

**О. М. Ткаченко, канд. техн. наук, доц.; І. Р. Арсенюк, канд. техн. наук, доц.;**  
**С. В. Хрущак, канд. техн. наук; С. П. Куліш**

## **ЗАСТОСУВАННЯ МЕТОДУ СУСІДНЬОГО СПУСКУ ДЛЯ ПОШУКУ ВЕКТОРІВ У КОДОВИХ КНИГАХ**

*Для ефективного використання пропускнув здатності каналів у сучасних системах цифрового зв'язку широко використовується ущільнення мовленнєвих сигналів. Найбільшого поширення набули методи ущільнення на основі векторного квантування із застосуванням кодових книг, оскільки вони, завдяки урахуванню кореляції між компонентами вектора, здатні забезпечити передачу інформації меншою кількістю бітів. Проте практичне застосування цих методів обмежується через необхідність використання великих обчислювальних ресурсів для їх реалізації.*

*У статті пропонується вдосконалений метод швидкого пошуку найближчого вектора у кодовій книзі. Основна ідея методу полягає у тому, щоб за рахунок попередньої структуризації кодової книги скоротити час пошуку в ній вектора, найближчого до вхідного. Структуризація кодової книги виконується на основі відношення мажорювання між векторами. Вводиться поняття рівнів мажоризації, за якими виконується структуризація. Безпосередній пошук найближчого вектора виконується за допомогою діаграм Вороного.*

*Розроблено математичну модель та структуру даних, на основі яких відбувається процес пошуку в кодовій книзі. Розглянуто кілька модифікацій метода швидкого спуску, що використовується у процесі пошуку найближчого вектора. На тестовому прикладі здійснено порівняння ефективності вказаних методів за показником середньої кількості вимірювань відстані. Показано, що найкращі результати забезпечує метод спрямованого пошуку на основі мажоризації. Проаналізовано негативні фактори, які здатні вплинути на ефективність квантування. Розглянуто можливість подальшого зниження складності обчислень за рахунок зменшення розмірів часового вікна, відведеного на пошук, яке досягається без суттєвої втрати продуктивності, що оцінювалася за спектральним спотворенням. Експериментальна перевірка підтвердила ефективність запропонованих методів.*

**Ключові слова:** ущільнення мовленнєвих сигналів, векторне квантування, кодові книги, сусідній спуск, діаграми Вороного, теорія мажоризації.

### **Вступ**

Зростання обсягів мультимедійної інформації, зокрема аудіоінформації, яка передається та обробляється у комп'ютерних системах, зумовила необхідність автоматизації процесів аналізу та пошуку даних. Тому в сучасних системах обробки аудіоінформації виникає необхідність автоматичного пошуку музичних творів (МТ) на основі аудіоконтенту в базах даних. Враховуючи великі обсяги аудіоінформації, що зберігається у корпусі МТ, велике значення має швидкість та надійність пошуку. Для підвищення надійності результатів пошуку у статті пропонується метод швидкого пошуку фрагмента аудіозапису у корпусі МТ з удосконаленою оцінкою ступеня близькості між невідомим аудіофрагментом і шаблонами, що дозволяє підвищити достовірність прийняття рішення під час пошуку за рахунок збільшення ступеня розбіжності власного МТ з іншими аудіозаписами.

Ущільнення мовленнєвих сигналів, що використовується в сучасних системах цифрового зв'язку, дозволяє підвищити ефективність використання смуги пропускання каналів і тим самим сприяє зниженню вартості на мобільні та інтернет-послуги. Натепер широко використовуються методи ущільнення на основі квантування сигналу з використанням кодових книг. Реалізація скалярного квантування (СК) лінійних спектральних пар (лінійних спектральних частот (line spectrum frequency, LSF) не потребує значних обчислювальних ресурсів та великої кількості пам'яті. Проте останнім часом у стандартах ущільнення мовлення часто використовували векторне квантування (ВК), що дозволяє отримати менше

