КИБЕРНЕТИКА

А.В. АНИСИМОВ, М.С. ЯДЖАК

УДК 519.681.5

ПОСТРОЕНИЕ ОПТИМАЛЬНЫХ АЛГОРИТМОВ МАССОВЫХ ВЫЧИСЛЕНИЙ В ЗАДАЧАХ ЦИФРОВОЙ ФИЛЬТРАЦИИ

Ключевые слова: *оптимальный параллельно-конвейерный алгоритм, задача цифровой фильтрации, многоразовая каскадная цифровая фильтрация, квазисистолическая структура, ограниченный параллелизм, ускорение алгоритма.*

ВВЕДЕНИЕ

Проблема оптимизации вычислений при решении задач фильтрации рассматривалась во многих работах [1–4]. В большинстве случаев такие задачи необходимо решать в режиме реального времени при минимальных затратах оборудования. С этой целью предложено ряд параллельных алгоритмов [5–8], ориентированных на различные типы архитектур вычислительных систем [9], и сконструировано систолические структуры [10–14]. Известны также результаты, касающиеся решения задач фильтрации на основе использования однородных вычислительных сред [4, 15], программируемых логических интегральных схем [16] и нейронных сетей [17]. Разработанные алгоритмы и вычислительные средства обладали рядом недостатков: либо из-за «неэкономного» использования пересчитанных значений уменьшалась скорость сходимости вычислительного процесса, либо использовались не все резервы распараллеливания, в частности, оставались невостребованными естественные соотношения ассоциативности и коммутативности для операций, либо имели место задержки потоков данных.

Постоянное увеличение объема обрабатываемой в режиме реального времени информации приводит к глубокому анализу и оптимизации по быстродействию уже существующих и поиску новых методов и алгоритмов численного решения задач фильтрации. С учетом современных тенденций в микроэлектронике [18] авторами настоящей статьи были развиты методы Кунга–Лейзерзона [10] для реализации фильтров нерекурсивного типа и метод [11] реализации фильтра рекурсивного типа с целью организации систолических вычислений для решения одномерного варианта задачи цифровой фильтрации (ЗЦФ) [19] и задачи многоразовой каскадной цифровой фильтрации (ЗМКЦФ) [20, 21]. В результате получены систолические алгоритмы и оптимальные по загрузке реализующие их вычислительные структуры. С практической точки зрения особый интерес представляет проблема поиска в заданных классах оптимальных по быстродействию алгоритмов и разработки средств их непосредственной реализации. Один из весомых шагов в этом направлении сделан в [22], где предложен параллельно-конвейерный алгоритм (ПКА) решения одномерной ЗЦФ, оптимальный по быстродействию и использованию памяти в классе эквивалентных ему по информационному графу (ИГ) алгоритмов с точностью до выполнения законов ассоциативности и коммутативности для операции сложения. Данный алгоритм позволяет в достаточно общей постановке реализовать цифровую фильтрацию и может быть применен не только для обработки информации в различных предметных областях (анализ и обработка сигналов, экспериментальных данных, высказываний и т.д.), но и для исследования функционирования некоторых вычислительных систем, в частности квазисистолических структур (КСС) или нейронных сетей [23].

В настоящей работе представлены результаты, касающиеся построения и реализации оптимальных по быстродействию ПКА решений ЗЦФ и ЗМКЦФ более высоких размерностей.

ФОРМУЛИРОВКА ПРОБЛЕМЫ

Рассматриваемая ЗЦФ состоит в выполнении C пересчетов сглаживания массива N значений переменных через плавающее окно размера M.

В двумерном случае пересчет осуществляется по формуле

$$x_{i_1,i_2}=\sum_{s_1=-m_1}^{m_1}\sum_{s_2=-m_2}^{m_2}x_{i_1+s_1,i_2+s_2}f_{s_1,s_2};\ i_1=\overline{1,l_1};\ i_2=\overline{1,l_2}. \tag{1}$$

Здесь $M=(2m_1+1)(2m_2+1)$, $N=l_1l_2$, а весовые коэффициенты f_{s_1,s_2} $(s_1=\overline{-m_1,m_1};$ $s_2=\overline{-m_2,m_2})$ и «заграничные» значения $x_{1-m_1,j^*},x_{2-m_1,j^*},\dots,x_{0,j^*};$ x_{l_1+1,j^*}, $x_{l_1+2,j^*},\dots,x_{l_1+m_1,j^*}$ $(j^*=\overline{1-m_2,l_2+m_2});$ $x_{i_1,1-m_2},x_{i_1,2-m_2},\dots,x_{i_1,0};$ x_{i_1,l_2+1}, $x_{i_1,l_2+2},\dots,x_{i_1,l_2+m_2}$ $(i_1=\overline{1,l_1})$ — известные константы.

В n-мерном случае (1) имеет вид

$$x_{i_1,i_2,\dots,i_n}=\sum_{s_1=-m_1}^{m_1}\sum_{s_2=-m_2}^{m_2}\dots\sum_{s_n=-m_n}^{m_n}x_{i_1+s_1,i_2+s_2,\dots,i_n+s_n}f_{s_1,s_2,\dots,s_n};$$

$$i_1=\overline{1,l_1};\ i_2=\overline{1,l_2};\dots;\ i_n=\overline{1,l_n}.$$

Здесь $M=(2m_1+1)(2m_2+1)\cdot\ldots\cdot(2m_n+1)$, $N=l_1l_2\cdot\ldots\cdot l_n$. Аналогично формулируется одно- и трехмерная ЗЦФ.

Как известно [1], многоразовая каскадная фильтрация осуществляется последовательно набором действующих один за другим простых фильтров. Следовательно, рассматриваемая ЗМКЦФ состоит в выполнении r актов фильтрации; при каждом r'-м акте реализуется $C_1(r')$ каскадов, в каждом из которых осуществляется пересчет N_1 значений переменных через плавающее окно размера $M_1(v',r')$, где v' — номер каскада при r'-м акте фильтрации.

Одномерная задача предполагает для произвольного r'-го акта фильтрации выполнение вычислений в заданном v'-м каскаде по формуле

$$x_{i_1}=\sum_{s_1=-m_1(v',r')}^{m_1(v',r')}x_{i_1+s_1}f_{s_1}(v',r');\ i_1=\overline{1,N_1}. \tag{2}$$

В данном случае $M_1(v',r')=2m_1(v',r')+1$, а значения $x_{1-m_1(v',r')},x_{2-m_1(v',r')},\dots$ $\dots,x_0;$ $x_{l_1+1},x_{l_1+2},\dots,x_{l_1+m_1(v',r')}$ и весовые коэффициенты $f_{s_1}(v',r')$ $(s_1=\overline{-m_1(v',r'),m_1(v',r')})$ — заданные константы.

