

УДК 621.397

АНАЛІЗ МЕТОДІВ СТИСНЕННЯ ЦИФРОВИХ ВІДЕОПОСЛІДОВНОСТЕЙ У ТЕЛЕВІЗІЙНИХ СИСТЕМАХ

ПАТЛАЄНКО М.О., СОЛОДКА В.І., АРДЕЛЯН М.І.

Одеська національна академія зв'язку ім. О. С. Попова

DIGITAL VIDEO SEQUENCES COMPRESSION METHODS IN TELEVISION ANALYSIS

PATLAENKO N.A., SOLODKA V.I., ARDELYAN M.I.

Odessa national academy of telecommunications n.a. O.S. Popov

Анотація. Показані алгоритми стиснення цифрових відео сигналів, які є актуальними в телевізійних системах. На базі основних алгоритмів дуже швидко збільшуються кількість нових алгоритмів з врахуванням швидкості стиснення і їх ефективності. Проведений аналіз відповідних алгоритмів стиснення за інформаційних параметрах.

Abstract. The algorithms of compression of digital videos of signals which are actual in the television systems are rotined. On the base of basic algorithms very quickly increased amount of new algorithms taking into account speed of compression and their efficiency. The analysis of the proper algorithms of compression is conducted on informative parameters.

На сьогодні алгоритми стиснення цифрових відео послідовностей набули значного поширення, у зв'язку з чим кількість алгоритмів стрімко зростає і все частіше і частіше виникає питання вибору того чи іншого алгоритму стиснення та їх ефективності.

Алгоритм *JPEG* можна пристосувати для роботи з рухомими зображеннями об'єднавши в відео потік, таким є алгоритм *MJPEG*. Даний алгоритм заснований на стисненні кожного кадру окремо з використанням стиснення за алгоритмом *JPEG* [3]. Після виконання операції стиснення послідовність стиснених зображень збирається у відеопотік. Нижче наводиться опис алгоритму стиснення для кожного кадру.

Відомо, що людське око має більшу чутливість до зміни яскравості, ніж до зміни кольору. Тому при кодуванні зображень з використанням *JPEG*-подібних алгоритмів використовується представлення картинки не в *RGB*-форматі, а у форматі *YCbCr* [3]. При цьому компоненти хроматичного червоного і хроматичного синього усереднюються, що дозволяє зменшити обсяг даних в 1,5 рази.

При кодуванні зображення розбивається на блоки розміром 8x8 елементів, і подальші операції виконуються над кожним з цих блоків. Кодування проводиться трьома фазами:

- швидке перетворення Фур'є,
- квантування,
- стиснення.

Перетворення Фур'є здійснює перетворення сигналів з просторової області в частотну. Для блоку розміром 8x8 елементів воно задається наступним виразом [1]:

$$I(u, v) = \frac{a(u)a(v)}{4} \times \sum_{x=0}^7 \sum_{y=0}^7 s(x, y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right), \quad (1)$$

де $a(u), a(v) = \frac{1}{\sqrt{2}}$, при $u, v = 0$; $a(u), a(v) = 1$, при $u, v \neq 0$.

$i(x, y)$ – значення елементів (просторова область),

$I(u, v)$ – амплітуда спектральних складових (частотна область).

Квантування здійснюється шляхом ділення амплітуд спектральних складових, отриманих після перетворення Фур'є, на заздалегідь задані числа, взяті з таблиці квантування. Вибір таблиці квантування впливає на число нулів, отриманих після квантування, а це, у свою чергу, визначає коефіцієнт стиснення.

Стиснення здійснюється у три стадії. Спочатку проводиться зигзаг сканування. Ця операція з упорядкування коефіцієнтів Фур'є необхідна для того, щоб отримати можливо більш довгі послідовності нулів, оскільки такі послідовності краще кодуються. Потім за допомогою алгоритму *RLE* відбувається стиснення, при якому кожен коефіцієнт кодується двома числами: перше число показує, скільки нульових елементів належить даному коефіцієнту, а друге – значення цього коефіцієнта. Потім проводиться стиснення по Хаффману [5] або арифметичне кодування [1].

