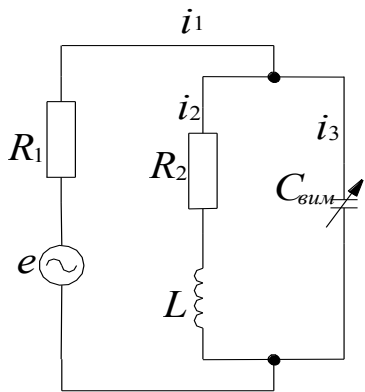


$$C_{\text{вим}} = \left| \frac{4L + R_1}{4\omega^2 L^2 + R_2^2} \right|, \quad (11)$$

де  $\omega$  – виміряна кутова частота.



$e$  – джерело змінної наруги високостабільної частоти,  $L$  – індуктивність,  $C_{\text{вим}}$  – ємність,  $R_1$  – вихідний опір джерела живлення,  $R_2$  – активний опір котушки індуктивності

Рис. 2. Еквівалентна схема ємнісного перетворювача

З урахуванням (11) визначимо зміну ємності  $\Delta C$  ємнісного перетворювача від зміни його кута нахилу

$$\Delta C = \frac{4L + R_1}{4\omega_0^2 L^2 + R_2^2} - \frac{4L + R_1}{4\omega_\alpha^2 L^2 + R_2^2}, \quad (12)$$

де  $\omega_0$  – кутова частота перетворювача, яка визначена, в початковий момент його встановлення;  $\omega_\alpha$  – кутова частота, визначена після нахилу перетворювача.

Таким чином, ми отримали значення зміни ємності від величини кута нахилу перетворювача. Оскільки виміряти безпосередньо ємність з високою точністю складно, то ми визначаємо її шляхом вимірювання частоти коливального контуру, елементом якого є вимірювальна ємність. Дана рішення використане при проведенні контролю величини кута нахилу технологічних об'єктів Краснопопівського підземного сховища газу в реальному часі.

1. Бугров А.В. Високочастотные емкостные преобразователи и приборы контроля качества. - М.: Машиностроение, 1982.-194 с. 2.Каллакорт Р.Диагностика поврежденных. -М.: Мир,1989.- 516 с.

УДК 620.179

## АПАРАТНА РЕАЛІЗАЦІЯ ШВИДКОГО ЗВОРОТНОГО ПРОЕКТУВАННЯ У ТРИВИМІРНІЙ РЕКОНСТРУКЦІЇ ЗОБРАЖЕНЬ

© Кабанова О.В., 2003

Івано-Франківський національний університет нафти й газу

*Представлені дві можливі апаратні реалізації алгоритму швидкого зворотного проектування, які дозволяють чітко організувати і оптимізувати обчислення в даному алгоритмі. Розглянуто організацію обчислень основного кроку алгоритму, який містить найбільше операцій обчислень. Для спрощення оптимізованим об'ємом ПЗУ, який необхідний для реалізації алгоритму, буде знехтувано.*

Алгоритми двовимірної томографії призначені для відновлення зрізу контрольованого об'єкта. Якщо необхідна реконструкція об'ємного сегмента, то вся процедура буде складатися з відновлення послідовних двовимірних перерізів, отриманих за допомогою руху з малою швидкістю об'єкта чи системи джерело-детектор. Для більш ефективного отримання даних для об'ємної комп'ютерної томографії використовується конусна схема

сканування і двовимірний детектор. Промені формують пучок у формі конуса з центром на детекторі і вершиною на джерелі. Джерело рентгенівського випромінювання на практиці випромінює конусний пучок променів, так що використання конусного пучка не тільки збільшує швидкість сканування, але й дозволяє ефективніше використовувати промені, які у в'язовій і паралельній схемах сканування вилучаються коліматором.

Тривимірні алгоритми реконструкції поділяються на точні і неточні алгоритми. При незашумлених проєкційних даних, зафіксованих з достатньою щільністю детектором, точний алгоритм здатний реконструювати об'єкт при будь-якому значенні допустимого відхилення між функціями об'єкта і відновленого зображення.

