

## ЗАСТОСУВАННЯ СЛОВНИКОВИХ МЕТОДІВ СТИСНЕННЯ ДЛЯ КОМПРЕСІЇ АУДІО ДАНИХ З РЕГУЛЬОВАНИМИ ВТРАТАМИ

*Робота присвячена розробці метода стиснення аудіо даних з регульованими втратами за допомогою різницевих псевдо-аналогів з використанням словників.*

*Ключові слова: словники, аудіосемпли, механізм стиснення, алгоритм стиснення, стиснення з втратами.*

На сьогоднішній день для ефективного стиснення звукової інформації використовують методи стиснення з втратами, серед яких можна виділити психоакустичні методи (MP3, AAC), а також методи стиснення на основі фракталів, вейвлетів та стохастичних диференціальних рівнянь [1].

Натомість, словникові методи безвтратного стиснення, такі, як метод Лемпеля-Зіва [2] та його модифікації [3–6], для звукової інформації дають лише незначну компресію, а іноді компресія взагалі відсутня. Спроби розробки словникового метода, який би вирішував цю проблему, були започатковані у роботі [7], де пропонується спосіб проєкційного стиснення інформації на основі словника. Однак, після проведення машинних експериментів зі стиснення звукової інформації було з'ясовано, що даний спосіб дає майже ті самі ступені компресії, що й метод Лемпеля-Зіва, будучи при цьому значно більш ресурсоемним.

Для досягнення більших ступенів компресії в таких задачах можна застосовувати словникове стиснення з втратами. Ціль даної роботи – продемонструвати спосіб, який би дозволяв це робити.

Спосіб стиснення, що пропонується, будемо називати способом різницевих псевдо-аналогів. Під час стиснення на вхід подається звукова інформація у вигляді потоку амплітудних значень звукового сигналу (аудіосемплів). Кодер містить словник у вигляді буферу, який вміщує  $L$  аудіосемплів. На початку стиснення словник заповнюється аудіосемплями з вхідного потоку. Кодер також містить спеціальний буфер вводу (кеш аудіосемплів), який вміщує  $l$  аудіосемплів ( $l \ll L$ ) і постійно поповнюється з вхідного потоку. Після поповнення кеша аудіосемплів відбувається пошук різницевих псевдо-аналогів вмісту кеша у словнику.

Різницеvim псевдо-аналогом (РПА) довжини  $p$  будемо називати блок з  $p \leq l$  аудіосемплів зі словника, який задовольняє наступній умові:

$$|q_{i+j} - s_i - r| \leq \delta, \quad i = \overline{1, p}, \quad (1)$$

де  $q_{i+j}$  – аудіосемпли зі словника;  $j$  – позиція РПА у словнику;  $s_i$  – відповідні аудіосемпли з кеша;  $r$  – середня різниця РПА;  $\delta$  – задана допустима похибка значення аудіосемплів.

Середня різниця РПА є середнім арифметичним різниць між значеннями відповідних аудіосемплів з РПА та кеша:

$$r = \frac{1}{p} \sum_{i=1}^p q_{i+j} - s_i.$$

Таким чином, значення аудіосемплів РПА відрізняються від відповідних значень аудіосемплів кеша на різницю  $r$  в межах заданої допустимої похибки  $\delta$ . Якщо  $r = 0$ , то такий РПА будемо називати простим псевдо-аналогом (ППА).

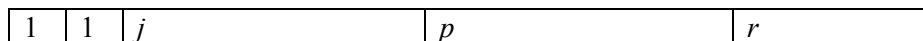
Механізм стиснення полягає у тому, що при знаходженні РПА замість відповідної частини кеша зберігається лише позиція РПА  $j$  у словнику, довжина РПА  $p$  та різниця  $r$ . Тобто, частина кеша семплів замінюється відповідним поданням через вміст словника. При цьому, якщо  $\delta > 0$ , то неважко переконатись, що відбувається стиснення з втратами.

Під час стиснення можливі 3 варіанти ситуацій:

- знайдений РПА ( $r \neq 0$ ) – необхідно зберегти  $j$ ,  $p$  та  $r$ ;
- знайдений ППА ( $r = 0$ ) – необхідно зберегти лише  $j$  та  $p$ ;
- не знайдено ані РПА, ані ППА – необхідно вилучити перший аудіосемпл з кешу і зберегти його в явному вигляді.

Таким чином, результати стиснення будуть одного з трьох типів: РПА, ППА або окремий аудіосемпл з кешу. Для збереження результатів стиснення пропонується використовувати такі бітові структури:

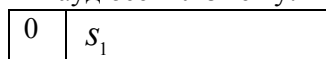
– РПА:



– ППА:



– окремий аудіосемпл з кешу:



Перший зліва біт цих структур дає змогу відрізнити псевдо-аналог від аудіосемпла з кешу. Наступний біт дозволяє відрізнити РПА від ППА і визначити, чи потрібно буде зчитувати різницю  $r$  наприкінці.