Для трехмерной ЗМКЦФ формула (2) принимает вид

$$x_{i_1,i_2,i_3}=\sum_{s_1=-m_1(v',r')}^{m_1(v',r')}\sum_{s_2=-m_2(v',r')}^{m_2(v',r')}\sum_{s_3=-m_3(v',r')}^{m_3(v',r')}x_{i_1+s_1,i_2+s_2,i_3+s_3}f_{s_1,s_2,s_3}(v',r');$$

$$i_1=\overline{1,l_1};\ i_2=\overline{1,l_2};\ i_3=\overline{1,l_3}.$$

Здесь $N_1 = l_1 l_2 l_3$, $M_1(v', r') = (2m_1(v', r') + 1)(2m_2(v', r') + 1)(2m_3(v', r') + 1)$. Аналогично формулируется и двумерная ЗМКЦФ.

Решаемая проблема состоит в разработке оптимальных по быстродействию и использованию памяти ПКА численного решения ЗЦФ размерности $n > 1$ и одно-, дву- и трехмерной ЗМКЦФ с последующей ориентацией этих алгоритмов для эффективной реализации на вычислительных системах специального (квазисистолических) и общего назначения.

Отметим, что в данном случае более слабое требование квазисистоличности отличается от требования систоличности тем, что из одной «инстанции» допускается одновременная рассылка данных сразу в несколько «точек приема». В соответствующих КСС это можно реализовать, используя, например, оптоэлектронные элементы.

Будем считать, что время выполнения операций сложения и умножения одинаково и равно одному такту, а единственным условием, требуемым для операций, является выполнение законов ассоциативности и коммутативности для операции сложения. Поэтому полученные в настоящей работе результаты верны и в случае, когда + и $*$ — любые другие операции, удовлетворяющие этому требованию: логические $\vee$ и $\wedge$, теоретико-множественные $\bigcup$ и $\bigcap$ или, например, max и min.

ОПТИМАЛЬНЫЕ АЛГОРИТМЫ ЧИСЛЕННОГО РЕШЕНИЯ ЗЦФ

Стандартный последовательный алгоритм выполнения C пересчетов сглаживания по формуле (1) имеет вид

$$\begin{aligned}
&\text{FOR } t = 1, C \text{ DO}\\
&\text{FOR } i_1 = 1, l_1 \text{ DO}\\
&\text{FOR } i_2 = 1, l_2 \text{ DO}\\
&p_1 = 0;\\
&\text{FOR } s_1 = -m_1, m_1 \text{ DO}\\
&\text{FOR } s_2 = -m_2, m_2 \text{ DO}\\
&p_1 = p_1 + x_{i_1+s_1, i_2+s_2} * f_{s_1,s_2};\\
&x_{i_1,i_2} = p_1.
\end{aligned} \tag{3}$$

Согласно алгоритму для пересчета переменной x_{i_1,i_2} на t-м шаге берутся значения $x_{i_1-m_1, i_2-m_2}$, $x_{i_1-m_1, i_2-m_2+1}, \ldots, x_{i_1-m_1, i_2}$, $x_{i_1-m_1, i_2+1}, \ldots$, $x_{i_1-m_1, i_2+m_2}$; $x_{i_1-m_1+1, i_2-m_2}$, $x_{i_1-m_1+1, i_2-m_2+1}, \ldots$, $x_{i_1-m_1+1, i_2}, x_{i_1-m_1+1, i_2+1}, \ldots, x_{i_1-m_1+1, i_2+m_2}; \ldots;$ $x_{i_1-1, i_2-m_2}, x_{i_1-1, i_2-m_2+1}, \ldots, x_{i_1-1, i_2}$, $x_{i_1-1, i_2+1}, \ldots, x_{i_1-1, i_2+m_2}$; $x_{i_1, i_2-m_2}, x_{i_1, i_2-m_2+1}, \ldots$ $\ldots, x_{i_1, i_2-1}$, которые являются уже пересчитанными на этом шаге.

Переход к параллельной версии алгоритма (3) осуществляем, используя квазисистолический метод (КСМ) [24] организации вычислений, согласно которому: 1) переменные пересчитываются в отдельных, сдвинутых между собой, ветвях; следовательно, главным приемом повышения степени параллелизма, кроме собственно распараллеливания, является конвейеризация; 2) допускается одновременная передача данных из одной «инстанции» сразу в M «точек приема»; 3) для пересчета значения произвольной переменной на некотором шаге используется максимальное количество уже пересчитанных на этом шаге значений; 4) функциональная эквивалентность с последовательным способом вычислений обеспечивается выполнением суммирования в определенном порядке, например, для пересчета x_{i_1,i_2} при $m_1 = 2$, $m_2 = 1$ он будет таким:

$$\begin{gathered}
x_{i_1,i_2} = x_{i_1,i_2} f_{0,0} + x_{i_1,i_2+1} f_{0,1} + x_{i_1-2,i_2-1} f_{-2,-1} + x_{i_1+1,i_2-1} f_{1,-1} + x_{i_1-2,i_2} f_{-2,0} + \\
+ x_{i_1+1,i_2} f_{1,0} + x_{i_1-2,i_2+1} f_{-2,1} + x_{i_1+1,i_2+1} f_{1,1} + x_{i_1-1,i_2-1} f_{-1,-1} + x_{i_1+2,i_2-1} f_{2,-1} + \\
+ x_{i_1-1,i_2} f_{-1,0} + x_{i_1+2,i_2} f_{2,0} + x_{i_1-1,i_2+1} f_{-1,1} + x_{i_1+2,i_2+1} f_{2,1} + x_{i_1,i_2-1} f_{0,-1};
\end{gathered}$$

5) каждое из пересчитанных значений используется $M-1$ ветвью, где оно является аргументом, на следующем же такте.

