

# Методи стискання даних за допомогою перетворення сплесків

Р.В.Лотоцький

Інститут кібернетики ім. Глушкова НАН України, т.(044)211-56-54, roman\_l@i.com.ua,

Розглядається застосування перетворення сплесків для однієї з найважливіших проблем обробки інформації – стискання графічних зображень. Наводяться основні факти теорії сплесків в об'ємі, достатньому для конструктивного опису побудованого алгоритму. Результати роботи розробленого програмного комплексу проаналізовані на прикладах стискання зображень реальних відбитків пальців. Обговорюються основні підходи до удосконалення розглянутого методу стискання.

## 1. Вступ

Методи стискання даних можна розділити на два класи: стискання із втратами і без втрат. Алгоритми з другого класу зазвичай застосовуються для перетворення текстової інформації, тому що в цьому випадку потрібне точне відновлення даних. На відміну від текстової інформації, для графічних напівтонових зображень допускаються втрати при стисканні. У цьому випадку відновлене зображення не буде точною копією оригіналу, але часто така умова і не потрібна, оскільки людське око не сприймає невеликі зміни окремих пікселів. Застосування алгоритмів із втратами дозволяє істотно підвищити коефіцієнт стискання в порівнянні з алгоритмами без втрат. У даній роботі розглядається метод стискання з втратами графічних зображень, основним етапом якого є перетворення сплесків (Wavelet). Результати роботи реалізованого алгоритму показано на прикладах обробки реальних фотографій відбитків пальців.

## 2. Основні відомості з теорії перетворення сплесків

**2.1. Неперервне перетворення сплесків.** Неперервне перетворення сплесків визначається таким чином:

$$WF(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t) \overline{\psi\left(\frac{t-b}{a}\right)} dt = \int_{-\infty}^{\infty} f(t) \overline{\psi_{a,b}} dt;$$

Тут, функція  $\psi$  задовільняє наступним умовам:

$$\psi \in L_2(\mathbb{R}), \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\xi)|^2}{|\xi|} d\xi < \infty. \quad (1)$$

Тут і далі  $\hat{\psi}$  - перетворення Фур'є функції  $\psi$ . Базисна система сплесків  $\{\psi_{a,b}\}$  складається з усіх комбінацій зсувів і стиснень функції  $\psi$ , що у термінах теорії сплесків називається "утворюючим сплеском" (mother wavelet).

$$\psi_{a,b} = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right), \quad (2)$$

Зворотнє перетворення:

$$f(t) = C_{\psi}^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} WF(a,b) \psi_{a,b}(t) \frac{dad b}{a^2}, \quad (3)$$

$$\text{де } C_{\psi} = 2\pi \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\xi)|^2}{|\xi|} d\xi.$$

**2.2. Кратно-масштабний аналіз.** Кратно-масштабний аналіз є математичним апаратом для конструктивного опису і побудови різноманітних базисів функцій сплесків. Вперше це поняття було введено Маллатом [1]. Основні положення теорії кратно-масштабного аналізу можна знайти також у [2,3].

Кратно-масштабний аналіз на  $L_2(\mathbb{R})$  визначається як послідовність вкладених підпросторів  $V_j$  на  $L_2(\mathbb{R})$  з такими властивостями:

- 1)  $V_j \subset V_{j+1}, j \in \mathbb{Z}$ ;
- 2)  $v(x) \in V_j \Leftrightarrow v(2x) \in V_{j+1}$ ;
- 3)  $v(x) \in V_0 \Leftrightarrow v(x+1) \in V_0$ ;
- 4)  $\overline{\bigcup_{j=-\infty}^{+\infty} V_j} = L_2(\mathbb{R}), \quad \bigcap_{j=-\infty}^{+\infty} V_j = \emptyset,$  де  $\overline{\bigcup_{j=-\infty}^{+\infty} V_j}$  -- замикання підпростору  $\bigcup_{j=-\infty}^{+\infty} V_j$ .
- 5) Існує функція  $\phi$  така, що система функцій  $\{\phi(x-k) | k \in \mathbb{Z}\}$  утворює ортонормований базис простору  $V_0$ . Функція  $\phi$  називається масштабуючою функцією.



Із властивостей 2) і 5) випливає, що множина функцій  $\varphi_k^j(x) = \sqrt{2^j} \varphi(2^j x - k)$ ,  $j \in \mathbf{Z}$ ,  $k \in \mathbf{Z}$ ,

утворює ортонормований базис підпростору  $V_j$ .