Хвильовий алгоритм *Wavelet*. При *Wavelet* перетворенні проводиться розкладання сигналу на базисні функції. На відміну від перетворення Фур'є в якості базових функцій використовуються не косинуси, а вейвлети. Вейвлети це добре локалізовані солітоноподібні функції. На рис. 1 подано чотири сімейства вейвлет функцій [7].

Для стиснення зображень використовуються два вейвлет перетворення з різними базисними функціями. При цьому базисні функції вибирають таким чином, щоб одне перетворення здійснювало фільтрацію низьких частот, а друге – високих. Низькочастотну складову сигналу після фільтрації будемо позначати буквою – *L*, а високочастотну – *H*. Оскільки оброблювана картинка двовимірною, проводиться двовимірне перетворення (рис. 2). Чотири найбільш відомих сімейства вейвлет [7] перетворення, при цьому вейвлет функцій [7] спочатку здійснюється фільтрація по рядках, а потім по стовпцях, або навпаки. В результаті двовимірного вейвлет перетворення ми отримуємо чотири компоненти: *LL*, *LH*, *HL* і *HH*. Найбільша кількість інформації містить *LL* – компонента. У *LH* – і *HH* – компонентах велика кількість нулів, а елементи *HH* – компоненти практично всі дорівнюють нулю (рис. 1). Для збільшення коефіцієнта стиснення низькочастотна *LL* – компонента знову піддається двовимірному вейвлет перетворенню і т. д. [9]

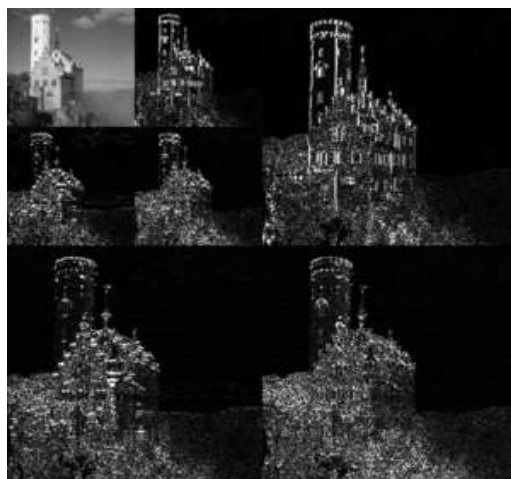


Рисунок 1 – Компоненти, отримані в результаті двовимірного вейвлет перетворення *Y* складової зображення