В случае, когда весовые коэффициенты равны между собой, т.е. $f_{-m_1,-m_2}=\ldots$ $\ldots=f_{0,0}=\ldots=f_{m_1,m_2}=f$, на основании КСМ предлагается ПКА решения двумерной ЗЦФ:

$$
\begin{aligned}
&\qquad\quad \text{FOR } i_1=1,l_1 \text{ DO SYNCH}\\
&\qquad\quad \text{DELAY } ((4m_2+2)(i_1-1))\\
&\qquad\quad \text{FOR } i_2=1,l_2 \text{ DO SYNCH}\\
&\qquad\quad \text{DELAY } (2i_2-2)\\
&\qquad\quad \text{FOR } t=1,C \text{ DO}\\
&\quad \text{BEGIN FOR } u=1,(2m_1+1)(2m_2+1)-1 \text{ DO}\\
&x_{i_1,i_2}=x_{i_1,i_2}+\text{IF } ((u/2)=[u/2]) \text{ THEN}\\
&\qquad\qquad\qquad x_{i_1-m_1+[(u-1)/(4m_2+2)],\, i_2-m_2+u/2-1-(2m_2+1)[(u-1)/(4m_2+2)]}\\
&\qquad\quad \text{ELSE } x_{i_1+[(u+2m_2+2)/(4m_2+2)],\, i_2+u/2+1/2-(2m_2+1)[(u+2m_2+2)/(4m_2+2)]}\\
&x_{i_1,i_2}=x_{i_1,i_2}*f;\\
&\text{DELAY}(2);\ \text{END}.
\end{aligned}
\tag{4}
$$

Первые два цикла в конструкции (4) обозначают одновременный запуск $l_1 l_2$ параллельных ветвей с синхронизацией их через каждую операцию. Оператор DELAY (p') осуществляет задержку на p' тактов. Цикл по переменной t соответствует выполнению C пересчетов сглаживания. Внутренний цикл определяет суммирование аргументов в нужном порядке. Далее результаты суммирования умножаются на f и пропускается два такта. После этого вычисляется следующая итерация. В приведенной конструкции и далее запись $[a]$ обозначает целую часть числа a.

Теорема 1. Алгоритм (4):

а) эквивалентен алгоритму (3) с точностью до соотношений ассоциативности и коммутативности;

б) использует переменные $x_{1-m_1,1-m_2}, x_{1-m_1,2-m_2}, \ldots, x_{1,1}, x_{1,2}, \ldots, x_{l_1+m_1,l_2+m_2}$ и только их;

в) имеет минимальное время реализации среди всех алгоритмов, удовлетворяющих пп. а) и б).

Доказательство этого утверждения детально изложено в [24] и основано на доказательстве двух вспомогательных утверждений. Первое из них говорит об отсутствии конкуренции над памятью при реализации алгоритма (4), второе — о минимальности высоты дерева этого алгоритма в классе всех эквивалентных деревьев.

Из теоремы 1 следует, что алгоритм (4) является оптимальным по быстродействию и использованию памяти (использует объем памяти, необходимый для ввода начальных данных) в классе эквивалентных по ИГ алгоритмов с точностью до выполнения соотношений ассоциативности и коммутативности.

В случае, когда весовые коэффициенты — разные между собой, ПКА (4) несколько изменится, поскольку в каждой из параллельных ветвей одновременно с суммированием происходит подготовка одного из последующих слагаемых, т.е. умножение на соответствующий весовой коэффициент. Для сохранения суммируемых величин для каждой ветви необходима, как минимум, одна дополнительная переменная. Соответствующую алгоритмическую конструкцию получаем из (4), заменив фрагмент между операторами BEGIN и END фрагментом

$$
\begin{aligned}
&\qquad \text{FOR}\quad u=1,(2m_1+1)(2m_2+1) \text{ DO}\\
&\text{BEGIN } x_{i_1,i_2}=\text{IF } (u=1) \text{ THEN } f_{0,0}*x_{i_1,i_2} \text{ ELSE } x_{i_1,i_2}+y_{i_1,i_2},\\
&\qquad y_{i_1,i_2}=f_{h_1(u,m_1,m_2),h_2(u,m_2)}*x_{i_1+h_1(u,m_1,m_2),i_2+h_2(u,m_2)} \text{ END};\\
&\qquad \text{DELAY } (2).
\end{aligned}
\tag{5}
$$

Здесь предполагается, что операторы $x_{i_1, i_2} = \ldots$ и $y_{i_1, i_2} = \ldots$, разделенные запятой, выполняются синхронно. Целочисленные функции $h_1(u, m_1, m_2)$, $h_2(u, m_2)$ — соответственно следующие условные операторы:

$$\text{IF } ((u/2) = [u/2]) \text{ THEN } [(u-1)/(4m_2+2)] - m_1 \text{ ELSE } [(u+2m_2+2)/(4m_2+2)],$$

$$\text{IF } ((u/2) = [u/2]) \text{ THEN } u/2 - m_2 - 1 - (2m_2+1)[(u-1)/(4m_2+2)]$$
$$\text{ELSE } u/2 + 1/2 - (2m_2+1)[(u+2m_2+2)/(4m_2+2)].$$

В работе [25] на основании предложенного КСМ построены оптимальные по быстродействию ПКА решения трехмерной ЗЦФ. Оптимальность доказана в соответствующих классах алгоритмов, эквивалентных по ИГ.

Последовательный алгоритм решения n-мерной ЗЦФ имеет вид

$$\begin{array}{l}
\text{FOR } t = \overline{1, C} \text{ DO} \\
\text{FOR } i_1 = \overline{1, l_1} \text{ DO} \\
\text{FOR } i_2 = \overline{1, l_2} \text{ DO} \\
\ldots\ldots\ldots\ldots\ldots\ldots \\
\text{FOR } i_n = \overline{1, l_n} \text{ DO} \\
p_1 = 0; \\
\text{FOR ALL } (s_1, s_2, \ldots, s_n) \in \{(r_1, r_2, \ldots, r_n) : r_1 = \overline{-m_1, m_1}; \\
\qquad r_2 = \overline{-m_2, m_2}; \ldots; r_n = \overline{-m_n, m_n}\} \text{ DO} \\
p_1 = p_1 + x_{i_1+s_1, i_2+s_2, \ldots, i_n+s_n} * f_{s_1, s_2, \ldots, s_n}; \\
x_{i_1, i_2, \ldots, i_n} = p_1.
\end{array} \tag{6}$$

Согласно (6) t-е приближение использует часть уже пересчитанных для этого приближения значений. Если более точно, то новое приближение используется при

$$(s_1 < 0) \vee (s_1 = 0 \wedge s_2 < 0) \vee$$
$$\vee (s_1 = 0 \wedge s_2 = 0 \wedge s_3 < 0) \vee \ldots \vee (s_1 = 0 \wedge s_2 = 0 \wedge \ldots \wedge s_{n-1} = 0 \wedge s_n < 0).$$

