

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПОЕТАПНОГО ПЕРЕТВОРЕННЯ НА ОСНОВІ РЕКУРСИВНОГО ПІРАМІДАЛЬНОГО АЛГОРИТМУ

Схема поетапного перетворення порівняно з класичною дозволяє зменшити складність обчислень, що обумовлено повторним використанням отриманих даних на етапі високочастотного фільтрування. Для ефективною реалізації малохвильового перетворення основна увага в статті концентрується на використанні рекурсивного пірамідального алгоритму.

The lifting scheme reveals a lower computational complexity than the classical wavelet filters due to the reuse of the data delivered by the high-pass filtering stage. In this article the attention is concentrated on the recursive pyramid algorithm for fast wavelet transform.

1. ВСТУП

На даний час відомі чотири основні підходи до реалізації малохвильових перетворень двовимірних сигналів, зокрема, класичний стрічково-стовпцевий, лінійно-базовий або ущільнений лінійно-базовий, рекурсивний пірамідальний алгоритм та блочно-базовий або локально-малохвильового перетворення [1,2].

Практично усі ці алгоритми базуються на швидкому малохвильовому перетворенні і спрямовані на програмно-апаратну його реалізацію. Оскільки серед відомих алгоритм поетапного перетворення має низку переваг [1,2,3], то в даному випадку пропонується використувати його для перетворення двовимірних сигналів зображень.

У порівнянні з стрічково-стовпцевою реалізацією, рекурсивний пірамідальний алгоритм має вищу ефективність та швидкодію.

2. РЕКУРСИВНИЙ ПІРАМІДАЛЬНИЙ АЛГОРИТМ

Для подальшого покращання ступеня деталізації, з якою виділяються дані з кодера, в працях [3,4] був запропонований метод рекурсивного пірамідального алгоритму (РПА) швидкого малохвильового перетворення для VLSI архітектури. Основний принцип методу полягає в тому, що як тільки є доступною достатня кількість даних чи в горизонтальному, чи у вертикальному напрямках на різних малохвильових

¹ Національний університет “Львівська політехніка”

рівнях перетворення, активується малохвильове фільтрування. На рис.1 наведена ілюстрація цього методу для 1-D випадку, з якого видно, що коефіцієнти фільтрів низьких та високих частот завжди обчислюються як тільки це стає можливим. Тому затримка, з якою вивільняються малохвильові коефіцієнти, зменшена порівняно з багаторівневим лінійним перетворенням.

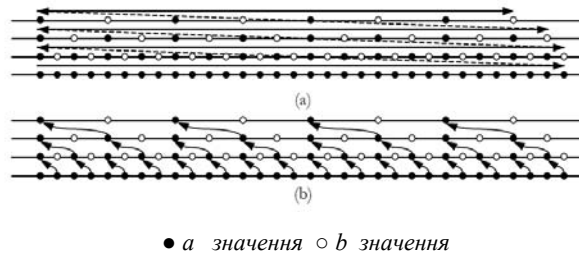


Рис. 1. Ілюстрація одновимірного (1D) стрічково-стовпцевого і лінійного методів фільтрування (a) в порівнянні з РПА (b), де задіяні інтенсивні перемикання між різними рівнями

Однак у випадку 2D сигналів необхідне здійснення буферування даних. Оскільки зображення сканується лінійним чином, то результати проміжного горизонтального фільтрування для різних малохвильових рівнів мають бути тимчасово збережені у буферній пам'яті. Така буферна пам'ять буде рівна буферній пам'яті багаторівневої лінійної реалізації [4]. Характеристики вхідної і вихідної пам'яті також залишаються без змін порівняно із багаторівневим лінійним методом (БЛМ). По-друге, якщо перенести запропоноване рішення на поетапну реалізацію з об'єднаними поетапними кроками, характеристики роботи пам'яті покращуються з двох причин: по-перше через вже згадане зменшення надлишковості поетапного методу і по-друге, навіть більш важливе, через зменшення кроку перемикання, обумовленого змішаним низько/високочастотним фільтруванням. Останнє, можливо, потребує більш детального пояснення: в об'єднаному методі фільтрування може активуватися лише кожними двома вибірками як у горизонтальному, так і вертикальному напрямках. Особливо це важливо у вертикальному напрямку, оскільки в цьому випадку вертикальне фільтрування буде задіяне лише кожних два рядки і звідси впливає, що також кожних два рядки буде відбуватися тільки горизонтальне фільтрування. В останньому випадку регістровий файл не повинен повністю перезавантажуватися з кожною вибіркою, як це відбувається для непарних рядків, наприклад W разів для вертикального фільтрування і $W/2$ разів для горизонтального (рис. 2).

