25	174	10	184	2,78	19,33	1,11	20,44
	$F(4 \times 4, 3 \times 3)$	16	L_1, L_2, L_3	64	332	24	356	2,25	20,75	1,50	22,25
Фильтр четного порядка размера 4×4	Прямой	1	–	16	15	0	15	16,00	15,00	0,00	15,00
	$F(2 \times 2, 4 \times 4)$	4	L_1, L_2, L_3	25	138	10	148	6,25	34,50	2,50	37,00
	$F(3 \times 3, 4 \times 4)$	9	L_1, L_2, L_3	36	291	24	315	4,00	32,33	2,67	35,00
	$F(4 \times 4, 4 \times 4)$	16	L_1	49	590	391	981	3,06	36,88	24,44	61,31
			L_2, L_3	49	576	196	772	3,06	36,00	12,25	48,25
	$F(5 \times 5, 4 \times 4)$	25	L_1	64	927	854	1781	2,56	37,08	34,16	71,24
L_2, L_3			64	927	320	1247	2,56	37,08	12,80	49,88	

Табл. 2. Время фильтрации изображений прямым методом и методом Винограда для различных фильтров, различных фрагментов изображения и различных наборов точек многочлена Лагранжа

Размер фильтра	Метод	Многочлен Лагранжа	Время выполнения фильтрации, у.е.	
			В общем случае	При $k=8$
3×3	Прямой	–	$95,2 \log_2 k + 67$	352,60
	$F(2 \times 2, 3 \times 3)$	L_1, L_2, L_3	$63,2 \log_2 k + 76$	265,60
	$F(3 \times 3, 3 \times 3)$	L_1, L_2, L_3	$65,34 \log_2 k + 95,66$	291,68
	$F(4 \times 4, 3 \times 3)$	L_1, L_2, L_3	$64,3 \log_2 k + 100,25$	293,15
4×4	Прямой	–	$170,8 \log_2 k + 140$	652,40
	$F(2 \times 2, 4 \times 4)$	L_1, L_2, L_3	$129 \log_2 k + 179,25$	566,25
	$F(3 \times 3, 4 \times 4)$	L_1, L_2, L_3	$105,2 \log_2 k + 160$	475,60
	$F(4 \times 4, 4 \times 4)$	L_1	$149,55 \log_2 k + 260,54$	709,19
		L_2, L_3	$123,43 \log_2 k + 208,3$	578,59
	$F(5 \times 5, 4 \times 4)$	L_1	$165,01 \log_2 k + 297,76$	792,79
L_2, L_3		$122,29 \log_2 k + 212,32$	579,19	

2. При этом время обработки изображения с глубиной цветных каналов, равной 8, цифровым фильтром с 8-битными коэффициентами сократилось до 27,10%. Для обоих рассмотренных фильтров наибольшее сокращение времени достигнуто при использовании фрагментов обработанного изображения размера на единицу меньше, нежели размер маски соответствующего фильтра, то есть для случаев $F(2 \times 2, 3 \times 3)$ и $F(3 \times 3, 4 \times 4)$.

3. Чем больше размер фрагментов изображения, тем меньше удельный вес пикселя по умножениям при цифровой фильтрации методом Винограда. При обработке фрагментов большого размера размер фильтра будет мал по отношению к ним, в результате чего удельный вес пикселя по умножениям будет немногим больше единицы. Например, при использовании фильтра размера 4×4 для обработки изображения с получаемыми на выходе фрагментами размеров 32×32 и 64×64 , количество умножений будет 1,20 и 1,10 соответственно, в то время как прямой метод требует 16 умножений.

4. Четность порядка фильтра не оказывает значимого влияния на вычислительную сложность используемых методов цифровой фильтрации, в отличие от размера маски фильтра при его использовании для обработки фрагментов исходного изображения размера $n \times n$, где $n = 3, 8$.

5. Использование точек $s_0, s_1, \dots, s_{n-2}, s_{n-1}$ со значениями, отличными от степеней двойки, для построения интерполяционного многочлена Лагранжа L и составления матрицы Вандермонда V приводит к значительному увеличению числа дополнительных суммирований и увеличению вычислительной сложности метода Винограда.