На основании КСМ [26] получаем ПКА решения n-мерной ЗЦФ:

$$\begin{array}{l}
t_1 = 2m_1 + 1;\ t_2 = 2m_2 + 1; \ldots; t_n = 2m_n + 1; \\
\text{FOR ALL } (i_1, i_2, \ldots, i_n) \in \{(i_1, i_2, \ldots, i_n): \\
\qquad i_1 = \overline{1, l_1};\ i_2 = \overline{1, l_2}; \ldots; i_n = \overline{1, l_n}\} \text{ DO SYNCH} \\
\text{DELAY } (2(t_2 t_3 \cdot \ldots \cdot t_n (i_1 - 1) + t_3 t_4 \cdot \ldots \cdot t_n (i_2 - 1) + \ldots + t_n (i_{n-1} - 1) + i_n - 1)) \\
\text{FOR } t = \overline{1, C} \text{ DO} \\
\text{BEGIN FOR } u = \overline{1, t_1 t_2 \cdot \ldots \cdot t_n} \text{ DO} \\
\text{BEGIN } x_{i_1, i_2, \ldots, i_n} = \text{ IF } (u = 1) \text{ THEN } f_{0,0,\ldots,0} * x_{i_1, i_2, \ldots, i_n} \\
\qquad \text{ELSE } x_{i_1, i_2, \ldots, i_n} + y_{i_1, i_2, \ldots, i_n}, \\
y_{i_1, i_2, \ldots, i_n} = f_{h_1^0, h_2^0, \ldots, h_n^0} * x_{i_1 + h_1^0, i_2 + h_2^0, \ldots, i_n + h_n^0} \text{ END;} \\
\text{DELAY (2); END.}
\end{array} \tag{7}$$

В алгоритме (7) операторы $x_{i_1, i_2, \ldots, i_n} = \ldots$ и $y_{i_1, i_2, \ldots, i_n} = \ldots$, разделенные запятой, выполняются в синхронном режиме. Целочисленные функции $h_1^0(u, m_1, m_2, \ldots, m_n)$, $h_2^0(u, m_2, m_3, \ldots, m_n), \ldots, h_n^0(u, m_n)$ — соответственно следующие условные операторы:

IF $((u/2)=[u/2])$ THEN $[(u-1)/(2t_2t_3\cdot\ldots\cdot t_n)]-m_1$

ELSE $[(u+1)/(2t_2t_3\cdot\ldots\cdot t_n)+1/2]$,

IF $((u/2)=[u/2])$ THEN

$[(u-1)/(2t_3t_4\cdot\ldots\cdot t_n)]-m_2-t_2[(u-1)/(2t_2t_3\cdot\ldots\cdot t_n)]$

ELSE $[(u+1)/(2t_3t_4\cdot\ldots\cdot t_n)+1/2]-t_2[(u+1)/(2t_2t_3\cdot\ldots\cdot t_n)+1/2]$,

……………………………………………………………………….

IF $((u/2)=[u/2])$ THEN $u/2-m_n-1-t_n[(u-1)/(2t_n)]$

ELSE $u/2+1/2-t_n[(u+1)/(2t_n)+1/2]$.

Теорема 2. Алгоритм (7):

а) эквивалентен алгоритму (6) с точностью до соотношений ассоциативности и коммутативности;

б) использует переменные только из набора

$$x_{1-m_1,1-m_2,\ldots,1-m_n},\ldots,x_{1,1,\ldots,1},\ldots,x_{l_1+m_1,l_2+m_2,\ldots,l_n+m_n};\ y_{1,1,\ldots,1},\ldots,y_{l_1,l_2,\ldots,l_n};$$

в) имеет минимальное время реализации среди всех алгоритмов, удовлетворяющих пп. а) и б).

Доказательство этого утверждения подробно изложено в [26].

На основании теоремы 2 можно утверждать, что алгоритм (7) является оптимальным по быстродействию в классе алгоритмов, эквивалентных по ИГ с точностью до выполнения соотношений ассоциативности и коммутативности.

Предложенные ПКА решения ЗЦФ ориентированы на реализацию в соответствующих квазисистолических вычислительных структурах [20].

ОПТИМАЛЬНЫЕ АЛГОРИТМЫ ВЫПОЛНЕНИЯ МНОГОРАЗОВОЙ КАСКАДНОЙ ФИЛЬТРАЦИИ

Последовательный алгоритм решения одномерной ЗМКЦФ имеет вид

FOR $u_1=1,r$ DO

FOR $p=1,C_1(u_1)$ DO

FOR $i_1=1,N_1$ DO

$t=0$; (8)

FOR $s_1=-m_1(p,u_1),m_1(p,u_1)$ DO

$t=t+f_{s_1}(p,u_1)*x_{i_1+s_1}$;

$x_{i_1}=t$.

Применяя КСМ организации вычислений для каждого акта фильтрации, получаем ПКА решения одномерной ЗМКЦФ:

FOR $i_1=1,N_1$ DO SYNCH

DELAY $(2i_1-2)$

FOR $u_1=1,r$ DO

FOR $p=1,C_1(u_1)$ DO

BEGIN FOR $s_1=1,2m_1(p,u_1)+1$ DO (9)

BEGIN $x_{i_1}=$ IF $(s_1=1)$ THEN $f_0(p,u_1)*x_{i_1}$ ELSE $x_{i_1}+z_{i_1}$,

$z_{i_1}=f_{h(s_1,m_1(p,u_1))}(p,u_1)*x_{i_1+h(s_1,m_1(p,u_1))}$ END;

DELAY (2); END.

В конструкции (9) операторы $x_{i_1}=\ldots$ и $z_{i_1}=\ldots$, разделенные запятой, выполняются синхронно, а функция $h(s_1,m_1(p,u_1))$ — условный оператор IF$((s_1/2)=[s_1/2])$ THEN $s_1/2-m_1(p,u_1)-1$ ELSE $(s_1+1)/2$.

Пусть Θ — класс алгоритмов, каждый из которых эквивалентен последовательному алгоритму (8) с точностью до выполнения соотношений ассоциативности и коммутативности и использует переменные только из набора $z_1, z_2, \ldots, z_{N_1}$; $x_{1-m_{10}}, x_{2-m_{10}}, \ldots, x_1, x_2, \ldots, x_{N_1}, x_{N_1+1}, \ldots, x_{N_1+m_{10}}$. Здесь $m_{10} = \max\{m_1(p, u_1)$: $p = \overline{1, C_1(u_1)}; u_1 = \overline{1, r}\}$.

Теорема 3. Алгоритм (9) принадлежит классу Θ и имеет минимальное время реализации среди всех алгоритмов этого класса.

