Практичне заняття № 7.
Паралельні алгоритми розв'язання двовимірної задачі цифрової фільтрації.

Формулювання двовимірної задачі фільтрації.

Загалом розглядувана нами задача цифрової фільтрації полягає у виконанні *C* переобчислень згладжування масиву значень *N* змінних через рухоме вікно розміром *M*.

У двовимірному випадку переобчислення згладжування виконуються за формулою (1).

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

У разі переобчислення значень x_{i_1, i_2} $(i_1 = \overline{1, l_1};\ i_2 = \overline{1, l_2})$ згідно з (1) одержуємо, що $N = l_1 l_2$, $M = (2m_1 + 1)(2m_2 + 1)$.

Звичайний послідовний алгоритм розв'язання сформульованої задачі має вигляд:

$$\begin{array}{ll}
 & FOR \quad t = 1, C \quad DO \\
\{ & FOR \quad i_1 = 1, l_1 \quad DO \\
\{ & FOR \quad i_2 = 1, l_2 \quad DO \\
\{ & p_1 = 0 \\
 & FOR \quad s_1 = -m_1, m_1 \quad DO \\
\{ & FOR \quad s_2 = -m_2, m_2 \quad DO \\
\{ & p_1 = p_1 + x_{i_1+s_1, i_2+s_2} * f_{s_1, s_2} \quad \} \quad \} \\
 & x_{i_1, i_2} = p_1 \quad \} \quad \} \quad \}.
\end{array} \tag{2}$$

Характерною особливістю алгоритму (2) є те, що для переобчислення значення змінної x_{i_1, i_2} на *t*-му кроці використовуються значення

$$x_{i_1-m_1, i_2-m_2}, x_{i_1-m_1, i_2-m_2+1}, \ldots, x_{i_1-m_1, i_2}, x_{i_1-m_1, i_2+1}, \ldots, x_{i_1-m_1, i_2+m_2};$$

$$x_{i_1-m_1+1, i_2-m_2}, \quad x_{i_1-m_1+1, i_2-m_2+1}, \ldots, \quad x_{i_1-m_1+1, i_2}, x_{i_1-m_1+1, i_2+1}, \ldots, x_{i_1-m_1+1, i_2+m_2};$$

$$\ldots; x_{i_1-1, i_2-m_2}, x_{i_1-1, i_2-m_2+1}, \ldots, x_{i_1-1, i_2}, x_{i_1-1, i_2+1}, \ldots, x_{i_1-1, i_2+m_2}; \ x_{i_1, i_2-m_2},$$

$$x_{i_1, i_2-m_2+1}, \ldots, x_{i_1, i_2-1},$$

які є вже також переобчисленими на цьому ж кроці.

Зазначена особливість алгоритму (2) дозволяє збільшувати швидкість процесу згладжування, тому її потрібно повністю або частково використати під час паралельної організації обчислень.

Для розв'язання сформульованої задачі фільтрації розглянемо і інший послідовний алгоритм (3).

$$
\begin{array}{ll}
 & FOR \quad t = 1,\ C \quad DO \\
\{ & FOR \quad i_1 = 1,\ l_1 \quad DO \\
\{ & FOR \quad i_2 = 1,\ l_2 \quad DO \\
\{ & p = 0 \\
 & FOR \quad s_1 = -m_1,\ m_1 \quad DO \\
\{ & FOR \quad s_2 = -m_2,\ m_2 \quad DO \\
\{ & p = p + x^{t-1}_{i_1+s_1,\ i_2+s_2} * f_{s_1,s_2} \quad \} \quad \} \\
 & x^{t}_{i_1,\ i_2} = p \quad \} \quad \} \quad \}.
\end{array}
\tag{3}
$$

У наведеній конструкції $x^0_{i_1,\ i_2}$, $x^t_{i_1,\ i_2}$ – відповідно початкове значення змінної $x_{i_1,\ i_2}$ та значення цієї ж змінної, переобчислене на t-му кроці.

Для свого виконання алгоритм (3) потребує час

$$T_1 = t_{op}(2m_1{+}1)(2m_2{+}1)Cl_1l_2 .$$

Тут t_{op} – час виконання подвійної операції додавання-множення.