6. Использование отрицательных степеней двойки в качестве значений точек $s_0, s_1, \dots, s_{n-2}, s_{n-1}$ приводит к значительному увеличению числа масштабирований и усложняет проектирование устройств обработки изображений на основе метода Винограда.

Обобщая вышесказанное, использование метода Винограда позволяет значительно уменьшить вычислительную сложность методов цифровой обработки

изображений. При этом матрицы преобразования метода Винограда могут быть составлены с использованием разработанных алгоритмов 1–3 для любых: размера двумерного изображения; размера его фрагментов; размера и порядка маски двумерного фильтра. Составленные матрицы преобразования A^T , G , B^T , представленные в приложении А, могут быть использованы для разработки методов и алгоритмов обработки изображений на основе метода Винограда. Чем больше размер фрагментов изображения, тем выше эффективность метода Винограда. Выбор точек со значениями степеней двойки и противоположными им числами для построения многочлена Лагранжа L и составления матрицы Вандермонда V позволяет минимизировать количество дополнительных суммирований и уменьшить вычислительную сложность обработки изображений.

Перспективным направлением дальнейших исследований является аппаратная реализация разработанных алгоритмов на FPGA и ASIC, разработка алгоритмов цифровой обработки изображений на основе метода Винограда в общем виде для одномерных вейвлет-фильтров с децимацией и для свертки с шагом, используемой в сверточных нейронных сетях.

Заключение

Разработаны алгоритмы цифровой обработки двумерных изображений на основе метода Винограда в общем виде. Анализ полученных результатов показал, что использование метода Винограда сокращает вычислительную сложность цифровой фильтрации до 84 % по сравнению с традиционным прямым методом в зависимости от параметров фильтра и фрагментов изображения, не влияя при этом на качество обработки изображения. Составленные матрицы преобразования метода Винограда и разработанные алгоритмы могут быть использованы в системах обработки изображений для улучшения эксплуатационных характеристик современных микроэлектронных устройств, осуществляющих очистку от шума и сжатие изображений, а также распознавание образов. Перспективным направлением дальнейших исследований является аппаратная реализация разработанных алгоритмов на FPGA и ASIC, разработка алгоритмов цифровой обработки изображений на основе метода Винограда в общем виде для одномерных вейвлет-фильтров с децимацией и для свертки с шагом, используемой в сверточных нейронных сетях.

Благодарности

Авторы выражают благодарность СКФУ за помощь в рамках проекта поддержки малых научных групп и отдельных ученых. Исследование в параграфе 1 проведено в Северо-Кавказском центре математических исследований в рамках соглашения с Министерством науки и высшего образования Российской Федерации (соглашение № 075-02-2022-892). Иссле-

дование в параграфе 2 проведено при поддержке Российского научного фонда (проект № 21-71-00017).