Доказательство данного утверждения осуществляется аналогично доказательству теоремы в [22], к тому же необходимо учесть, что: а) результат пересчета переменной x_{i_1} при q-м акте фильтрации в p-м каскаде генерируется на такте с номером $2(m_1(1,1) + m_1(2,1) + \ldots + m_1(C_1(1),1) + m_1(1,2) + m_1(2,2) + \ldots + m_1(C_1(2),2) + \ldots$ $\ldots + m_1(1,q-1) + m_1(2,q-1) + \ldots + m_1(C_1(q-1),q-1) + m_1(1,q) + m_1(2,q) + \ldots + m_1(p,q)) +$ $+3(C_1(1) + C_1(2) + \ldots + C_1(q-1) + p) + 2i_1 - 4$; б) считывание значения этой переменной как аргумента осуществляется для всех $u = -m_1(p+1,q), -m_1(p+1,q)+1, \ldots$ $\ldots, 0, 1, \ldots, m_1(p,q)-1, m_1(p,q)$, если $p < C_1(q)$, или же для всех $u = -m_1(1,q+1)$, $-m_1(1,q+1)+1, \ldots, 0, 1, \ldots, m_1(C_1(q),q)-1, m_1(C_1(q),q)$, если $p = C_1(q)$, в ветвях $i_1 + u$.

Таким образом, ПКА (9) является оптимальным по быстродействию в классе алгоритмов, эквивалентных по ИГ с точностью до выполнения соотношений ассоциативности и коммутативности. Этот алгоритм предполагает изменение количества фильтров и их характеристик для каждого акта реализации каскадной фильтрации, что с практической точки зрения весьма важно.

Оптимальный по быстродействию ПКА численного решения двумерной ЗМКЦФ предложен в работе [27].

Последовательный алгоритм решения трехмерной ЗМКЦФ задается гнездом циклов

$$
\begin{aligned}
&\text{FOR } u_1 = 1, r \text{ DO} \\
&\text{FOR } p = 1, C_1(u_1) \text{ DO} \\
&\text{FOR } i_1 = 1, l_1 \text{ DO} \\
&\text{FOR } i_2 = 1, l_2 \text{ DO} \\
&\text{FOR } i_3 = 1, l_3 \text{ DO} \\
&t = 0; \\
&\text{FOR } (s_1, s_2, s_3) \in \{(r_1, r_2, r_3): r_1 = \overline{-m_1(p, u_1), m_1(p, u_1)}; \\
&\qquad r_2 = \overline{-m_2(p, u_1), m_2(p, u_1)}; r_3 = \overline{-m_3(p, u_1), m_3(p, u_1)}\} \text{ DO} \\
&t = t + f_{s_1, s_2, s_3}(p, u_1) * x_{i_1+s_1, i_2+s_2, i_3+s_3}; \\
&x_{i_1, i_2, i_3} = t.
\end{aligned}
\tag{10}
$$

Переход к функционально эквивалентной параллельной версии алгоритма (10) осуществляем с использованием КСМ [28]. В результате получаем ПКА решения трехмерной ЗМКЦФ:

$$
\begin{aligned}
&\text{FOR } i_1 = 1, l_1 \text{ DO SYNCH} \\
&\text{DELAY } ((4m_2(1,1)+2)(2m_3(1,1)+1)(i_1-1)) \\
&\text{FOR } i_2 = 1, l_2 \text{ DO SYNCH} \\
&\text{DELAY } ((4m_3(1,1)+2)(i_2-1)) \\
&\text{FOR } i_3 = 1, l_3 \text{ DO SYNCH} \\
&\text{DELAY } (2(i_3-1)) \\
&\text{FOR } u_1 = 1, r \text{ DO}
\end{aligned}
\tag{11}
$$

FOR $p = 1, C_1(u_1)$ DO

BEGIN FOR $s = 1, (2m_1(p, u_1)+1)(2m_2(p, u_1)+1)(2m_3(p, u_1)+1)$ DO

BEGIN x_{i_1,i_2,i_3} = IF $(s=1)$ THEN $f_{0,0,0}(p, u_1) * x_{i_1,i_2,i_3}$

ELSE $x_{i_1,i_2,i_3} + y_{i_1,i_2,i_3}$, $y_{i_1,i_2,i_3} =$

$= f_{h_1^*(s, m_1(p,u_1), m_2(p,u_1), m_3(p,u_1)), h_2^*(s, m_2(p,u_1), m_3(p,u_1)), h_3^*(s, m_3(p,u_1))}(p, u_1) *$

$* x_{i_1 + h_1^*(s, m_1(p,u_1), m_2(p,u_1), m_3(p,u_1)), i_2 + h_2^*(s, m_2(p,u_1), m_3(p,u_1)), i_3 + h_3^*(s, m_3(p,u_1))}$ END;

DELAY $(L(p, u_1))$; END.

Используемые в (11) целочисленные функции $h_1^*(s, m_1(p, u_1), m_2(p, u_1), m_3(p, u_1))$, $h_2^*(s, m_2(p, u_1), m_3(p, u_1))$, $h_3^*(s, m_3(p, u_1))$ и $L(p, u_1)$ определены в [28].

Пусть выполняются следующие равенства:

$$m_d(1,1) = m_d(2,1) = \ldots = m_d(C_1(1),1) = m_d(1,2) = m_d(2,2) = \ldots = m_d(C_1(2),2) = \ldots$$
$$\ldots = m_d(1,r) = m_d(2,r) = \ldots = m_d(C_1(r),r) = M_d; \ d = 2, 3. \qquad (12)$$

Рассмотрим Ξ — класс алгоритмов, эквивалентных алгоритму (10) с точностью до выполнения законов ассоциативности и коммутативности и таких, что используют переменные только из набора $x_{1-m_0,1-M_2,1-M_3}$, $x_{1-m_0,1-M_2,2-M_3}, \ldots$ $\ldots, x_{1,1,1}, \ldots, x_{l_1+m_0, l_2+M_2, l_3+M_3}$; $y_{1,1,1}, y_{1,1,2}, \ldots, y_{l_1,l_2,l_3}$. Здесь $m_0 = \max\{m_1(1,1), m_1(2,1), \ldots, m_1(C_1(1),1), \ldots, m_1(1,r), m_1(2,r), \ldots, m_1(C_1(r),r)\}$.

Теорема 4. Если выполняется условие (12), то алгоритм (11) принадлежит классу Ξ и имеет минимальное время реализации среди всех алгоритмов этого класса.

Доказательство данного утверждения приведено в [28].

Предложенные ПКА решения ЗМКЦФ ориентированы на реализацию на КСС [20, 28].