На основі незашумлених даних точний алгоритм реконструкції дозволяє отримати в результаті зображення, в якому відхилення від реального об'єкта може бути зведене до мінімуму шляхом збільшення роздільної здатності детектора і кількості проєкцій. Один з перших і дотепер найточніших алгоритмів для тривимірної томографії є FDK-алгоритм [1]. Даний алгоритм як складовий етап містить тривимірне зворотне проєктування, яке, в принципі, є найбільш тривалою і об'ємною за обчисленням частиною алгоритма. Комплексність FDK-алгоритму визначається комплексністю зворотного проєктування.

Пропонується оптимізована модифікація даного алгоритму, в якому зворотне проєктування виконується за  $O(N^3 \ln N)$  операції замість  $O(N^4)$  операції, які притаманні традиційному зворотному проєктуванню. Реалізація алгоритму вимагає  $O(N^3)$  введень значень у пам'ять і за попередніми підрахунками отримується зображення з якістю одного порядку як і у традиційному FDK-методі.

Для реконструкції вокселя зображення  $(x, y, z)$  необхідно виконати зворотне проєктування над проєкційними даними від всіх променів, що перетнули даний воксель. У проєкційному просторі зворотне проєктування може розглядатися як процес сумування значень вздовж синусоїдної кривої. Ідея швидкого зворотного проєктування полягає в тому, щоб суму

$$f(x, y, z) \approx \frac{2\pi}{N_\theta} \sum_{i=0}^{N_\theta-1} \tilde{p} [k(x, y, \theta), m(x, y, z, \theta)]$$

розбити на проміжні суми, які можуть використовуватися для обчислення інших точок у наступних кроках. За стратегією "прискорення" алгоритму таке сумування вздовж синусоїди апроксимовується сумою коротких кривих (сегментів синусоїди), які обчислюються з попередньо обчислених коротших і так далі. В [2] вперше застосовано термін "зв'язки" для таких кривих. Довжина зв'язку визначається як різниця  $\theta$ -координат між проєкційними значеннями його кінцевих точок. Усі зв'язки мають свої кінцеві точки на сітці перепакованого тривимірного проєкційного простору. Суть алгоритму – це обчислення значення будь-якого зв'язку як суми восьми зв'язків з вдвічі коротшою довжиною.

Для реалізації алгоритму необхідно здійснити деякі попередні обчислення. Спочатку для вокселів, які будуть відновлюватися, потрібно визначити, які  $\pi/2$ -зв'язки є необхідними для обчислення їх значень. Індеси цих зв'язків зводяться в таблицю. В наступному кроці визначаються і зводяться в таблицю  $\pi/4$ -зв'язки, які є необхідними для виконання головного кроку на основі вже зведених у таблицю  $\pi/2$ -зв'язків. Такі операції мають бути виконані для всіх зв'язків, закінчуючи табуляцією необхідних 2-зв'язків (найкоротших), поступово визначаючи необхідні зв'язки, рухаючись від більш довгих до менших зв'язків. Переважно в цей самий час, вагові коефіцієнти інтерполяційної функції також заносяться в таблицю. Необхідно звернути увагу на те, що всі ці значення залежать тільки від схеми сканування, а не від реконструйованого об'єкта, і їх достатньо попередньо обчислити один раз і назавжди.

Обчислення значення зв'язку спрощується до мінімуму як тільки виготовлені таблиці. Дане обчислення і визначає сам алгоритм, який здійснюється за так званий "час виконання". Починаючи з самих коротких зв'язків і рухаючись до більших, даний час не включає час обчислення середньої точки або вагових коефіцієнтів інтерполяційної функції, а тільки обчислення основного кроку.

Для реалізації алгоритму необхідні дві основні області пам'яті: перша область для обчислених значень зв'язку, друга – для таблиці конструкції зв'язку, в якій містяться коефіцієнти інтерполяції і адреси значень зв'язку. В якості першої області може виступати операційна пам'ять (ОЗП), а для іншої – постійна пам'ять (ПЗП).