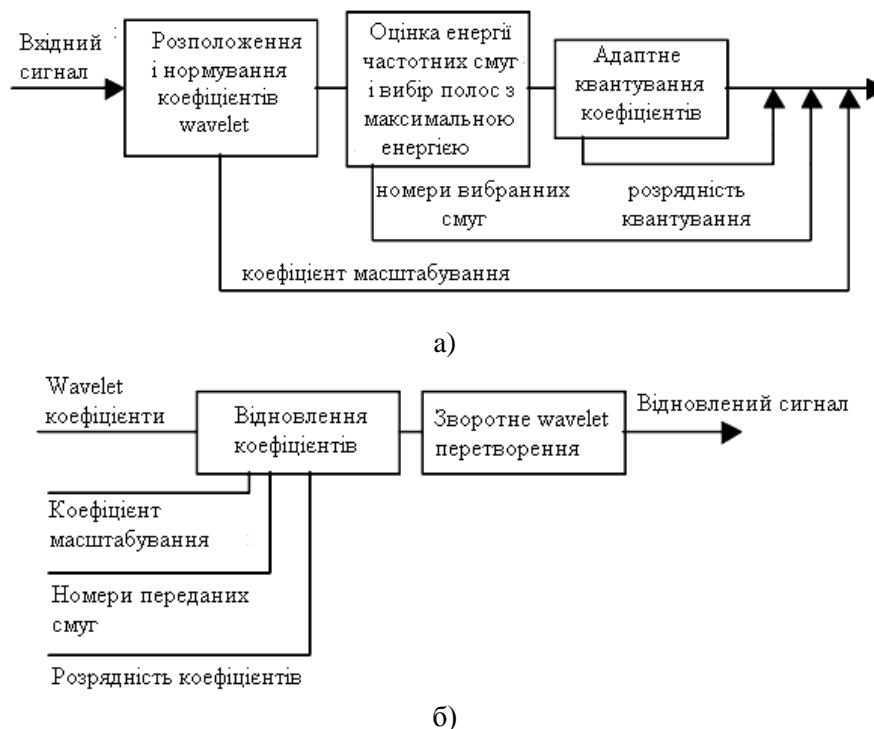


Рисунок 2 – Структурна схема кодера: а) та декодера: б)

Розглянемо операцію стиснення, кодер та декодер наведену на рис. 2. Зображення подається у вигляді компонент $YCrCb$, кожна з компонент обробляється окремо.

Кодування кожної компоненти відбувається у 4 стадії:

- дискретне двовимірне вейвлет перетворення,
- квантування,
- для низькочастотних компонент – кодування з використанням дискретної імпульсно-кової модуляції. Для високочастотних – сканування нуля дерева,
- арифметичне кодування,

При кодуванні відеопослідовності, проводиться стиснення кожного кадру окремо і тим самим не усувається тимчасова надмірність інформації, що є одним з основних недоліків Wavelet стиснення.

Проаналізуємо алгоритм *MPEG4*. У стандарті *MPEG4* [2] так само, як і в стандарті *JPEG*, відеодані подаються у форматі $YCrCb$ [3]. Кодування відео потоків виробляється з використанням об'єктних відео площин (*VOP*-площин) [2]. При стисненні відео потоку кожне зображення можна подати у вигляді суперпозиції *VOP*-площин, які в подальшому кодуються незалежно один від одного. *VOP*-площина містить певну частину кадру, що зберігає деяку візуально помітну область вихідного зображення. Оскільки на даний момент не існує ефективних методів розділення зображень на окремі об'єкти, найчастіше використовується всього одна відео-площина, цілком містить вихідне зображення. При стисненні відео-площина розділяється на послідовність кадрів. У стандарті *MPEG4* використовуються три типи кадрів: "I" і "P" і "B":

"I" кадри кодуються повністю, без використання інформації про інші кадри;

"P" кадри кодуються з використанням інформації про попередні "I" і "P" кадри;

"B" кадри використовують інформацію як про попередні, так і про подальші "I" і "P" кадри.

Тепер розглянемо безпосередньо кодування кадрів. Кадри розбиваються на макроблоки. Для *Y*-компоненти макроблоки мають розмір 16 x 16 елементів, а для C_r і C_b компонент розмір

макроблоків дорівнює 8×8 елементів. Макроблоки бувають двох типів: intra-макроблоки і inter-макроблоки. Макроблоки різних типів кодуються по-різному. “I” кадр складається тільки з intra-макроблоків, а “P” кадр містить і inter і intra-макроблоки.

Макроблоки Y-компоненти розбиваються на під блоки розміром 8×8 , і здійснюється кодування.

Розглянемо кодування intra-макроблоків для C_r і C_b компонент і підблоків Y-компоненти. Спочатку виконується перетворення Фур’є точно таке, як описано вище у стандарті JPEG. Після цього проводиться квантування коефіцієнтів Фур’є. Старший з коефіцієнтів Фур’є відображає середнє значення яскравості в даному блоці, тому його називають DC-коефіцієнтом. Оскільки найчастіше яскравість сусідніх блоків однакова, кодування DC-коефіцієнта цілком є надлишковим, достатньо закодувати різницю між поточним і одним з попередніх блоків. Подібне передбачення виконується і для старших AC-коефіцієнтів. Якщо при передбаченні кількість інформації не зменшується, то блок кодується без урахування передбачення. Якщо AC-передбачення здійснюється, то залежно від того, по відношенню до якого блоку воно виконується, вибирається один з трьох можливих методів зигзаг-сканування. Після цього виконується RLE-стиснення і стиснення за Хаффманом так само, як і в стандарті JPEG.