ОПТИМИЗАЦИЯ КВАЗИСИСТОЛИЧЕСКИХ ВЫЧИСЛЕНИЙ ПРИ РЕШЕНИИ ЗЦФ

Квазисистолические вычислительные структуры для реализации разработанных в предыдущих разделах оптимальных по быстродействию ПКА состоят из элементарных функциональных элементов, реализующих одну из двух операций: $+$, $*$; $\vee$, $\wedge$; $\bigcup$, $\bigcap$ или min, max. При этом в случае решения ЗЦФ максимальное количество используемых в этих элементах выходных каналов связи равно размеру плавающего окна. Следовательно, для каждого значения M необходимо проектировать отдельную вычислительную структуру, что невозможно сделать в типичных ситуациях. Более реальным выглядит случай, когда разработанный для произвольного M ПКА необходимо вложить в аппаратно реализованную схему с фиксированным количеством входных и выходных каналов каждого ее исполняющего элемента. Таким образом, с точки зрения практической реализации важной оказывается проблема уменьшения количества выходных каналов связи и оценки в этом случае быстродействия соответствующих ПКА. Следовательно, можно говорить о своеобразной задаче оптимизации числа коммутационных элементов, решение которой значительно увеличит возможности реализации этих алгоритмов на специализированных вычислительных системах.

Итак, организацию квазисистолических вычислений в ЗЦФ осуществляем при условии, что количество выходных каналов связи K является предварительно заданным, причем $1 < K < M$. Требуем, чтобы информационные графы соответствующих ПКА без ограничений $(K = M)$ и с ограничениями на число каналов совпадали.

В результате анализа потактовых схем оптимальных по быстродействию алгоритмов решения одно-, дву- и трехмерной ЗЦФ, учитывая регулярную структуру задаваемых ими вычислений, предложена техника [29, 30] определения возможных значений K для каждой задачи. Разработаны алгоритмические конструкции для оптимальной организации квазисистолических вычислений, отличающиеся от приведенных в [22, 24, 25] видом операторов задержки. Так, в случае решения одномерной ЗЦФ первый и второй из этих операторов имеют соответственно следующий вид: DELAY $(D(i_1 - 1))$, DELAY (L_0). В табл. 1 для каждого из возможных значений K определены целочисленные величины D и L_0, а в табл. 2 приведены численные оценки ускорения $S(K)$ [29] соответствующих ПКА решения одномерной задачи для разных значений m_1 в случае, когда весовые коэффициенты являются равными и $C = 5$, $N = 1000$. Видно, что при уменьшении числа выходных каналов связи от $(2m_1 + 1)$ до 2 ускорение уменьшается приблизительно в 1,5 раза. Здесь $M = 2m_1 + 1$.

Таблица 1

K	D	L_0
2	3	$2m_1 + 3$
3	3	$m_1 + 2$
$m_1 + 1$	2	3

При решении двумерной ЗЦФ три оператора задержки имеют соответственно вид: DELAY $(D_1(i_1 - 1))$, DELAY $(D_2(i_2 - 1))$, DELAY (L^*). Выражения для целочисленных величин D_1, D_2, L^* в зависимости от значения K приведены в табл. 3. В работе [30] определены два оператора задержки для трехмерной задачи и установлено, что в случае, когда весовые коэффициенты являются равными, уменьшение K от M до 2 приводит к уменьшению ускорения квазисистолических вычислений приблизительно в 1,53 раза при решении двумерной и в 1,85 раза при решении трехмерной ЗЦФ. Таким образом, полученные численные оценки ускорения предложенных ПКА решения ЗЦФ при условии $1 < K < M$ подтверждают целесообразность использования этих алгоритмов на практике.

Таблица 2

m_1	$S(2m_1 + 1)$	$S(m_1 + 1)$	$S(3)$	$S(2)$
3	17,14	17,11	11,46	11,40
6	31,38	31,32	21,00	20,81
9	45,21	45,13	30,29	29,91
15	71,72	71,59	48,13	47,19

Таблица 3

K	D_1	D_2	L^*
2	$6m_2 + 3$	3	$(2m_1 + 1)(2m_2 + 1) + 2$
3	$6m_2 + 3$	3	$2m_1m_2 + m_1 + m_2 + 2$
$2m_2 + 2$	$4m_2 + 3$	2	$2m_1 + 3$
$2m_1m_2 + m_1 + m_2 + 1$	$4m_2 + 2$	2	3

ПЕРЕХОД К ОГРАНИЧЕННОМУ ПАРАЛЛЕЛИЗМУ

Как было отмечено, описанные ПКА решения ЗЦФ ориентированы, как правило, на реализацию на специализированных вычислительных системах — КСС. При построении этих алгоритмов не накладывались никакие ограничения на количество параллельно выполняемых ветвей, которое равнялось числу пересчитываемых значений переменных. Однако, чтобы эффективно использовать предложенные алгоритмические конструкции для решения соответствующей задачи фильтрации на некоторой универсальной ЭВМ параллельного действия с заданным числом параллельно работающих процессоров, необходимо решать проблему перехода от неограниченного параллелизма к ограниченному. В связи со сложностью этой проблемы ее решение приведено для отдельных случаев задания числа параллельно выполняемых ветвей P при условии, что $[N/P] = N/P$, к тому же $1 < P < N$. Если весовые коэффициенты разные, то соответствующие алгоритмы с ограниченным параллелизмом (АОП) строятся с учетом равенства $[P/2] = P/2$.

Так, АОП решения одномерной ЗЦФ предлагаются для следующих случаев [31]:

1) $1 < P < m_1 + 1$, $m_1 > 1$; 3) $P = lm_1$, $m_1 > 1$, $l \in \mathbb{N} \setminus \{1\}$;
2) $1 < P < N$, $m_1 = 1$, $(N/P) \geq 3$; 4) $P = l(m_1 + 1)$, $m_1 > 1$, $l \in \mathbb{N}$.

Структура данных алгоритмов похожа на структуру соответствующих оптимальных по быстродействию алгоритмов из [22]. Основные отличия заключаются в следующем:

а) первый цикл АОП задает синхронное выполнение P (если весовые коэффициенты одинаковы) или $P/2$ (если весовые коэффициенты разные), а не N параллельных ветвей;

б) в зависимости от рассматриваемого случая первый оператор DELAY содержит соответствующее выражение для определения количества тактов задержки;

в) после цикла, задающего последовательное выполнение C пересчетов сглаживания, вводится новый цикл, определяющий переменные, значения которых пересчитываются в данной параллельной ветви;

г) второй оператор задержки отсутствует.

Например, в третьем случае при условии равенства весовых коэффициентов АОП имеет вид

$$\begin{array}{l} \text{FOR } v=1,P \text{ DO SYNCH} \\ \text{DELAY } (2v-2+w_P(v)(2m_1+1)) \\ \text{FOR } t=1,C \text{ DO} \\ \text{FOR } i_1=v+w_P(v)m_1,\, v+(w_P(v)+g_P)m_1,\, m_1 \text{ DO} \\ \text{BEGIN FOR } s=1,2m_1 \text{ DO} \\ x_{i_1}=x_{i_1}+\text{ IF } ((s/2)=[s/2]) \text{ THEN } x_{i_1+s/2-m_1-1} \text{ ELSE } x_{i_1+(s+1)/2} \\ x_{i_1}=x_{i_1}*f;\ \text{END.} \end{array} \tag{13}$$

В приведенной конструкции $g_P=N/P-1,\, w_P(v)=[(v-1)/m_1]g_P$.