Значення зв'язків, що створюються в кожному  $\lg N_\theta - 2$  кроці алгоритму, необхідні тільки як вхідні дані для наступного кроку і можуть бути відкинуті. Отже, операційна пам'ять, що зберігає ці значення може бути розбита на дві частини: одна достатньо велика для збереження значень, обчислених в попередньому кроці, і іншу, досить велику, для поточних обчислень. Після того, як один крок виконаний, вміст першої частини пам'яті стає непотрібним і може використовуватися для наступного кроку. Досліджено, що кількість зв'язків зменшується з кожним наступним кроком. Отже необхідний об'єм пам'яті визначається кількістю 2-зв'язків, яка завжди рівна  $O(N^3)$ .

Пропонується два можливих варіанта апаратної архітектури для реалізації алгоритму. Розглянемо основний крок алгоритму, припустивши, що фільтрування, перепаковка і обчислення значень вокселів з  $\pi/2$ -зв'язків вже виконані.

Для спрощення припускаємо, що структурні зв'язки (ті, що будуть використовуватися для побудови довгих зв'язків) і вагові коефіцієнти

інтерполяційної функції вже занесені в пам'ять.

Перший варіант апаратної архітектури зображений на рис. 1.

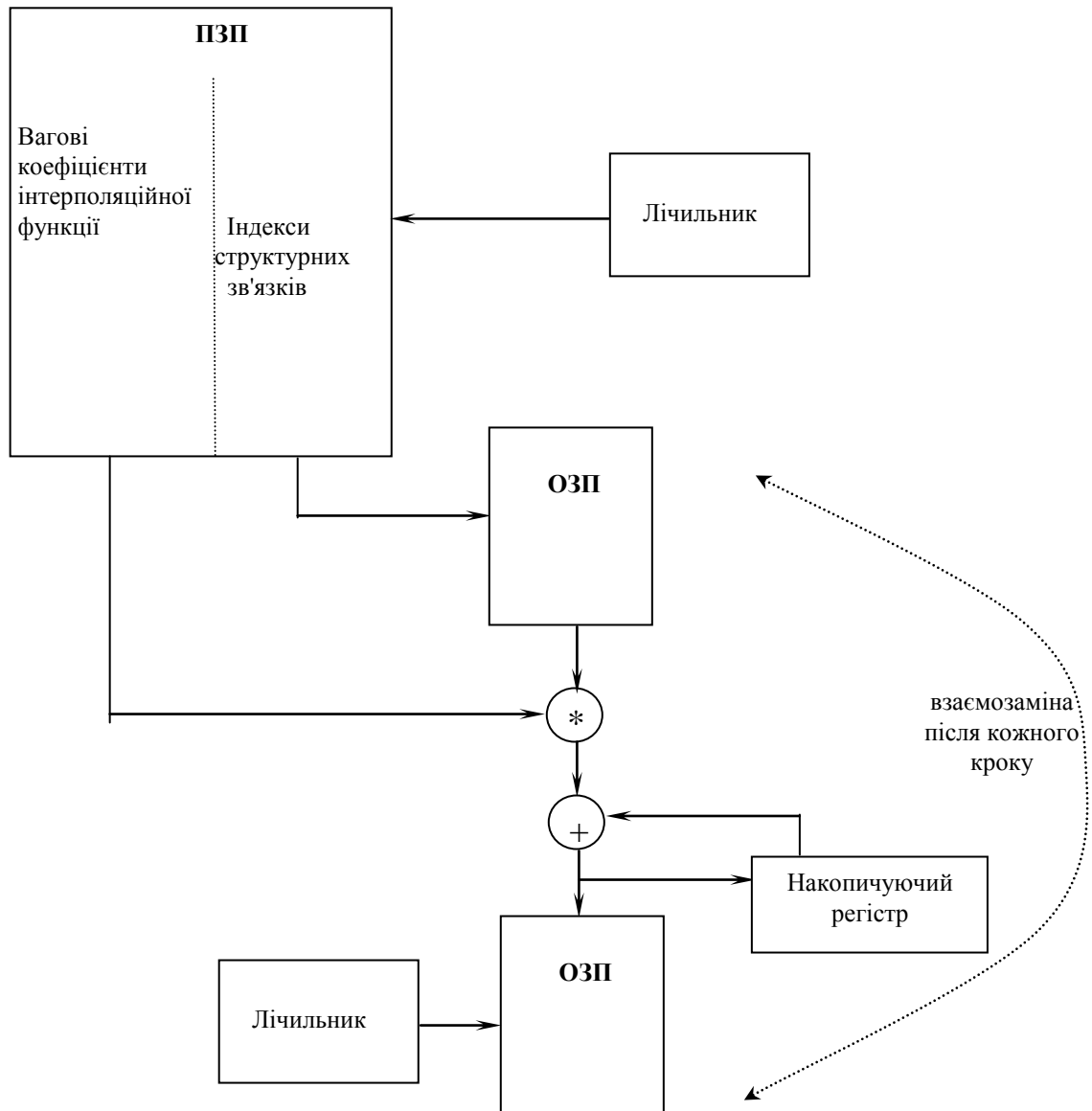


Рис. 1. Спрощена апаратна архітектура, що виконує основну частину алгоритму зворотного проектування

Попередньо обчислені значення структурних зв'язків і вагові коефіцієнти інтерполяційної функції зберігаються у постійній пам'яті з індексами зв'язків і відповідними їм адресами вагових коефіцієнтів. Таким чином, кожний основний крок використовує вісім значень з пам'яті. Дані вісім значень зв'язку необхідно лише посортувати за довжиною (внутрішнє розташування зв'язків однієї довжини є довільним). Значення зв'язку зберігається в операційній пам'яті, яка розділена на дві половини. Спочатку верхня половина операційної пам'яті