Для inter-макроблоків спочатку здійснюється операція компенсації руху [2], і тільки після цього вони кодуються. Для кожного макроблоку використовується один вектор швидкості, при цьому передбачення здійснюється з точністю до $0,125-0,5$ пікселя. Для знаходження вектора швидкості виробляється пошук найбільш схожого місця на попередніх “I” і “P” кадрах. Пошук векторів руху проводиться тільки для Y-компоненти, для C_r і C_b компонент вектор руху виходить шляхом розподілу навпіл вектора для компоненти Y. Після знаходження вектора руху з кодованого макроблок віднімається найбільш схожий блок, і надалі кодуються різниця між блоками, позиція кадру, по відношенню до якого закодований даний макроблок, і вектор руху. Якщо кодування різниці займає більше біт, ніж кодування вихідного блоку, то блок маркується як intra-блок і після цього проводиться його кодування за відповідним алгоритмом. Надалі inter-блоки кодуються точно так само як і intra-блоки, за винятком того, що для них не виробляються AC/DC-передбачення.

Проте, для оцінки ефективності алгоритмів був обраний відеофрагмент довжиною 55 с, що має дозвіл 1920×1080 , частоту кадрів 25 Гц і містить як сцени з інтенсивним рухом, так і статичні сцени. Для кожного алгоритму вироблялося стиск відеофрагменту з різним коефіцієнтом стиснення.

Після цього кожен з закодованих зразків порівнювався з вихідним відео фрагментом для обчислення сумарного відношення сигнал/шум, що наведено у формулі (2). Відношення сигнал/шум не дає повної картини того, який з алгоритмів краще, проте цілком підходить для якісного аналізу, і у зв’язку з простотою реалізації був обраний саме цей метод.

$$PSNR = -10 \times \log \left(\frac{\sum_{x=1}^{1920} \sum_{y=1}^{1080} |P'(x, y) - P(x, y)|}{\sum_{x=1}^{1920} \sum_{y=1}^{1080} |P(x, y)|} \right), \quad (2)$$

де $P(x, y)$ – значення яскравості (колірної) компоненти пікселя вихідного кадру, а $P'(x, y)$ – кадру, отриманого після декомпресії стисненого кадру.

Для стиснених відеофрагментів обчислювався коефіцієнт стиснення. Далі на графіку по осі абсцис відкладався коефіцієнт стиснення, а по осі ординат співвідношення сигнал/шум (рис. 3). Оскільки ефективність MPEG4 алгоритму сильно залежить від реалізації пошуку векторів швидкостей для компенсації руху, для більшої наочності були досліджені дві реалізації даного алгоритму, а саме реалізації створені групами DIVX і XVID [4].

На рис. 3 подано результати порівняльного аналізу різних алгоритмів стиснення цифрових відеопотоків. Порівнювалися три типи алгоритмів: MJPEG, MPEG4 і wavelet. Чим вище розташування лінії графіка, тим вища ефективність кодека.

Розглянемо взаємне розташування графіків для MJPEG і MPEG4. Не важко помітити, наскільки значний внесок у стиснення дає компенсація руху. При однаковому співвідношен-

ні сигнал/шум *MPEG4* забезпечує в 3 рази більший коефіцієнт стиснення в порівнянні з *MJPEG*. Проте на ефективність роботи кодека впливає не тільки алгоритм, за яким проводиться стиснення, але і те, наскільки добре реалізований цей алгоритм. Це впливає з взаємного розташування графіків для *MPEG4.3vх* *MPEG4.divx*. Зрозуміло, чому алгоритм, реалізований групою *DIVX*, користується більшою популярністю.