В алгоритмі (3) для переобчислення значень змінних на t-му кроці використовуються значення, переобчислені виключно на (t-1)-му кроці.

Паралельні алгоритми фільтрації.

Паралельний режим обробки під час розв'язування двовимірної задачі фільтрації, пов'язаний із одночасним переобчисленням значень всіх змінних $x_{i_1,\ i_2}$ $(i_1 = \overline{1, l_1};\ i_2 = \overline{1, l_2})$, можна здійснити, наприклад, з допомогою алгоритму (4).

$$
\begin{array}{ll}
 & FOR \quad t = 1,\ C \quad DO \\
\{ & FOR \ \ ALL \ \ (i_1, i_2) \in \{(i_1, i_2):\ i_1 = \overline{1, l_1};\ i_2 = \overline{1, l_2}\} \quad DO \quad PAR \\
\{ & p_{i_1,i_2} = 0 \\
 & FOR \quad s_1 = -m_1,\ m_1 \quad DO \\
\{ & FOR \quad s_2 = -m_2,\ m_2 \quad DO \\
\{ & p_{i_1,i_2} = p_{i_1,i_2} + x_{i_1+s_1,\ i_2+s_2} * f_{s_1,s_2} \quad \} \quad \} \\
 & x_{i_1,\ i_2} = p_{i_1,i_2} \quad \} \quad \}.
\end{array}
\tag{4}
$$

У конструкції (4) тип паралелізму *PAR* може бути *SIM* або *CONC*. Тоді цей паралельний алгоритм реалізуватиме відповідно синхронний або асинхронний метод обчислень. У другому випадку кожне із значень x_{i_1, i_2} переобчислюватиметься незалежно від інших, використовуючи як аргументи поточні значення

$$x_{i_1 - m_1, i_2 - m_2}, \; x_{i_1 - m_1, i_2 - m_2 + 1}, \dots, \; x_{i_1, i_2}, \; x_{i_1, i_2 + 1}, \dots, \; x_{i_1 + m_1, i_2 + m_2}.$$

Для реалізації синхронного методу обчислень за (4) потрібен час

$$T_{s_1} = t_{op}(2m_1+1)(2m_2+1)C.$$

Використовуючи метод пірамід для розпаралелювання циклів, реалізацію синхронного методу обчислень у даному випадку можна здійснити з допомогою алгоритму (5).

$$\begin{array}{l}
FOR \;\; ALL \;\; (k_1, k_2) \in \{(k_1, k_2): \; k_1 = \overline{1, l_1}; \; k_2 = \overline{1, l_2}\} \quad DO \quad PAR \\
\quad \{ \quad FOR \;\; t = 1, \, C \quad DO \\
\qquad \{ \quad FOR \;\; i_1 = \max\{1, \, (t - C)m_1 + k_1\}, \, \min\{l_1, \, (C - t)m_1 + k_1\} \quad DO \\
\qquad \{ \quad FOR \;\; i_2 = \max\{1, \, (t - C)m_2 + k_2\}, \, \min\{l_2, \, (C - t)m_2 + k_2\} \quad DO \\
\qquad\quad \{ \quad p = 0 \\
\qquad\qquad FOR \;\; s_1 = -m_1, \; m_1 \quad DO \\
\qquad\qquad \{ \quad FOR \;\; s_2 = -m_2, \; m_2 \quad DO \\
\qquad\qquad\quad \{ \quad p = p + x^{t-1}_{i_1 + s_1, \, i_2 + s_2} * f_{s_1, s_2} \quad \} \\
\qquad\qquad\qquad x^{t}_{i_1, i_2} = p \quad \} \quad \} \quad \} \quad \} \quad \}.
\end{array} \tag{5}$$

У цьому разі *PAR* є *AUTON* і (5) задає паралельне виконання $l_1 l_2$ автономних гілок. Час виконання даного алгоритму обчислюється за формулою

$$T_{p_1} = t_{op}(2m_1+1)\,(2m_2+1)C(1 + ((2/3)m_1 m_2 (2C - 1) + m_1 + m_2)(C - 1)).$$