заповнюється профільтованими проекційними даними. Далі в кожному часовому циклі значення структурного зв'язку нормується і додається. В кожній восьмій ітерації отримуємо повне нове значення зв'язку, після чого накопичуючий регістр пам'яті очищується, а лічильник адресації в нижній частині операційної пам'яті збільшується на одиницю. Після одного кроку алгоритму, тобто коли значення всіх зв'язків певної довжини обчислені, верхня і нижня частини операційної пам'яті взаємно міняються. Цю взаємозаміну бажано здійснювати за

допомогою адресної логіки операційної пам'яті, а не переносом дійсних даних. Коли операцію завершено, значення всіх  $\pi/2$ -зв'язків зберігаються у нижній частині ОЗП, вони готові до застосування у обчисленні готового значення вокселя, яке буде здійснюватися іншим апаратним або програмним забезпеченням.

Вміст постійної пам'яті зчитується у послідовному порядку  $i$ , таким чином, може зберігатися на якомусь додатковому пристрої, наприклад, жорсткому диску. Недоліком даної схеми є те, що для обчислення кожного значення зв'язку необхідно вісім часових циклів і всі коефіцієнти інтерполяційної функції зберігаються двічі в постійній пам'яті.

Розглянемо детально обчислення одного значення зв'язку за часовий цикл.

При двоточковій лінійній інтерполяції завжди використовуються дві точки (зліва і справа від бажаної точки) на сітці дискретизації. Іншими

словами використовується точка з парним і непарним індексом. Якщо узагальнити дане спостереження для основного кроку, можна помітити, що вісім коротких зв'язків довжиною  $l$  завжди можна поділити на наступні вісім непов'язаних класів:

$$\theta_{n_1} / l - \text{парне}, k_2 - \text{парне}, t_2 - \text{парне};$$

$$\theta_{n_1} / l - \text{парне}, k_2 - \text{парне}, t_2 - \text{непарне};$$

$$\theta_{n_1} / l - \text{парне}, k_2 - \text{непарне}, t_2 - \text{парне};$$

⋮

$$\theta_{n_1} / l - \text{непарне}, k_2 - \text{непарне}, t_2 - \text{непарне}$$

по одному зв'язку в кожному класі.

Таким чином можливо кожен половину операційної пам'яті, що призначена для зберігання значень зв'язків, поділити на ті самі вісім класів і виконати завершений крок інтерполяції за один часовий цикл (рис.2).