References

- [1] Gonzalez RC, Woods RE. Digital image processing. Pearson Education Limited; 2018.
- [2] Rossinelli D, Fourestey G, Schmidt F, Busse B, Kurtcuoglu V. High-throughput lossy-to-lossless 3d image compression. IEEE Trans Med Imaging 2021; 40(2): 607-620. DOI: 10.1109/TMI.2020.3033456.
- [3] Smistad E, Østvik A, Pedersen A. High performance neural network inference, streaming, and visualization of medical images using FAST. IEEE Access 2019; 7: 136310-136321. DOI: 10.1109/ACCESS.2019.2942441.
- [4] Avenido HGD, Crisostomo RV. Image reconstruction from a large number of projections in proton and 12C ions computed tomography using sequential and parallel ART algorithms. Procedia Comput Sci 2022; 197: 126-134. DOI: 10.1016/J.PROCS.2021.12.126.
- [5] Mittal S, Vibhu. A survey of accelerator architectures for 3D convolution neural networks. J Syst Archit 2021; 115: 102041. DOI: 10.1016/J.SYSARC.2021.102041.
- [6] Chervyakov NI, Lyakhov PA, Nagornov NN, Valueva MV, Valuev GV. Hardware implementation of a convolutional neural network using calculations in the residue number system. Computer Optics 2019; 43(5): 857-868. DOI: 10.18287/2412-6179-2019-43-5-857-868.
- [7] Le NT, Wang J-W, Le DH, Wang C-C, Nguyen TN. Fingerprint enhancement based on tensor of wavelet subbands for classification. IEEE Access 2020; 8: 6602-6615. DOI: 10.1109/ACCESS.2020.2964035.
- [8] Winograd S. Arithmetic complexity of computations. Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics; 1980. ISBN: 0-89871-163-0.
- [9] Lavin A, Gray S. Fast algorithms for convolutional neural networks. 2016 IEEE Conf on Computer Vision and Pattern Recognition (CVPR) 2016: 4013-4021. DOI: 10.1109/CVPR.2016.435.
- [10] Yepez J, Ko SB. Stride 2 1-D, 2-D, and 3-D Winograd for convolutional neural networks. IEEE Trans Very Large Scale Integr Syst 2020; 28(4): 853-863. DOI: 10.1109/TVLSI.2019.2961602.
- [11] Mehrabian A, Miscuglio M, Alkabani Y, Sorger VJ, El-Ghazawi T. A Winograd-based integrated photonics accelerator for convolutional neural networks. IEEE J Sel Top Quantum Electron 2020; 26(1): 6100312. DOI: 10.1109/JSTQE.2019.2957443.
- [12] Shen J, Huang Y, Wen M, Zhang C. Toward an efficient deep pipelined template-based architecture for accelerating the entire 2-D and 3-D CNNs on FPGA. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 2020; 39(7): 1442-1455. DOI: 10.1109/TCAD.2019.2912894.
- [13] Wang X, Wang C, Cao J, Gong L, Zhou X. WinoNN: Optimizing FPGA-based convolutional neural network accelerators using sparse Winograd algorithm. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 2020; 39(11): 4290-4302. DOI: 10.1109/TCAD.2020.3012323.
- [14] Wu D, Fan X, Cao W, Wang L. SWM: A high-performance Sparse-Winograd matrix multiplication CNN accelerator. IEEE Trans Very Large Scale Integr Syst 2021; 29(5): 936-949. DOI: 10.1109/TVLSI.2021.3060041.
- [15] Valueva M, Lyakhov P, Valuev G, Nagornov N. Digital filter architecture with calculations in the residue number system by Winograd method F (2×2, 2×2). IEEE Access 2021; 9: 143331-143340. DOI: 10.1109/ACCESS.2021.3121520.

- [16] Waring EFRS. VII. Problems concerning interpolations. *Philos Trans R Soc* 1779; 69: 59-67. DOI: 10.1098/RSTL.1779.0008.
- [17] Horn RA, Johnson CR. *Topics in matrix analysis*. Cambridge: Cambridge University Press; 1991. ISBN: 978-0-521-30587-7.
- [18] Valueva MV, Lyakhov PA, Nagornov NN, Valuev GV. High-performance digital image filtering architectures in the residue number system based on the Winograd method. *Computer Optics* 2022; 46(5): 752-762. DOI: 10.18287/2412-6179-CO-933.
- [19] Zimmerman, R. Binary adder architectures for cell-based VLSI and their synthesis. A dissertation thesis for the degree of Doctor of technical sciences. Konstanz Hartung-Gorre; 1998. Diss ETH No 12480.
- [20] Kogge PM, Stone HS. A parallel algorithm for the efficient solution of a general class of recurrence equations. *IEEE Trans Comput* 1973; C-22(8): 786-793. DOI: 10.1109/TC.1973.5009159.
- [21] Parhami B. *Computer arithmetic: Algorithms and hardware designs*. Oxford University Press; 2010.
- [22] Lyakhov P, Valueva M, Valuev G, Nagornov N. High-performance digital filtering on truncated multiply-accumulate units in the residue number system. *IEEE Access* 2020; 8: 209181-209190. DOI: 10.1109/ACCESS.2020.3038496.

Приложение А

Для построения матрицы Вандермонда V использованы следующие наборы точек интерполяционного многочлена Лагранжа L .