3. ОЦІНКА ЧИСЛА ПЕРЕСИЛАНЬ ДАНИХ

Таким чином, опис звернення до пам'яті можна переписати як [2]:

$$T_{P2,R} = T_{P2 \rightarrow P1} = WH, \quad (1)$$

де $T_{P2,R}$ – число пересилань між основною пам'яттю $P2$, яка розташована поза кристалом мікросхеми і регістром процесора; $T_{P2 \rightarrow P1}$ – число пересилань між основною пам'яттю $P2$ і лінійним додатковим буфером $P1$; WH – добуток ширини зображення на висоту

$$\begin{aligned} T_{P1 \rightarrow R} &= \underbrace{2(M_p^F + M_u^F)H \sum_{i=1}^L 2^{1-i}}_{\text{Нейроодерживающая операция}} + \underbrace{\frac{I_L}{2} + (2M_{\max}^F + M_{\min}^F + 1) \frac{I_L}{4}}_{\text{Адресирование и управление}} \\ &+ \underbrace{(2M_{\max}^F + M_{\min}^F + 1) \frac{I_L}{2}}_{\text{Адресирование и управление}} \\ &= 2(M_p^F + M_u^F)H \sum_{i=1}^L 2^{1-i} + (6M_{\max}^F + 3M_{\min}^F + 5) \frac{I_L}{4} \end{aligned} \quad (2)$$

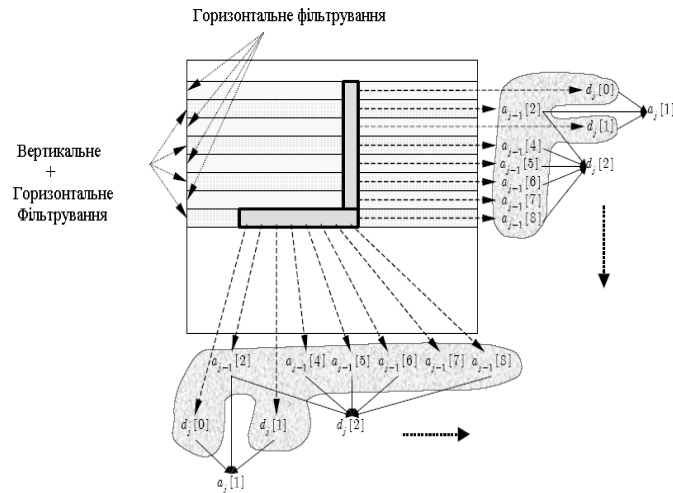


Рис. 2. Процес фільтрування і вимоги буферизації для 2D методу РПА

де M_p^F і M_u^F – довжини фільтрів прогнозування і оновлення відповідно; $M_{\max}^F = \max\{M_p^F, M_u^F\}$, а $M_{\min}^F = \min\{M_p^F, M_u^F\}$;

$I_L = WH \sum_{i=1}^L 4^{1-i}$; L – число рівнів поетапного перетворення.

Операція зворотного запису з регістрового файлу в буфер пам'яті $P1$ тепер містить додатковий доданок через неперервне перемикання між горизонтальним і вертикальним фільтруванням кожного з двох рядків (це пояснює наявність у наступному виразі $H/2$ компоненти):

$$T_{R \rightarrow P1} = 2M_u \left(\frac{H}{2} + W \right) \sum_{i=1}^L 2^{1-i} + 2I_L \quad (3)$$

$$T_{P2,W} = T_{P1 \rightarrow P2} = WH \quad (4)$$

$$T_{P1,R} = T_{P1 \rightarrow R} + T_{P1 \rightarrow P2} = 2(M_p^F + M_u^F)H \sum_{i=1}^L 2^{1-i} +$$

$$+ \left(6M_{\max}^F + 3M_{\min}^F + 5 \right) \frac{I_L}{4} + WH \quad (5)$$

$$T_{P1,W} = T_{R \rightarrow P1} + T_{P2 \rightarrow P1} = 2M_u \left(\frac{H}{2} + W \right) \sum_{i=1}^L 2^{1-i} + 2I_L + WH \quad (6)$$

4. ВИСНОВОК

Як впливає з рівностей (1)–(6), рекурсивний пірамідальний алгоритм, при використанні однакового об'єму пам'яті, генерує дані з меншою затримкою ніж багаторівневий лінійний алгоритм. Такий ефект відбувається за рахунок більшого завантаження шини даних між регістровим файлом і пам'яттю лінійного додаткового буфера $P1$. Таким чином, можна зробити висновок, що запропонована архітектура ефективніша для VLSI спрямованих розробок.

1. Воробьев В.И., Грибунин В.Г. Теория и практика вейвлет-преобразования // ВУС, С.-Петербург, 1999. – 204 с. 2. Наконечный А.И. Підвищення ефективності поетапного перетворення двовимірних сигналів // Міжвідомчий збірник наукових праць Національної АН України «Відбір і обробка інформації». 2006. Вип.24. – С.87 – 92. 3. Sweldens W. "Lifting Scheme: A New Philosophy in Biorthogonal Wavelet Constructions" *Proceedings of SPIE Wavelet Applications in Signal and Image Processing III. – September 1995 – vol. 2569. – P. 68-79.* 4. Sweldens W. "The Lifting Scheme: A custom-design construction of biorthogonal wavelets" // *Applied and Computational Harmonic Analysis. – 1996. – №2. – P.186 - 200.*