Найбільший інтерес становлять графік для *Wavelet* стиснення. Хоча в алгоритмі *Wavelet* стиснення жодним чином не усувається тимчасова надмірність, ефективність даного алгоритму не набагато нижче ефективності алгоритму *MPEG4*, а при низьких коефіцієнтах стиснення навіть перевищує її.

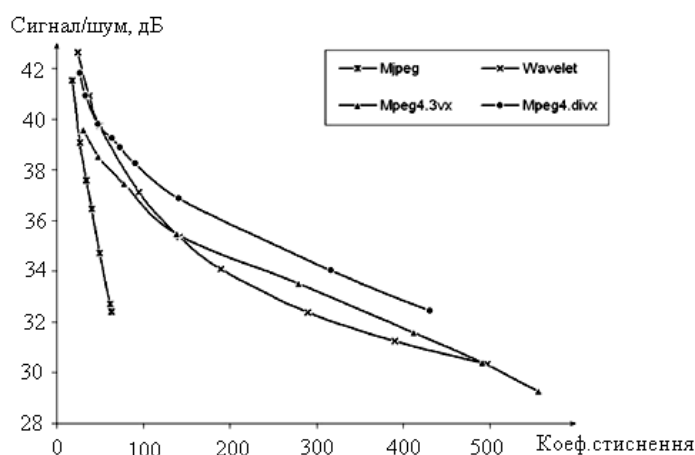


Рисунок 3 – PSNR при різних коефіцієнтах стиснення

ВИСНОВКИ

З проведеного аналізу видно, що *Wavelet* стиснення має більший потенціал, ніж *MPEG*-подібні стандарти. Головне завдання, яке треба вирішити для створення більш ефективних *Wavelet* подібних методів стиснення полягає в усуненні тимчасової надмірності.

Одним з найбільш перспективних *Wavelet* подібних методів стиснення відео потоків є *Wavelet 3D*, в якому тимчасова вісь розглядається як третій вимір за цим виміром також проводиться *Wavelet* перетворення. До основних недоліків цього методу слід віднести великі ресурсні витрати, а також той факт, що тимчасова вісь має більш грубий крок дискретизації, ніж просторові осі.

ЛІТЕРАТУРА

- 1 Witten, I. Arithmetic coding for data compression / Witten, I., Radford, M., Cleary, J. // Communications of the ACM. – 1987. – Vol. 30. – No. 6. – P. 40–520
- 2 Jacklin, M. MPEG-4 - The Media Standard / M. Jacklin // MPEG-4 Industry Forum. – 2002
- 3 Wallace, G. The JPEG algorithm for image compression standard / G. Wallace // Communications of the ACM. – 1991. – Vol. 34. – No. 4. – P. 31–44
- 4 Niedemayer M. DIVX3, MS-MPEG4v1-v3, WMV7-8 / M. Niedemayer // GNU Free Documentation. – 2003
- 5 Huffman, D. A method for the construction of minimum redundancy codes / D. Huffman // In proceedings IRE. – 1962. – Vol. 40. – pp. 1098–101
- 6 Li S. Coding of Moving Pictures and Audio: MPEG-4 Video Verification Model / S. Li // Pisa. – 2001. – v 18, N3908 (ISO/IEC JTC1/SC29/WG11)
- 7 Astaf'eva, N. Wavelet analysis: basic theory and some applications / N. Astaf'eva // Space Research Institute, Russian Academy of Science. – 1996
- 8 Donoho D.L. Ideal spatial adaptation via wavelet shrinkage / D.L. Donoho // Biometrika. – 1994. – 81. – p.425–455
- 9 E.P. Simoncelli, E.H. Adelson Noise removal via Bayesian wavelet coring in Proc / E.P. Simoncelli, E.H. Adelson // 3rd Int. Conf. Image Processing. – Lausanne, Switzerland. – Sep.1996. – vol. I. – Pp. 379–382.