1. L_1 на точках $0, 1, -1, 2, -2, 3, -3, \dots, l, -l, l+1, -(l+1), \dots, \infty$.
2. L_2 на точках $0, 1, -1, 2, -2, 4, -4, \dots, 2^l, -2^l, 2^{l+1}, -2^{l+1}, \dots, \infty$.
3. L_3 на точках $0, 1, -1, 2, -2, 1/2, -1/2, \dots, 2^l, -2^l, 1/2^l, -1/2^l, \dots, \infty$.

Случай 1. Метод Винограда $F(2 \times 2, 3 \times 3)$ будет одинаков для всех выбранных многочленов Лагранжа L_1, L_2 и L_3 . Матрицы преобразования A^T, G, B^T для данного случая представлены в Примере 2.

Случай 2. Метод Винограда $F(3 \times 3, 3 \times 3)$ будет одинаков для всех выбранных многочленов Лагранжа L_1, L_2 и L_3 . Используя Алгоритмы 1–3, построим матрицы A^T, G, B^T :

$$A^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & 0 \\ 0 & 1 & 1 & 4 & 1 \end{pmatrix}, \quad G = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 1/2 \\ 1/6 & -1/6 & 1/6 \\ 1/6 & 1/3 & 2/3 \\ 0 & 0 & 1 \end{pmatrix}, \quad B^T = \begin{pmatrix} 2 & -1 & -2 & 1 & 0 \\ 0 & 2 & 1 & -1 & 0 \\ 0 & -2 & 3 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 2 & -1 & -2 & 1 \end{pmatrix}.$$

Случай 3. Метод Винограда $F(4 \times 4, 3 \times 3)$ будет одинаков для всех выбранных многочленов Лагранжа L_1, L_2 и L_3 . Используя Алгоритмы 1–3, построим матрицы A^T, G, B^T :

$$A^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & -2 & 0 \\ 0 & 1 & 1 & 4 & 4 & 0 \\ 0 & 1 & -1 & 8 & -8 & 1 \end{pmatrix}, \quad G = \begin{pmatrix} 1/4 & 0 & 0 \\ 1/6 & 1/6 & 1/6 \\ 1/6 & -1/6 & 1/6 \\ 1/24 & 1/12 & 1/6 \\ 1/24 & -1/12 & 1/6 \\ 0 & 0 & 1 \end{pmatrix}, \quad B^T = \begin{pmatrix} 4 & 0 & -5 & 0 & 1 & 0 \\ 0 & 4 & 4 & -1 & -1 & 0 \\ 0 & -4 & 4 & 1 & -1 & 0 \\ 0 & -2 & -1 & 2 & 1 & 0 \\ 0 & 2 & -1 & -2 & 1 & 0 \\ 0 & 4 & 0 & -5 & 0 & 1 \end{pmatrix}.$$

Случай 4. Метод Винограда $F(2 \times 2, 4 \times 4)$ будет одинаков для всех выбранных многочленов Лагранжа L_1, L_2 и L_3 . Используя Алгоритмы 1–3, построим матрицы A^T, G, B^T :

$$A^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & 1 \end{pmatrix}, \quad G = \begin{pmatrix} 1/2 & 0 & 0 & 0 \\ 1/2 & 1/2 & 1/2 & 1/2 \\ 1/6 & -1/6 & 1/6 & -1/6 \\ 1/6 & 1/3 & 2/3 & 4/3 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad B^T = \begin{pmatrix} 2 & -1 & -2 & 1 & 0 \\ 0 & 2 & 1 & -1 & 0 \\ 0 & -2 & 3 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 2 & -1 & -2 & 1 \end{pmatrix}.$$