У (5) процес згладжування можна дещо покращити, якщо останні п’ять рядків замінити фрагментом:

$$\begin{array}{l}
\{ \quad p_{i_1, i_2} = 0 \\
\qquad FOR \;\; s_1 = -m_1, \; m_1 \quad DO \\
\{ \quad FOR \;\; s_2 = -m_2, \; m_2 \quad DO \\
\{ \quad p_{i_1, i_2} = p_{i_1, i_2} + x_{i_1 + s_1, \, i_2 + s_2} * f_{s_1, s_2} \quad \} \\
\qquad x_{i_1, i_2} = p_{i_1, i_2} \quad \} \quad \} \quad \} \quad \} \quad \}.
\end{array}$$

Одержана унаслідок цього паралельна конструкція дозволить у кожній гілці під час переобчислення значення деякої змінної на заданому кроці використовувати значення, що є вже переобчисленими на цьому ж кроці.

Алгоритми з обмеженим паралелізмом.

Далі розглянемо деякі паралельні алгоритми з обмеженою кількістю гілок P $(P < N)$ для розв'язання двовимірної задачі фільтрації. Зокрема, такий алгоритм, що реалізує синхронний метод обчислень, матиме вигляд:

$$
\begin{array}{l}
FOR \quad t = 1, C \quad DO \\
\{ \quad FOR \quad p = 1, P \quad DO \quad SIM \\
\{ \quad FOR \ ALL \ \ (i_1, i_2) \in \{(i_1, i_2): \ i_1 = \overline{J_1(p), J_2(p), \ J_3}; \ i_2 = \overline{1, l_2}\} \quad DO \\
\qquad \{ \quad p_{i_1, i_2} = 0 \\
\qquad \quad FOR \quad s_1 = -m_1, \ m_1 \quad DO \\
\qquad \{ \quad FOR \quad s_2 = -m_2, \ m_2 \quad DO \\
\qquad \{ \quad p_{i_1, i_2} = p_{i_1, i_2} + x_{i_1 + s_1, \ i_2 + s_2} * f_{s_1, s_2} \quad \} \quad \} \quad \} \quad \} \\
\qquad \quad FOR \ ALL \ \ (j_1, j_2) \in \{(j_1, j_2): \ j_1 = \overline{1, l_1}; \ j_2 = \overline{1, l_2}\} \quad DO \\
\qquad \{ \quad x_{j_1, j_2} = p_{j_1, j_2} \quad \} \quad \}.
\end{array} \tag{6}
$$

Запис $i_1 = \overline{J_1(p), J_2(p), \ J_3}$ означає, що змінна i_1 набуває значень від $J_1(p)$ до $J_2(p)$ з кроком J_3, де $J_1(p) = p, \ J_2(p) = p + (l_1 / P - 1)(m_1 + 1), \ J_3 = m_1 + 1$.

Даний алгоритм працює за умови, що l_1 є кратним до P, де $P = m_1 + 1$. Якщо знехтувати часовими затратами на синхронізацію паралельних гілок та реалізацію подвійного циклу за змінними $j_1, \ j_2$, то час виконання цього алгоритму обчислюється так:

$$T_{sob} = t_{op}(2m_1 + 1)(2m_2 + 1) l_1 l_2 C / P .$$

У наведеній конструкції (6) третій рядок можна замінити фрагментом:

$$\{ \qquad FOR \ ALL \ \ (i_1, i_2) \in \{(i_1, i_2): \ i_1 = \overline{1, l_1}; \ i_2 = \overline{J^1(p), J^2(p), \ J^3}\} \quad DO.$$

Тут $J^1(p) = p, \ J^2(p) = p + (l_2 / P - 1)(m_2 + 1)$, $J^3 = m_2 + 1$. При цьому l_2 має бути кратним до P, де $P = m_2 + 1$. Час роботи одержаного унаслідок такої заміни алгоритму буде співпадати з часом виконання алгоритму (6), якщо $m_1 = m_2$.