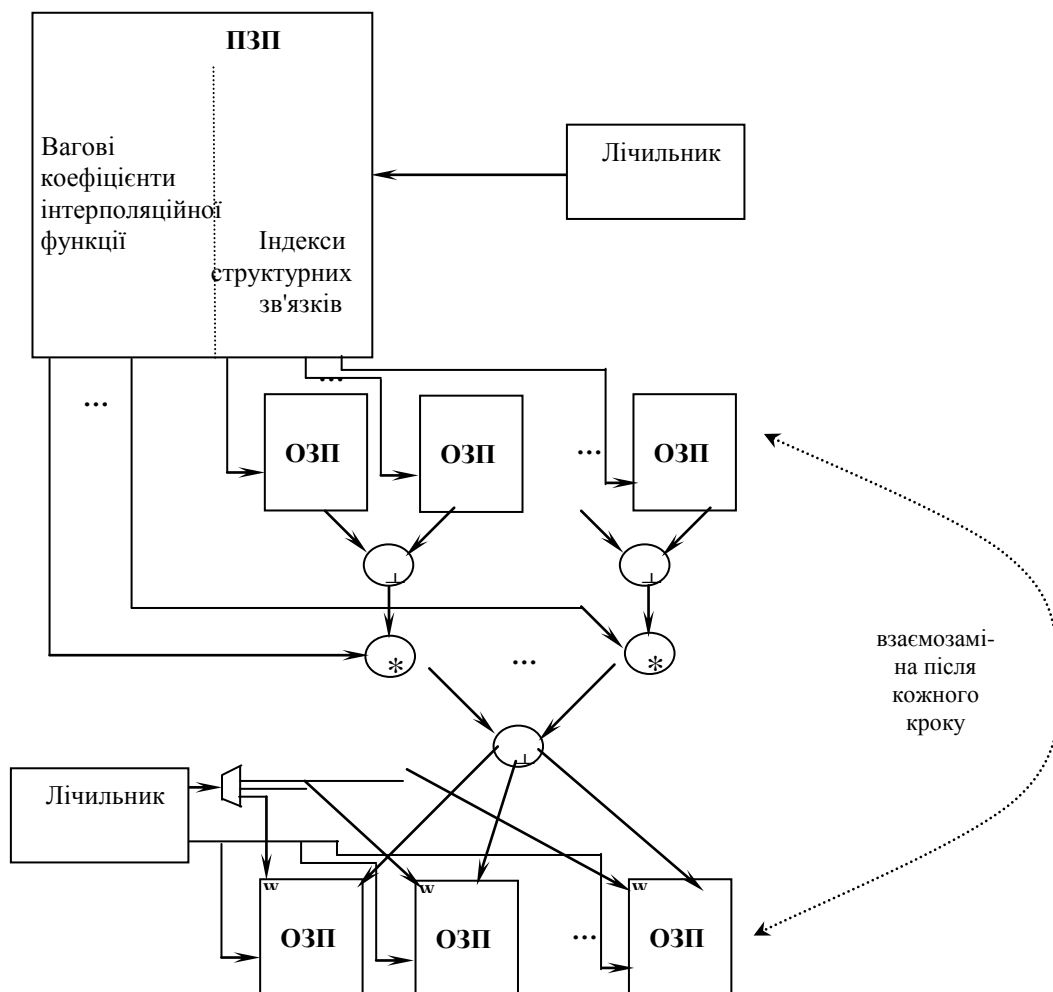


Рис.2. Апаратна реалізація обчислення одного значення зв'язку за один часовий цикл

Нові значення зв'язку зберігають у відповідній класу нижньої половини ОЗП. Рішення щодо відповідності секції можна прийняти попереднім сортуванням зв'язків у постійній пам'яті у зручному порядку так, щоб три найменш важливі біти лічильника адресації нижньої частини ОЗП могли повторюватися вісьма секціями, як показано на рис.2. Кожний крок  $i = 2, \dots, \lg N_\theta - 1$  лічильник адресації постійної пам'яті буде збільшуватися в кожному  $N_\theta 2^{1-i}$  часовому циклі.