Оскільки  $V_0 \subset V_1$ , то будь-яку функцію з  $V_0$  можна записати як лінійну комбінацію базисних функцій  $\sqrt{2}\varphi(2x - k)$ . Отже, маємо

$$\varphi(x) = \sum_k h(k) \sqrt{2} \varphi(2x - k),$$

де коефіцієнти  $h(k)$  внаслідок ортогональності  $\varphi(x)$  визначаються таким чином:

$$h(k) = (\varphi(x), \sqrt{2}\varphi(2x - k)). \quad (4)$$

Нехай  $W_j$  -- ортогональне доповнення підпростору  $V_j$  до  $V_{j+1}$ ,  $V_{j+1} = V_j \oplus W_j$ . З останньої рівності і властивості 1) можна записати

$$\forall j, j_1, j < j_1: V_{j_1} = V_j \oplus \bigoplus_{k=0}^{j_1-j-1} W_{j_1-k}.$$

Таким чином, враховуючи, що  $\bigcup_{j=-\infty}^{+\infty} V_j = L_2(\mathbf{R})$ ,

маємо  $\bigoplus_{j \in \mathbf{Z}} W_j = L_2(\mathbf{R})$ .

Головною властивістю кратно-масштабного аналізу є той факт, що якщо послідовність вкладених підпросторів  $V_j$  задовольняє умовам 1)-5), то існує можливість побудови такого ортонормованого базису  $\{\psi_k^j; j, k \in \mathbf{Z}\}$  простору  $L_2(\mathbf{R})$ ,  $\psi_k^j \in W_j$ , що  $\forall f \in L_2(\mathbf{R})$ :

$$P_{j+1}(f) = P_j(f) + \sum_k (f, \psi_k^j) \psi_k^j,$$

де  $P_j(f)$  -- ортогональна проекція функції  $f$  на  $V_j$ .

Базис  $\{\psi_k^j; j, k \in \mathbf{Z}\}$  називається базисом сплесків. Так як  $W_j \subset V_{j+1}$ , то має місце таке подання функції  $\psi^j(x)$ :

$$\psi^j(x) = \sum_k g(k) \varphi_k^{j+1}(x). \quad \text{Доведено [2], що}$$

умова ортонормованості базису  $\{\psi_k^j; j, k \in \mathbf{Z}\}$  виконується при наступному виборі коефіцієнтів  $g(k)$ :  $g(k) = (-1)^k h(-k+1)$ . Таким чином, формули залежності  $\psi^j(x)$  від  $\{\varphi_k^{j+1}(x)\}$  і  $\varphi^j(x)$  від  $\{\varphi_k^{j+1}(x)\}$  мають такий вид:

$$\varphi^j(x) = \sum_k h(k) \varphi_k^{j+1}(x), \quad (5)$$

$$\psi^j(x) = \sum_k (-1)^k h(-k+1) \varphi_k^{j+1}(x). \quad (6)$$

### 2.3. Швидке перетворення сплесків.

Кратно-масштабний аналіз дозволяє побудувати швидкий алгоритм перетворення сплесків. Так як  $V_{j+1} = V_j \oplus W_j$ , то функцію  $v_{j+1} \in V_{j+1}$  можна записати

$$v_{j+1} = \sum_k \lambda_k^{j+1} \varphi_k^{j+1}(x) = v_j(x) + w_j(x) = \sum_l \lambda_l^j \varphi_l^j(x) + \sum_l \gamma_l^j \psi_l^j(x), \quad (7)$$

де  $v_j \in V_j$  і  $w_j \in W_j$

Встановимо зв'язок між коефіцієнтами  $\lambda_k^{j+1}$ ,  $\lambda_l^j$  і  $\lambda_k^{j+1}$ ,  $\gamma_l^j$ . З формули (5) випливає:

$$\varphi_k^j(x) = 2^{2^j} \varphi(2^j x - k) = 2^{2^j} \sum_l h(l) \varphi(2^{j+1} x - 2k - l) = \sum_l h(l) \varphi_{2k+l}^{j+1}(x) = \sum_l h(l - 2k) \varphi_l^{j+1}(x) \quad (8)$$

Аналогічно:

$$\psi_k^j(x) = \sum_l g(l - 2k) \varphi_l^{j+1}(x) \quad (9)$$

Звідси маємо

$$\lambda_k^j = \sum_l h(l - 2k) \lambda_l^{j+1}, \quad (10)$$

$$\gamma_k^j = \sum_l g(l - 2k) \lambda_l^{j+1}. \quad (11)$$

Обернене перетворення отримуємо аналогічно. З (7)-(9) випливає, що:

$$\lambda_k^{j+1} = \sum_l \lambda_l^j (\varphi_l^j, \varphi_k^{j+1}) + \sum_l \gamma_l^j (\psi_l^j, \varphi_k^{j+1}) = \sum_l h(l - 2k) \lambda_l^j + \sum_l g(l - 2k) \gamma_l^j. \quad (12)$$

Таким чином, якщо відомі коефіцієнти розкладу даної функції по базисних функціях у деякому підпросторі  $V_j$ , то, скориставшись (10), (11), можна знайти розклад цієї функції також в усіх підпросторах  $W_J$ ,  $J < j$ , а скориставшись (12), можна відновити вихідний розклад.