Для решения двумерной ЗЦФ разработаны АОП для таких случаев:

1) $P=l_2$;

2) $P<\rho+1<l_2$;

3) $P=j_0\rho,\ P<l_2$;

4) $P=j_0(\rho+1),\ P<l_2$;

5) $P=P_1P_2,\ 1<P_1\le m_1,\ P_2=l_2$;

6) $P_1=l_1,\ P_2=\rho$;

7) $1<P_1\le m_1,\ P_2=\rho$;

8) $P_1=l_1,\ P_2=j_0\rho$;

9) $P_1=j_0m_1,\ P_2=l_2,\ m_1\ge 2$;

10) $P_1=j_0m_1,\ P_2=k_0\rho,\ m_1\ge 2$;

11) $P_1=j_0m_1,\ P_2=\rho,\ m_1\ge 2$.

Здесь $\rho=2m_1m_2+m_1+m_2;\ j_0, k_0\in\mathbb{N}\setminus\{1\}$.

Структура АОП для решения двумерной ЗЦФ определяется следующей конструкцией [32]:

$$\begin{array}{l} \text{FOR } v=1,A_1 \text{ DO SYNCH} \\ \text{DELAY } (D_{10}(v)) \\ \text{FOR } w=1,A_2 \text{ DO SYNCH} \\ \text{DELAY } (D_{12}(w)) \\ \text{FOR } t=1,C \text{ DO} \\ \text{FOR } i_1=I_1(v),I_2(v),I_3 \text{ DO} \\ \text{FOR } i_2=J_1(w),J_2(w),J_3 \text{ DO} \\ \qquad T(i_1,i_2). \end{array} \tag{14}$$

Если весовые коэффициенты одинаковы, то $A_1=P_1$, $A_2=P_2$, а $T(i_1,i_2)$ — фрагмент, состоящий из пяти последних строк конструкции (4), в котором оператор DELAY (2) заменен DELAY (D_{13}). В случаях 1–4 полагаем: $P_1=P$, $P_2=1$. Целочисленные величины $D_{10}(v)$, $D_{12}(w)$, $I_1(v)$, $I_2(v)$, I_3, $J_1(w)$, $J_2(w)$, J_3, D_{13} определены для каждого из рассмотренных выше случаев задания P. Например, в пятом случае они вычисляются по формулам [32]: $D_{10}(v)=(4m_2+2)(v-1)$, $D_{12}(w)=$ $=2w-2$, $I_1(v)=v$, $I_2(v)=v+l_1-P_1$, $I_3=P_1$, $J_1(w)=J_2(w)=w$, $J_3=1$, $D_{13}=0$.

В случае, если весовые коэффициенты разные, $T(i_1, i_2)$ в (14) представляется фрагментом, состоящим из заключенного между операторами BEGIN и END фрагмента (5), в котором оператор DELAY (2) заменен DELAY (D_{13}). При задании циклов по переменным v и w, а также вычислении $D_{10}(v)$, $D_{12}(w)$, $I_1(v)$, $I_2(v)$, I_3, $J_1(w)$, $J_2(w)$, J_3 учитывается, что $P = P_1P_2/2$. Кроме того, требуется выполнение следующих равенств: $[l_1/P_1] = l_1/P_1$, $[l_2/P_2] = l_2/P_2$.

В работе [32] предложены АОП для решения трехмерной ЗЦФ в отдельных случаях задания количества параллельно выполняемых ветвей.

Получены условия, при которых ускорение S_P предложенных АОП [31, 32] равно оптимальному (или близкому к нему) значению. Например, для алгоритма (13) при условии $C >> P$ получаем, что $S_P \approx P$. Если предположить, что $P = C(2m_1 + 1)$, т.е. $C = i_0 m_1$, $i_0 \in \mathbb{N}$, то при $C << N$ ускорение этого алгоритма вычисляется по формуле $S_P \approx (i_0^2 m_1^2 (2m_1 + 1)) / (3i_0 m_1 + i_0 - 1)$.

В случае, когда весовые коэффициенты одинаковы, для АОП решения двумерной ЗЦФ, построенного для пятого случая, получаем, что при $l_1 = l_2$ его ускорение выражается формулой $S_P \approx l_2 P_1 / (1 + 2m_1 / (C(2\rho + 1)))$. При $m_1 = 2$, $C = m_2 = 1$ получаем, что $S_P \approx 15 l_2 P_1 / 19$. С увеличением значений C, m_1, m_2 ускорение алгоритма будет приближаться к своему оптимальному значению.

Исследованы [33] также возможности реализации некоторых АОП на универсальных параллельных вычислительных системах с общей и распределенной памятью.

ЗАКЛЮЧЕНИЕ

В работе предложен КСМ организации вычислений при решении ЗЦФ произвольной размерности. Данный метод развит для решения одно-, дву- и трехмерной ЗМКЦФ. На основании КСМ построены оптимальные по быстродействию ПКА, ориентированные на реализацию на специализированных вычислительных системах — КСС. Возможности такой реализации значительно расширены за счет решения проблемы оптимизации квазисистолических вычислений при наложении ограничений на количество выходных каналов связи. Разработаны АОП для решения ЗЦФ на универсальных параллельных вычислительных системах и установлены условия, при которых ускорение этих алгоритмов равно своему оптимальному (или близкому к нему) значению.

Предложенные в работе алгоритмы могут быть применены не только для решения широкого спектра вычислительных задач (анализ и обработка плоских и пространственных изображений, исследование *n*-мерных объектов в некотором фазовом пространстве, обработка больших массивов экспериментальных данных и т.д.), но и для исследования, разработки и реализации вычислительных систем, например систолических, квазисистолических структур и нейронных сетей [23]. Корректность построенных алгоритмических конструкций подтверждена результатами моделирования их работы на ПЭВМ [34].