Случай 5. Метод Винограда $F(3 \times 3, 4 \times 4)$ будет одинаков для всех выбранных многочленов Лагранжа L_1, L_2 и L_3 . Используя Алгоритмы 1–3, построим матрицы A^T, G, B^T :

$$A^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & -2 & 0 \\ 0 & 1 & 1 & 4 & 4 & 1 \end{pmatrix},$$

$$G = \begin{pmatrix} 1/4 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & -1/6 & 1/6 & -1/6 \\ 1/24 & 1/12 & 1/6 & 1/3 \\ 1/24 & -1/12 & 1/6 & -1/3 \\ 0 & 0 & 0 & 1 \end{pmatrix}, B^T = \begin{pmatrix} 4 & 0 & -5 & 0 & 1 & 0 \\ 0 & 4 & 4 & -1 & -1 & 0 \\ 0 & -4 & 4 & 1 & -1 & 0 \\ 0 & -2 & -1 & 2 & 1 & 0 \\ 0 & 2 & -1 & -2 & 1 & 0 \\ 0 & 4 & 0 & -5 & 0 & 1 \end{pmatrix}.$$

Случай 6. Метод Винограда $F(4 \times 4, 4 \times 4)$. Используя Алгоритмы 1–3, построим матрицы A^T, G, B^T на основе многочлена L_1 :

$$A^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & -2 & 3 & 0 \\ 0 & 1 & 1 & 4 & 4 & 9 & 0 \\ 0 & 1 & -1 & 8 & -8 & 27 & 1 \end{pmatrix}, G = \begin{pmatrix} 1/12 & 0 & 0 & 0 \\ 1/12 & 1/12 & 1/12 & 1/12 \\ 1/24 & -1/24 & 1/24 & -1/24 \\ 1/24 & 1/12 & 1/6 & 1/3 \\ 1/120 & -1/60 & 1/30 & -1/15 \\ 1/120 & 1/40 & 3/40 & 9/40 \\ 0 & 0 & 0 & 1 \end{pmatrix}, B^T = \begin{pmatrix} 12 & -4 & -15 & 5 & 3 & -1 & 0 \\ 0 & 12 & 8 & -7 & -2 & 1 & 0 \\ 0 & -12 & 16 & -1 & -4 & 1 & 0 \\ 0 & -6 & -1 & 7 & 1 & -1 & 0 \\ 0 & 6 & -5 & -5 & 5 & -1 & 0 \\ 0 & 4 & 0 & -5 & 0 & 1 & 0 \\ 0 & -12 & 4 & 15 & -5 & -3 & 1 \end{pmatrix}.$$

Используя Алгоритмы 1–3, построим матрицы A^T, G, B^T на основе многочлена L_2 :

$$A^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & -2 & 4 & 0 \\ 0 & 1 & 1 & 4 & 4 & 16 & 0 \\ 0 & 1 & -1 & 8 & -8 & 64 & 1 \end{pmatrix}, G = \begin{pmatrix} 1/16 & 0 & 0 & 0 \\ 1/18 & 1/18 & 1/18 & 1/18 \\ 1/30 & -1/30 & 1/30 & -1/30 \\ 1/48 & 1/24 & 1/12 & 1/6 \\ 1/144 & -1/72 & 1/36 & -1/18 \\ 1/720 & 1/180 & 1/45 & 4/45 \\ 0 & 0 & 0 & 1 \end{pmatrix}, B^T = \begin{pmatrix} 16 & -4 & -20 & 5 & 4 & -1 & 0 \\ 0 & 16 & 12 & -8 & -3 & 1 & 0 \\ 0 & -16 & 20 & 0 & -5 & 1 & 0 \\ 0 & -8 & -2 & 9 & 2 & -1 & 0 \\ 0 & 8 & -6 & -7 & 6 & -1 & 0 \\ 0 & 4 & 0 & -5 & 0 & 1 & 0 \\ 0 & -16 & 4 & 20 & -5 & -4 & 1 \end{pmatrix}.$$

Используя Алгоритмы 1–3, построим матрицы A^T, G, B^T на основе многочлена L_3 :