Розміри полів структури під  $j$  та  $p$  визначаються таким чином:

$$size(j) = \text{ceil}(\log_2 L),$$

$$size(p) = \text{ceil}(\log_2 l),$$

де  $\text{ceil}$  – операція округлення до найближчого більшого;  $L$ ,  $l$  – відповідно кількість аудіосемплів у словнику та у кеші.

Розміри полів під  $r$  та  $s_1$  однакові і дорівнюють бітовій розрядності  $b$  семплів аудіопотоку. Якщо значення  $r$  неможливо розмістити у  $b$  бітах:

$$\text{ceil}(\log_2 r) > b, \quad (2)$$

то такий РПА повинен бути відкинтий.

Наведемо алгоритм стиснення потоку аудіосемплів у вигляді послідовності дій. На початку стиснення задається розмір словника  $L$ , розмір кешу аудіосемплів  $l$ , а також допустима похибка значень аудіосемплів  $\delta$ . На вхід алгоритму подається потік аудіосемплів, на виході генерується потік результатів стиснення (архів).

Алгоритм стиснення.

- Крок 1. Заповнити словник  $L$  аудіосемплами з вхідного потоку та зберегти початковий стан словника у архіві.
- Крок 2. Обнулити поточне заповнення  $w$  кеша аудіосемплів:  $w = 0$  (кеш пустий).
- Крок 3. Поповнити кеш аудіосемплів з вхідного потоку з оновленням поточного заповнення кеша  $w$ .
- Крок 4. Відповідно до бітової структури для збереження РПА обчислити мінімальну кількість  $z$  аудіосемплів у РПА, при якій буде досягнута хоча б мінімальна компресія:

$$z = \text{ceil} \left[ \frac{2 + size(j) + size(p)}{b} + 1 \right].$$

- Крок 5. Відповідно до умови (1) для  $p$  починаючи з  $w$  до  $z$  шукати по всьому вмісту словника РПА довжиною  $p$ . При цьому відкидати РПА, які задовольняють умові (2). Серед всіх знайдених РПА обрати той, для якого

$$\sum_{i=1}^p |q_{i+j} - s_i| \rightarrow \min,$$

де  $q_{i+j}$  – аудіосемпли зі словника;  $j$  – позиція РПА у словнику;  $S_i$  – відповідні аудіосемпли з кеша.

- Крок 6. Якщо РПА на кроці 5 був знайдений, то:
- зафіксувати значення  $j$ ,  $p$  та різниці  $r$ ;
  - якщо  $r \neq 0$ , то зберегти РПА в архіві за допомогою відповідної бітової структури; інакше – зберегти РПА за допомогою структури для ППА;
  - вштовхнути знайдений РПА у словник (словник організований у вигляді черги FIFO).
- Якщо РПА не було знайдено, то:
- прийняти довжину РПА  $p = 1$ ;
  - зберегти в архіві тільки перший аудіосемпл з кеша за допомогою відповідної бітової структури;
  - вилучити цей аудіосемпл з кеша та вштовхнути його у словник.
- Крок 7. Вилучити оброблену частину кеша довжиною  $p$  та оновити заповнення кеша:  $w = w - p$ .
- Крок 8. Якщо не досягли кінця вхідного потоку, перейти до кроку 3.
- Крок 9. Кінець алгоритму.

Відтворення аудіосемплів з архіву є значно більш простою задачею. Необхідно лише прочитати з архіву початковий стан словника, а далі читати закодовані результати стиснення та виконувати декодування і відповідні дії:

- якщо одержані дані про РПА або ППА ( $r$ ,  $p$ ), то відтворити відповідний блок аудіосемплів на основі вмісту словника, записати його у відтворюваний аудіопотік і вштовхнути у словник;
- якщо одержаний окремий аудіосемпл, то записати його в явному вигляді у відтворюваний аудіопотік і вштовхнути у словник.

Вищезазначені дії виконуються до тих пір, поки не буде досягнуто кінець архіву.

Запропонований спосіб різницевих псевдо-аналогів (СРПА) був реалізований програмно мовою програмування С. Також, програмно був реалізований спосіб стиснення LZSS [4] (одна з модифікацій метода Лемпеля-Зіва). Були проведені машинні експерименти зі стиснення монофонічних звукових файлів формату RAW PCM, 8 біт/семпл, 22 кГц за допомогою цих способів. У табл. 1 та 2 наведені результати стиснення звукових файлів, що містять реальні музичні аудіозаписи:

- 1) voodooch.raw – пісня «Voodoo child» Джими Хендрікса (блюзово-рок-н- рольна музика).
- 2) hero.raw – пісня «Hero» групи Nickelback (рок-музика);
- 3) oskolok.raw – пісня «Осколок льда» групи АРИЯ (важка рок-музика);
- 4) disciple.raw – пісня «Disciple» групи Slayer (надважка рок-музика).