Розглянемо значення зв'язку в перших чотирьох сегментах ОЗП і припустимо, що попереднє сортування в ОЗП здійснюється так, що вони сортуються за нисхідним пріоритетом  $\theta_{n_1}, t_{k_1}, q_{m_1}, t_{k_2}, q_{m_2}$ . Тоді чотири перші значення структурного зв'язку матимуть приблизно однакові значення для даного зв'язку. Таким чином, дуже ефективно було б зберігати різницю, зміщення першого індексу зв'язку з індексами 2, 3 і 4 замість їх дійсних значень. Таку ж схему можна застосовувати для останніх чотирьох сегментів ОЗП, якщо вони сортувалися за нисхідним принципом  $\theta_{n_2}, t_{k_2}, q_{m_2}, t_{k_1}, q_{m_1}$ .

Реконструювати об'ємне зображення можна поділивши його в середній площині на дві частини і використовуючи проєкційні дані з верхньої і нижньої половин проєкційних рядів відповідно. Але обчислені значення  $\pi/2$ -зв'язків з коренями в різних  $\theta$ -координатах розділити неможливо. Отже, проєкційні дані можна поділити на такі вісім класів (груп):

$$0 \leq \theta < \frac{\pi}{2}, q < 0;$$

$$0 \leq \theta < \frac{\pi}{2}, q \geq 0;$$

$$\frac{\pi}{2} \leq \theta < \pi, q < 0;$$

$$\vdots$$

$$\frac{3\pi}{2} \leq \theta < \pi, q \geq 0$$

з апаратним забезпеченням, що буде виконувати основну частину алгоритму зворотного проєктування по кожній з груп паралельно.

Якщо отримання проєкційних значень і реконструкцію розглядати як єдину систему, то швидкодію обробки даних знижує їх перепаківка до даних від паралельної схеми сканування. Першу паралельну проєкцію можна отримати тільки після повороту джерела на кут  $2\gamma_{\max}$  і надалі швидкість обробки даних є сталою. Швидке зворотне проєктування можна починати одразу після

визначення квадратну даних від паралельної схеми сканування.

Головною метою попереднього розрахунку ПЗП було спрощення й оптимізація "обрізки" (відкидання) непотрібних зв'язків. Саме дані в ПЗП для даної операції вимагають нисхідну побудову дерев, тобто визначення зв'язків від більших до менших. Розглянемо апаратне забезпечення, яке дозволяє зменшити розмір ПЗП шляхом зберігання інформації для "обрізки", обчислених вагових коефіцієнтів інтерполяційної функції та адрес значень зв'язку в процесі обчислення.

Інформація для виконання "обрізки" – це значення одного біту пам'яті для кожного можливого зв'язку, яке вказує потрібно чи ні обчислювати дане значення зв'язку. Розпаковка такого групового кодування є фактично миттєвою операцією і тому дану область пам'яті можна використовувати для створення потоку  $\langle t_{k_1}, q_{m_1}, 0; t_{k_2}, q_{m_2}, \theta_{n_2} \rangle$  ідентифікаторів зв'язку. Рівняння для визначення ідентифікаторів восьми структурних зв'язків і відповідних вагових коефіцієнтів інтерполяційної функції спрощуються шляхом прийняття  $\theta_{n_1} = 0$ , внаслідок чого значення вагових коефіцієнтів для  $\langle N_\theta / n_2 - 1 \rangle$  дублікатів зв'язків однакової довжини будуть рівними.

Лишається проблема адресації значення зв'язку в ОЗП. Схема прямого адресування вимагала б  $O(N^4)$  розміру ОЗП з переважно непотрібними адресами, оскільки в кожному кроці обчислення є лише  $O(N^3)$  "необрізаних" зв'язків. Отже, якщо отримати інформацію про зв'язки, які будуть обрізатися заздалегідь, то розмір ОЗП  $O(N^3)$  є достатнім.

*1. Feldkamp, L. A., L. Davis, and J. Kress. Practical Cone-beam Algorithm. J.Opt.Soc.Am.—1984, 1.—P. 612-619. Danielsson, P.-E. and M. Ingerhed. Backprojection in  $O(N^2 \log N)$  time. In IEEE Medical Imaging, Albuquerque, New Mexico, USA,—1997, № 9.—P. 13-15.*