Формули (10), (11) можна записати в більш компактній і зручній для практичного застосування формі, використовуючи так зване операторне подання фільтрів.

Отже, нехай вектор  $a = (a_l)$ ,  $l \in [0, N]$  складається з коефіцієнтів розкладу деякої функції



по базису підпростору  $V_j$ . Визначимо оператори  $H$  і  $G$ :

$$(Ha)_k = \sum_{l=0}^N h(l-2k)a_l, \quad (13)$$

$$(Ga)_k = \sum_{l=0}^N g(l-2k)a_l, \quad \forall k \in \left[0, \left[\frac{N}{2}\right]\right], \quad (14)$$

[•] - ціла частина числа. Ці оператори відповідають перетворенням (10), (11) і виконують один крок у перетворенні сплесків. Відповідні оператори для оберненого перетворення (12) мають такий вигляд:

$$(H^*a)_l = \sum_{k=0}^{\left[\frac{N}{2}\right]} h(l-2k)a_k, \quad (15)$$

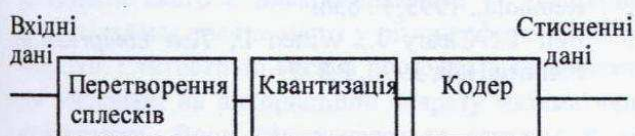
$$(G^*a)_l = \sum_{k=0}^{\left[\frac{N}{2}\right]} g(l-2k)a_k, \quad \forall l \in [0, N]. \quad (16)$$

### 3. Алгоритм стискання зображень методом сплесків.

На сьогодні найбільш відомим алгоритмом стискання графічних файлів з втратами є алгоритм JPEG [4], в якому використовується косинусоїдальне перетворення Фур'є (КПФ). Цей алгоритм дозволяє стискати графічні зображення в 25-30 разів без суттєвого погіршення якості. Проте якщо зміна інтенсивності пікселів зображення не є плавною (як, наприклад, у відбитках пальців), то коефіцієнт стиску різко падає. Пропонований алгоритм має структуру, подібну алгоритму JPEG, але використовує замість КПФ перетворення сплесків.

У наступних викладах будемо вважати, що зображення є двомірною дискретною функцією,  $f(x, y) \in [0, 255]$  значеннями якої є 256 відтінків сірого (0 -- чорний, 255 -- білий)

Процес стискання складається з таких трьох кроків: перетворення сплесків, квантизація, кодування (Рис. 3).



Вхідне зображення розбивається на блоки  $8 \times 8$  пікселів. Наведені вище операції застосовуються окремо до кожного блоку.

На першому кроці виконується перетворення сплесків. Вище було розглянуто одномірне перетворення сплесків для векторів. У двомірному випадку для перетворення матриці даних застосовуємо одномірне перетворення для кожного рядка, і для отриманої в результаті матриці знову

застосовуємо одномірне перетворення для кожного стовпчика.

Оскільки різні коефіцієнти перетворення по-різному впливають на відновлення вихідного зображення, то, для підвищення ступеня стискання, отримані коефіцієнти  $w_i$  квантизуються, тобто діляться на коефіцієнти квантизації:  $k_i \cdot r_i = w_i / k_i$ . Потім, округлені до найближчого цілого, вони надходять на вхід ентропічного кодера.