$$A^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & -2 & 1/2 & 0 \\ 0 & 1 & 1 & 4 & 4 & 1/4 & 0 \\ 0 & 1 & -1 & 8 & -8 & 1/8 & 1 \end{pmatrix}, G = \begin{pmatrix} 1/4 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 \\ 1/18 & -1/18 & 1/18 & -1/18 \\ 1/72 & 1/36 & 1/18 & 1/9 \\ 1/120 & -1/60 & 1/30 & -1/15 \\ 1/45 & 1/90 & 1/180 & 1/360 \\ 0 & 0 & 0 & 1 \end{pmatrix}, B^T = \begin{pmatrix} 4 & -8 & -5 & 10 & 1 & -2 & 0 \\ 0 & -4 & 4 & 9 & -1 & -2 & 0 \\ 0 & -4 & 12 & -7 & -3 & 2 & 0 \\ 0 & 2 & -3 & -4 & 3 & 2 & 0 \\ 0 & 2 & -5 & 0 & 5 & -2 & 0 \\ 0 & 128 & 0 & -160 & 0 & 32 & 0 \\ 0 & -4 & 8 & 5 & -10 & -1 & 2 \end{pmatrix}.$$

Случай 7. Метод Винограда $F(5 \times 5, 4 \times 4)$. Используя Алгоритмы 1–3, построим матрицы A^T, G, B^T на основе многочлена L_1 :

$$A^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & -2 & 3 & -3 & 0 \\ 0 & 1 & 1 & 4 & 4 & 9 & 9 & 0 \\ 0 & 1 & -1 & 8 & -8 & 27 & -27 & 0 \\ 0 & 1 & 1 & 16 & 16 & 81 & 81 & 1 \end{pmatrix}, G = \begin{pmatrix} 1/36 & 0 & 0 & 0 \\ 1/48 & 1/48 & 1/48 & 1/48 \\ 1/48 & -1/48 & 1/48 & -1/48 \\ 1/120 & 1/60 & 1/30 & 1/15 \\ 1/120 & -1/60 & 1/30 & -1/15 \\ 1/720 & 1/240 & 1/80 & 3/80 \\ 1/720 & -1/240 & 1/80 & -3/80 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$B^T = \begin{pmatrix} 36 & 0 & -49 & 0 & 14 & 0 & -1 & 0 \\ 0 & 36 & 36 & -13 & -13 & 1 & 1 & 0 \\ 0 & -36 & 36 & 13 & -13 & -1 & 1 & 0 \\ 0 & -18 & -9 & 20 & 10 & -2 & -1 & 0 \\ 0 & 18 & -9 & -20 & 10 & 2 & -1 & 0 \\ 0 & 12 & 4 & -15 & -5 & 3 & 1 & 0 \\ 0 & -12 & 4 & 15 & -5 & -3 & 1 & 0 \\ 0 & -36 & 0 & 49 & 0 & -14 & 0 & 1 \end{pmatrix}.$$

Используя Алгоритмы 1–3, построим матрицы A^T , G , B^T на основе многочлена L_2 :

$$A^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & -2 & 4 & -4 & 0 \\ 0 & 1 & 1 & 4 & 4 & 16 & 16 & 0 \\ 0 & 1 & -1 & 8 & -8 & 64 & -64 & 0 \\ 0 & 1 & 1 & 16 & 16 & 256 & 256 & 1 \end{pmatrix}, \quad G = \begin{pmatrix} 1/64 & 0 & 0 & 0 \\ 1/90 & 1/90 & 1/90 & 1/90 \\ 1/90 & -1/90 & 1/90 & -1/90 \\ 1/288 & 1/144 & 1/72 & 1/36 \\ 1/288 & -1/144 & 1/72 & -1/36 \\ 1/5760 & 1/1440 & 1/360 & 1/90 \\ 1/5760 & -1/1440 & 1/360 & -1/90 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$B^T = \begin{pmatrix} 64 & 0 & -84 & 0 & 21 & 0 & -1 & 0 \\ 0 & 64 & 64 & -20 & -20 & 1 & 1 & 0 \\ 0 & -64 & 64 & 20 & -20 & -1 & 1 & 0 \\ 0 & -32 & -16 & 34 & 17 & -2 & -1 & 0 \\ 0 & 32 & -16 & -34 & 17 & 2 & -1 & 0 \\ 0 & 16 & 4 & -20 & -5 & 4 & 1 & 0 \\ 0 & -16 & 4 & 20 & -5 & -4 & 1 & 0 \\ 0 & -64 & 0 & 84 & 0 & -21 & 0 & 1 \end{pmatrix}.$$