СПИСОК ЛИТЕРАТУРЫ

1. Ярославский Л.П. Цифровая обработка сигналов в оптике и голографии. Введение в цифровую оптику. — М.: Радио и связь, 1987. — 296 с.
2. Задирака В.К., Мельникова С.С. Цифровая обработка сигналов. — К.: Наук. думка, 1993. — 294 с.
3. Яцимірський М.М. Швидкі алгоритми ортогональних тригонометричних перетворень. — Львів: Академічний Експрес, 1997. — 219 с.
4. Тимченко О.В. Різницеві методи цифрової фільтрації. — Львів: Фенікс, 1999. — 388 с.
5. Lamport L. The parallel execution of DO loops // Comm. ACM. — 1974. — **17,** N 2. — P. 83–93.
6. Параллельная обработка информации. Параллельные методы и средства распознавания образов / Под ред. А.Н. Свенсона. — К.: Наук. думка, 1985. — Т. 2. — 280 с.
7. Люляков А.В. Построение плотного расписания для конвейерной вычислительной системы при цифровой фильтрации сигнала // Теоретические проблемы систем обработки информации. — Новосибирск: ВЦ СО АН СССР, 1986. — С. 72–85.

8. Мархівка В.С., Яцимірський М.М. Розпаралелювання програм швидких ортогональних перетворень у мультипроцесорних системах // Вісн. держ. ун-ту «Львівська політехніка». Комп'ютерна інженерія та інформаційні технології. — 1998. — № 349. — С. 21–26.
9. Алгоритмы, математическое обеспечение и архитектура многопроцессорных вычислительных систем / Под ред. А.П. Ершова. — М.: Наука, 1982. — 336 с.
10. Kung H.T., Leiserson C.E. Systolic arrays (for VLSI) // Proc. of the Symp. On Sparse Matrix Comput., Knoxville, 1978. — Philadelphia: SIAM, 1979. — P. 256–282.
11. Тищенко Т.П. О цифровой фильтрации на систолических структурах // Теория программирования и средства описания параллелизма дискретных систем. — Новосибирск: ВЦ СО АН СССР, 1985. — С. 90–103.
12. Кухарев Г.А., Тропченко А.Ю., Шмерко В.П. Систолические процессоры для обработки сигналов. — Минск: Беларусь, 1988. — 127 с.
13. Каневский Ю.С. Систолические процессоры. — Киев: Техника, 1991. — 173 с.
14. Костюнин А.Н. Применение систолических процессоров для адаптивной фильтрации сигналов // УСиМ. — 1991. — № 6. — С. 32–35.
15. Систолический процессор свертки / Под ред. Я.А. Дуброва. — Львов, 1991. — 73 с. — (Препр. / НТЦ «Интеграл»; № 5-91).
16. Иванов С.М., Тропченко А.Ю. Конвейерные разрядно-срезовые алгоритмы ранговой фильтрации и их реализация на ПЛИС // Proc. of Fifth Intern. Conf. «Pattern Recognition and Information Processing». — Minsk, 1999. — **2.** — P. 239–243.
17. Ліскевич О.І., Ліскевич Р.І., Яцимірський М.М. Алгоритмічний підхід до побудови нейронних мереж лінійної фільтрації сигналів // Праці міжнар. конф. з управління АВТОМАТИКА–2000, Львів, 11–15 вересня 2000 р., секція 7. — 2000. — Ч. 1. — С. 318–323.
18. Ли Т. Трехмерная микроэлектроника // Открытые системы. — 2002. — № 2 (http://www.osp.ru/os/2002/02/022.htm).
19. Яджак М.С. Оптимізація систолічних обчислень при розв'язанні задачі цифрової фільтрації // Теорія обчислень: Зб. наук. праць Ін-ту кібернетики ім. В.М. Глушкова НАН України. — К., 1999. — С. 386–390.
20. Яджак М.С. Деякі обчислювальні засоби реалізації алгоритмів цифрової фільтрації // Волин. мат. вісн. — 2002. — Вип. 9. — С. 90–99.
21. Яджак М.С. До питання організації систолічних обчислень під час виконання каскадної цифрової фільтрації // Там же. Сер. прикл. математики. — 2003. — Вип. 1(10). — С. 153–160.
22. Val'kovskii V.A. An optimal algorithm for solving the problem of digital filtering // Pattern Recognition and Image Analysis. — 1994. — **4,** N 3. — P. 241–247.
23. Вальковский В.А., Яджак М.С. Моделирование и реализация нейронных сетей на моделях параллельной обработки информации // Тр. междунар. конф. «Методы и средства преобразования и обработки аналоговой информации», Ульяновск (Россия), 8–10 июня 1999 г. — Ульяновск: УлГТУ, 1999. — **1.** — С. 53–55.
24. Вальковский В.А., Яджак М.С. Оптимальный алгоритм решения двумерной задачи цифровой фильтрации // Проблемы управления и информатики. — 1999. — № 6. — С. 92–102.
25. Яджак М.С. Об оптимальном в одном классе алгоритме решения трёхмерной задачи цифровой фильтрации // Там же. — 2000. — № 6. — С. 66–81.
26. Яджак М.С. Про оптимальність алгоритму чисельного розв'язання узагальненої задачі цифрової фільтрації // Волин. мат. вісн. — 2000. — Вип. 7. — С. 181–192.
27. Яджак М.С. Про чисельну реалізацію каскадної цифрової фільтрації // Вісн. Львів. ун-ту. Сер. прикл. математики та інформатики. — 2000. — Вип. 3. — С. 75–79.
28. Яджак М.С. О численном алгоритме решения пространственной задачи каскадной цифровой фильтрации // Проблемы управления и информатики. — 2004. — № 3. — С. 107–120.
29. Вальковський В.О., Яджак М.С., Полєтаєв С.М. До питання оптимізації квазісистолічних обчислень при розв'язанні задачі цифрової фільтрації // Комп'ютерна математика. Оптимізація обчислень: Зб. наук. праць Ін-ту кібернетики ім. В.М. Глушкова НАН України. — К., 2001. — **1.** — С. 77–82.
30. Яджак М.С. Організація квазісистолічних обчислень під час розв'язування задач цифрової фільтрації // Вісн. Львів. ун-ту. Сер. прикл. математики та інформатики. — 2002. — Вип. 5. — С. 178–183.
31. Яджак М.С. Алгоритмы с ограниченным параллелизмом для решения одной задачи цифровой фильтрации // Проблемы управления и информатики. — 2001. — № 6. — С. 109–118.
32. Яджак М.С. О построении алгоритмов с ограниченным параллелизмом для решения задач цифровой фильтрации // Там же. — 2002. — № 6. — С. 92–103.
33. Яджак М.С. Реалізація алгоритмів з обмеженим паралелізмом для розв'язання задачі цифрової фільтрації // Відбір і обробка інформації. — 2006. — Вип. 25(101). — С. 103–108.
34. Яджак М.С. Про моделювання алгоритмів з обмеженим паралелізмом для розв'язання задач цифрової фільтрації // Волин. мат. вісн. — 2001. — Вип. 8. — С. 105–109.

Поступила 10.01.2008