При підборі правильної таблиці квантизації  $Q = (k_{i,j})$  необхідно враховувати вплив кожного коефіцієнта на відновлення зображення. У даній роботі пропонується такий алгоритм для знаходження матриці квантизації. Нехай  $E$  - деякий рівень похибки, тобто  $E = \|f(x, y) - \tilde{f}(x, y)\|$ ,

де  $\tilde{f}(x, y)$  -- відновлене зображення. Для кожного коефіцієнта перетворення знаходимо таке максимальне значення  $k_j$ , що

$$\|f(x, y) - \tilde{f}(x, y)\| \leq E. \quad \text{Нехай } k_{\min} = \min_j k_j$$

$$q_j = \frac{k_j}{k_{\min}}. \quad \text{Тоді шукана матриця має такий$$

вигляд:  $Q = (q_j k_{\min})_j$ . За допомогою коефіцієнта  $k_{\min}$  будемо регулювати коефіцієнт стискання - при більшому  $k_{\min}$  стискання буде більшим. Так як не вигідно знаходити матрицю квантизації для кожного зображення окремо в зв'язку із значним часом роботи алгоритму, то достатньо побудувати одну матрицю для деякого класу зображень. Такими класами можуть бути, наприклад, відбитки пальців, фотографії обличчя людини.

На заключному етапі алгоритму стискання квантизовані коефіцієнти кодуються за допомогою одного з алгоритмів ентропійного кодування. У даній роботі використовується динамічний алгоритм кодування Хаффмана. Цей алгоритм добре відомий у теорії кодування ([4, 5]).

**3.1. Особливості розробленої програми.** Розроблена програма дозволяє стискати зображення з градацією сірого. У якості базису сплесків використовується система функцій Добеші [2].

Таблиця квантизації, побудована для класу зображень відбитків пальців, має такий вид:

5k, 6k, 7k, 7k, 8k, 10k, 10k, 8k,  
 6k, 7k, 7k, 7k, 10k, 10k, 10k, 9k,  
 7k, 7k, 8k, 7k, 11k, 11k, 10k, 10k,  
 7k, 7k, 8k, 7k, 10k, 10k, 10k, 10k,  
 8k, 9k, 10k, 10k, 11k, 11k, 11k, 10k,  
 9k, 10k, 10k, 10k, 11k, 11k, 11k, 10k,  
 9k, 9k, 10k, 10k, 11k, 11k, 11k, 11k,



**3.2. Отримані результати.** Результати роботи програми будемо розглядати на прикладі фотографії відбитка пальця. У наступній таблиці наведені деякі значення "відстані" між оригінальним зображенням і зображенням, відновленим після стиснення. "Відстань" обчислюється по формулі середньоквадратичного значення різниць між інтенсивностями відповідних пікселів двох зображень:

$$D = \sqrt{\frac{\sum_{i=0}^{width-1} \sum_{j=0}^{height-1} (f(i,j) - \tilde{f}(i,j))^2}{width * height}}$$

де  $f(i,j)$  і  $\tilde{f}(i,j)$  -- функції інтенсивності пікселів відповідно оригінального і відновленого зображень,  $width \times height$  -- розміри зображення (Таблиця).

Таблиця

Коефіцієнт стиснення	Відстань
3	2.93
5	4.15
10	6.84
15	8.10
20	9.51
25	11.53
30	12.59
35	13.52
40	14.19
45	15.15

Нижче приведено результати роботи програми на прикладі фотографії відбитку пальця

Оригінальне зображення: Коефіцієнт стиснення 10:



Коефіцієнт стиснення 20:

Коефіцієнт стиснення 40:



## ВИСНОВКИ

Основним вдосконаленням аналізованого методу є, звичайно, використання кращих систем сплесків, тобто систем, що краще відбивають специфіку вхідної функції, в даному випадку зображення. На сьогодні найкращими базисами сплесків, що застосовуються для стиснення зображень, є сплески Добеші [2]. Кращі результати можна одержати, використовуючи для перетворення суперпозицію різних базисних функцій сплесків. Також важливим чинником, що дуже сильно впливає на коефіцієнт стиску, є правильний вибір таблиці квантизації. У даній роботі при знаходженні коефіцієнтів квантизації враховується вплив кожного коефіцієнта окремо, хоча в загальному випадку втрати при квантизації двох коефіцієнтів не будуть рівні сумі втрат для кожного коефіцієнта окремо. Найкращим методом знаходження цих коефіцієнтів буде використання класичних методів оптимізації, але час роботи таких алгоритмів може бути дуже великим.

## Література

1. Mallat S. *Multiresolution approximation and wavelets*. // Trans. AMS – 1989 --315, -- P. 69-88.
2. Daubechies I., *Ten lectures on wavelets*. // SIAM Philadelphia, 1992. -- P.377
3. Новиков И.Я., Стечкин С.Б. *Основы теории всплесков* // Успехи мат. наук. – 1998 -- 53. № 6 (324) -- С. 53-128
4. Pennebaker W.B., Mitchell J.L., *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold., 1993, P. 656.
5. Bell T., Cleary J., Witten I., *Text compression*. Prentice Hall, 1990, P.337