Используя Алгоритмы 1–3, построим матрицы A^T , G , B^T на основе многочлена L_3 :

$$A^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 2 & -2 & 1/2 & -1/2 & 0 \\ 0 & 1 & 1 & 4 & 4 & 1/4 & 1/4 & 0 \\ 0 & 1 & -1 & 8 & -8 & 1/8 & -1/8 & 0 \\ 0 & 1 & 1 & 16 & 16 & 1/16 & 1/16 & 1 \end{pmatrix}, \quad G = \begin{pmatrix} 1/4 & 0 & 0 & 0 \\ 1/18 & 1/18 & 1/18 & 1/18 \\ 1/18 & -1/18 & 1/18 & -1/18 \\ 1/360 & 1/180 & 1/90 & 1/45 \\ 1/360 & -1/180 & 1/90 & -1/45 \\ 1/45 & 1/90 & 1/180 & 1/360 \\ 1/45 & -1/90 & 1/180 & -1/360 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$B^T = \begin{pmatrix} 4 & 0 & -21 & 0 & 21 & 0 & -4 & 0 \\ 0 & -4 & -4 & 17 & 17 & -4 & -4 & 0 \\ 0 & 4 & -4 & -17 & 17 & 4 & -4 & 0 \\ 0 & 2 & 1 & -10 & -5 & 8 & 4 & 0 \\ 0 & -2 & 1 & 10 & -5 & -8 & 4 & 0 \\ 0 & 64 & 128 & -80 & -160 & 16 & 32 & 0 \\ 0 & -64 & 128 & 80 & -160 & -16 & 32 & 0 \\ 0 & -4 & 0 & 21 & 0 & -21 & 0 & 4 \end{pmatrix}.$$

Сведения об авторах

Ляхов Павел Алексеевич, 1988 года рождения, в 2009 году окончил Ставропольский государственный университет по специальности «Математика», кандидат физико-математических наук, заведующий кафедрой математического моделирования, факультет математики и компьютерных наук имени профессора Н.И. Червякова ФГАОУ ВО «Северо-Кавказский федеральный университет»; заведующий отделом модулярных вычислений и искусственного интеллекта, региональный научно-образовательный математический центр «Северо-Кавказский центр математических исследований». Область научных интересов: цифровая обработка сигналов и изображений, искусственный интеллект, нейронные сети, модулярная арифметика, параллельные вычисления, высокопроизводительные вычисления, цифровые устройства, аппаратные ускорители. E-mail: ljahov@mail.ru.

Нагорнов Николай Николаевич, 1992 года рождения, в 2014 году окончил ФГАОУ ВО «Северо-Кавказский федеральный университет» по специальности «Прикладная математика и информатика», кандидат технических наук, доцент кафедры математического моделирования, факультет математики и компьютерных наук имени профессора Н.И. Червякова ФГАОУ ВО «Северо-Кавказский федеральный университет». Область научных интересов: цифровая обработка изображений, модулярная арифметика, параллельные вычисления, высокопроизводительные вычисления, цифровые устройства, аппаратные ускорители. E-mail: sparta1392@mail.ru.

Семёнова Наталия Фёдоровна, 1951 года рождения, в 1972 году окончила физико-математический факультет Ставропольского государственного педагогического института, кандидат физико-математических наук, доцент кафедры математического моделирования, факультет математики и компьютерных наук имени профессора Н.И. Червякова ФГАОУ ВО «Северо-Кавказский федеральный университет». Область научных интересов: математическое моделирование, параллельные вычисления, система остаточных классов, модулярная арифметика, диагональная функция. E-mail: n03c51@yandex.ru.

Абдулсальмова Альбина Шихаевна, 2000 года рождения, бакалавриant 4 курса специальности «Математика и компьютерные науки» ФГАОУ ВО «Северо-Кавказский федеральный университет», лаборант отдела модулярных вычислений и искусственного интеллекта, региональный научно-образовательный математический центр «Северо-Кавказский центр математических исследований». Область научных интересов: цифровая обработка изображений, параллельные вычисления, высокопроизводительные вычисления, цифровые устройства. E-mail: a.abdulsalyamova@mail.ru.