Результати стиснення за способом різницевих псевдо-аналогів одержані при наступних параметрах:

- 1) розмір словника  $L = 125$  аудіосемплів;
- 2) розмір кешу аудіосемплів  $l = 7$  аудіосемплів;
- 3) допустима похибка значень аудіосемплів  $\delta = 1$  (мінімальні втрати).

Результати стиснення за способом LZSS одержані при розміру словника 4096 символів і розміру буфера вводу/виводу 16 символів.

Таблиця 1

#### Розміри звукових файлів до і після стиснення

Звуковий файл	Розмір до стиснення, байт	Розмір після стиснення, байт	
		СРПА	LZSS
voodooch.raw	6 964 417	2 971 696	4 452 455
hero.raw	3 254 977	1 867 521	2 681 712
oskolok.raw	7 181 569	4 174 509	6 017 960
disciple.raw	4 757 761	3 955 270	4 874 740

Таблиця 2

#### Досягнуті ступені компресії звукових файлів

Звуковий файл	Ступінь компресії при стисненні	
	СРПА	LZSS
voodooch.raw	2,34	1,56
hero.raw	1,74	1,21
oskolok.raw	1,72	1,19
disciple.raw	1,20	0,98

Як можна бачити, для обох способів (СРПА, LZSS) має місце загальна тенденція: для аудіозаписів більш важкої музики ступені компресії є меншими. При цьому, навіть за умов мінімальних втрат, СРПА дозволяє досягти більшої компресії, ніж LZSS, при значно меншому розміру словника.

Єдиним недоліком СРПА є те, що відбувається спотворення аудіопотоку. З'являється шум з амплітудою  $\delta$  ( $\delta$  – допустима похибка значень аудіосемплів). Для аудіозаписів з 8 бітами на семпл шум є відчутним на слух вже при  $\delta > 1$ . Ця обставина поки що обмежує можливість використання розробленого способу для ефективного стиснення звукових даних.

У подальшому планується вдосконалення запропонованого способу з метою усунення шуму та підвищення ступені компресії.

### **Список використаних джерел**

1. Приходько С. Б. Сжатие звука на основе стохастических дифференциальных уравнений второго порядка // Вестник Херсонского государственного технического университета. — Херсон : ХГТУ. — 2002. — Вып. 2(15). — С. 386—388.
2. Ziv J., Lempel A. A universal algorithm for sequential data compression // IEEE Transaction on Information Theory — 1977. — Vol. 23(3). — P. 337—343.
3. Ziv J., Lempel A. Compression of individual sequences via variable-rate coding // IEEE Transaction on Information Theory — 1978. — Vol. 24(5). — P. 530—536.
4. Storer J. A., Szymanski T. G. Data compression via textual substitution // Journal of ACM. — 1982. — Vol. 29(4). — P. 928—951.
5. Welch T. A. A technique for high-performance data compression // IEEE Computer — 1984. — Vol. 17(6). — P. 8—19.
6. Bender P., Wolf J. New asymptotic bounds and improvements on the Lempel-Ziv data compression algorithm // IEEE Transactions on Information Theory — 1991. — Vol. 37(3). — P. 721—727.
7. Кириченко Н. Ф., Лепеха Н. П., Попив И. А. Допустимая аппроксимация функций дискретного аргумента и ее применение к сжатию информации // Проблемы управления и информатики. — 1998. — № 5. — С. 113—127.

**Serhiy USTENKO, Serhiy LUKYANCHIKOV**  
Mykolaiv

## **APPLICATION DICTIONARY COMPRESSION METHOD FOR COMPRESSION OF AUDIO DATA WITH ADJUSTABLE LOSS**

*The work is dedicated to the development of audio data compression method with adjustable losses using pseudo-difference analogs using dictionaries*

*Key words: dictionary, audio samples, the mechanism of compression, compression algorithm, lossy compression.*

**Сергей УСТЕНКО, Сергей ЛУКЪЯНЧИКОВ**  
г. Николаев

## **ПРИМЕНЕНИЕ СЛОВАРНОГО МЕТОДА СЖАТИЯ ДЛЯ КОМПРЕССИИ АУДИО ДАННЫХ С РЕГУЛИРУЕМЫМИ ПОТЕРЯМИ**

*Работа посвящена разработке метода сжатия аудио данных с регулируемыми потерями с помощью разностных псевдоаналогов с использованием словарей.*

*Ключевые слова: словари, аудиосемплы, механизм сжатия, алгоритм сжатия, сжатие с потерями.*

Стаття надійшла до редколегії 05.10.2016