Обмежуючи кількість паралельно виконуваних гілок, алгоритм (5) можна записати у вигляді (7).

$$
\begin{array}{l}
\qquad FOR \quad ALL \quad (p, r) \in \{(p, r): p = \overline{1, P_1}; \ r = \overline{1, P_2}\} \quad DO \quad PAR \\
\quad \{ \quad FOR \quad t = 1, C \quad DO \\
\{ \ FOR \quad i_1 = \max\{1, (t - C)m_1 + (p - 1) l_1 / P_1 + 1\}, \min\{l_1, (C - t)m_1 + p l_1 / P_1\} \quad DO \\
\{ \ FOR \quad i_2 = \max\{1, (t - C)m_2 + (r - 1) l_2 / P_2 + 1\}, \min\{l_2, (C - t)m_2 + r l_2 / P_2\} \quad DO
\end{array}
$$

$$
\begin{array}{l}
\{ \quad p = 0 \\
\quad FOR \quad s_1 = -m_1,\ m_1 \quad DO \\
\quad \{ \quad FOR \quad s_2 = -m_2,\ m_2 \quad DO \\
\quad\quad \{ \quad p = p + x^{t-1}_{i_1+s_1,\, i_2+s_2} * f_{s_1, s_2} \quad \} \\
\quad\quad\quad x^{t}_{i_1, i_2} = p \quad \} \quad \} \quad \} \quad \} \quad \}.
\end{array}
\tag{7}
$$

У наведеній конструкції кількість паралельних гілок $P = P_1 P_2$, при цьому l_1 є кратним до P_1, а l_2 є кратним до P_2. Час виконання паралельного алгоритму (7) обчислюється за формулою

$$
T_{pob} = (l_1 l_2 /(P_1 P_2) + (m_1 l_2 / P_2 + m_2 l_1 / P_1)(C-1) + (2/3) m_1 m_2 (C-1)(2C-1)) \times \\
\times t_{op}(2m_1+1)(2m_2+1)C.
$$

Оцінювання прискорення паралельних обчислень.

Використовуючи оцінки часу виконання паралельних (4)–(7) та послідовного (3) алгоритмів, отримуємо відповідні оцінки прискорення паралельних обчислень. Зокрема, прискорення алгоритму (4), який реалізує синхронну схему, обчислюємо так:

$$
S_{(4)} = T_1 / T_{s_1} = l_1 l_2 ,
$$

тобто воно набуває свого оптимального значення за зробленого вище припущення стосовно синхронізації паралельних гілок.

Прискорення паралельного алгоритму (5) подається формулою:

$$
S_{(5)} = T_1 / T_{p_1} = l_1 l_2 /(1 + (2 m_1 m_2 (2C-1)/3 + m_1 + m_2)(C-1)).
$$

Зазвичай на практиці $l_1 >> m_1,\ l_2 >> m_2,\ l_1 >> C,\ l_2 >> C$ та l_1, l_2 відрізняються від $m_1 C^2,\ m_2 C^2$ $(l_1 > m_1 C^2,\ l_2 > m_2 C^2)$ відповідно не менш, як на декілька порядків, тому $S_{(5)}$ буде суттєвим. Наприклад, для $l_1 = 100,\ l_2 = 200,\ m_1 = m_2 = 1,\ C = 5$ отримуємо $S_{(5)} \approx 606$, а для $l_1 = l_2 = 1000,\ m_1 = m_2 = 5,\ C = 10$ маємо: $S_{(5)} \approx 340$.

Для алгоритму (6) прискорення подається формулою

$$
S_{(6)} = T_1 / T_{sob} = P ,
$$

тобто воно набуває оптимального значення за виконання зроблених вище припущень стосовно часу синхронізації паралельних гілок.

Прискорення паралельного алгоритму (7) обчислюємо за формулою

$$
S_{(7)} = T_1 / T_{pob} = \left(\frac{1}{P_1 P_2} + \left(\frac{m_1}{P_2 l_1} + \frac{m_2}{P_1 l_2} \right)(C-1) + 2 m_1 m_2 (C-1)(2C-1)/(3 l_1 l_2) \right)^{-1}.
$$

За зроблених припущень стосовно співвідношень між $l_1, l_2, m_1, m_2, C, P_1, P_2$ та m_1C, m_2C одержуємо, що $S_{(7)} \approx P_1P_2 = P$. Отже, прискорення алгоритму (7) у цьому разі є близьким до свого оптимального значення.

Вправа для самостійної роботи.

Записати варіант алгоритму з обмеженим паралелізмом для розв’язання двовимірної задачі цифрової фільтрації, який реалізує асинхронну схему обчислень. Обчислити складність цього алгоритму.