ГРНТИ: 28.23.15

Поступила в редакцию 5 апреля 2022 г. Окончательный вариант – 29 июня 2022 г.

Development of digital image processing algorithms based on the Winograd method in general form and analysis of their computational complexity

P.A. Lyakhov^{1,2}, N.N. Nagornov¹, N.F. Semyonova¹, A.S. Abdulsalyamova²

¹North-Caucasus Federal University, 355017, Stavropol, Russia, Pushkin street, 1;

²North-Caucasus Center for Mathematical Research, 355017, Stavropol, Russia, Pushkin street, 1

Abstract

The fast increase of the amount of quantitative and qualitative characteristics of digital visual data calls for the improvement of the performance of modern image processing devices. This article proposes new algorithms for 2D digital image processing based on the Winograd method in a general form. An analysis of the obtained results showed that the use of the Winograd method reduces the computational complexity of image processing by up to 84 % compared to the traditional direct digital filtering method depending on the filter parameters and image fragments, while not affecting the quality of image processing. The resulting Winograd method transformation matrices and the algorithms developed can be used in image processing systems to improve the performance of the modern microelectronic devices that carry out image denoising, compression, and pattern recognition. Research directions that show promise for further research include hardware implementation on a field-programmable gate array and application-specific integrated circuit, development of algorithms for digital image processing based on the Winograd method in a general form for a 1D wavelet filter bank and for stride convolution used in convolutional neural networks.

Keywords: digital image processing, digital filtering, Winograd method, computational complexity.

Citation: Lyakhov PA, Nagornov NN, Semyonova NF, Abdulsalyamova AS. Development of digital image processing algorithms based on the Winograd method in general form and analysis of their computational complexity. *Computer Optics* 2023; 47(1): 68-78. DOI: 10.18287/2412-6179-CO-1146.

Acknowledgements: The authors thank the North-Caucasus Federal University for the award of funding in the contest of competitive projects of scientific groups and individual scientists of North-Caucasus Federal University. The research in section 1 was supported by the North-Caucasus Center for Mathematical Research under agreement with the Ministry of Science and Higher Education of the Russian Federation (Agreement No. 075-02-2022-892). The research in section 2 was supported by the Russian Science Foundation (Project No. 21-71-00017).

Authors' information

Pavel Alekseyevich Lyakhov (b. 1988) graduated from Stavropol State University, specialty "Mathematics" in 2009. PhD of Physical and Mathematical Sciences. Head of Mathematical Modeling department, North-Caucasus Federal University, Head of Modular Computing and Artificial Intelligence department, regional scientific and educational mathematical center «North-Caucasus Center for Mathematical Research». Research interests are digital signal and image processing, artificial intelligence, neural networks, modular arithmetic, parallel computing, high-performance computing, digital circuits and hardware accelerators. E-mail: ljahov@mail.ru.

Nikolay Nikolaevich Nagornov (b. 1992) graduated from North-Caucasus Federal University, specialty "Applied Mathematics and Computer Science" in 2014. PhD of Computer Sciences. Associate professor of Mathematical Modeling department, North-Caucasus Federal University. Research interests are digital image processing, modular arithmetic, parallel computing, high-performance computing, digital circuits and hardware accelerators. E-mail: sparta1392@mail.ru.

Nataliya Fedorovna Semyonova (b. 1951) graduated from Physics and Mathematics faculty of the Stavropol State Pedagogical Institute in 1972. PhD of Physical and Mathematical Sciences. Associate professor of Mathematical Modeling department, North-Caucasus Federal University. Research interests are mathematical modeling, parallel computing, residue number system, modular arithmetic, diagonal function. E-mail: n03c51@yandex.ru.

Albina Shikhaevna Abdulsalyamova (b. 2000) 4th year bachelor of the specialty "Mathematics and Computer Science" at the North-Caucasus Federal University. Laboratory assistant of Modular Computing and Artificial Intelligence department, regional scientific and educational mathematical center «North-Caucasus Center for Mathematical Research». Research interests are digital image processing, parallel computing, high-performance computing and digital circuits. E-mail: a.abdulsalyamova@mail.ru.

Received April 5, 2022. The final version – June 29, 2022.