спектральне спотворення під час кодування сигналу з однаковою кількістю бітів. Це пов'язано з тим, що ВК ефективніше використовує кореляцію між окремими компонентами вектора параметрів. Перевага зростає зі збільшенням кількості спільно квантованих компонентів, тобто зі збільшенням розміру вектора. Проте разом із цим збільшуються витрати пам'яті, а також час пошуку кодованого вектора у кодовій книзі, що зумовлює значні обмеження щодо практичного застосування ВК у режимі реального часу. Тому в наявних стандартах використовується субоптимальне кодування, коли 10-вимірний вектор параметрів ділиться на два або більше підвектори меншого виміру [1].

Для скорочення часу пошуку в [2] запропоновано кілька способів впорядкування векторів у кодовій книзі, що отримали назву методів швидкого векторного квантування (fast vector quantization methods). Показано, що складність розрахунків під час використання цих методів становить 25 % від складності розрахунків при повному пошуку без значної втрати продуктивності, яка оцінювалася за спектральним спотворенням.

У [3] запропоновано метод структурування кодових книг на основі відношення мажорювання, використання якого дозволило зменшити складність розрахунків у 5 – 6 разів порівняно з відповідною складністю повного пошуку.

Метод "сусіднього спуску", запропонований у [4], дозволяє скоротити час пошуку сусіднього вектора у 50 разів відносно часу повного пошуку у наведеному в роботі прикладі. Проте цей високий показник був досягнений із значним збільшенням середнього спектрального спотворення (на 0,22 дБ), що може бути неприйнятним для практичного використання.

Запропонований метод поєднує переваги методів, наведених у роботах [3] та [4]. Попередньо підготовлений список сусідів для кожного кластера згодом впорядковується відповідно до відношення мажорювання. Це дозволяє скоротити час пошуку найближчого вектора у кодовій книзі в 40 разів відносно часу повного пошуку без суттєвого збільшення спектрального спотворення.

Зменшення складності обчислень при кодуванні мовленнєвих сигналів можна досягти у результаті спрямованого пошуку найближчого вектора у кодовій книзі. Для цього потрібно розв'язати такі задачі:

- розробити модель та структуру даних, що зберігаються у кодових книгах;
- розробити метод пошуку найближчого вектора у створеній кодовій книзі;
- оцінити ефективність розробленого методу.

#### Математична модель та метод спрямованого пошуку

Припустимо кодова книга містить кінцеву множину векторів  $Q = \{Y_1, Y_2, \dots, Y_N\}$ ,  $Y_i = \{Y_{i1}, Y_{i2}, \dots, Y_{iM}\}$ . Таким чином, з кожним вектором  $Y_j$  у кодовій книзі пов'язаний індекс або кодове слово  $j$ , яке може бути записане як  $N$ -розрядне ціле число. На вхід квантизатора надходить вектор  $X = \{x_1, x_2, \dots, x_M\}$ .

У результаті кодування необхідно вибрати таке кодове слово  $j$ , яке мінімізує спотворення  $d(X, Y_j)$  (правило вибору найближчого сусіднього вектора). Для обчислення спотворення використовуватимемо Евклідову відстань:

$$d^2(X, Y_i) = \sum_{k=1}^M (x_k - y_{ik})^2 \quad (1)$$

Множина вхідних векторів  $X$ , які кодуються тим самим індексом  $j$ , утворюють діаграму Вороного:

$$V_j = \{X : d(X, Y_j) \leq d(X, Y_i); \forall i \in I\},$$

де  $I = \{1, 2, \dots, N\}$  – множина індексів. Альтернативним варіантом завдання регіона Вороного

є така формула:

$$V_j = \{ X : d(X, Y_j) \leq d(X, Y_i); \forall i \in A_j \},$$

де  $A_j = \{ i : V_i \cap V_j \neq \emptyset \}$  – множина кодових слів, у яких діаграми Вороного є суміжними з даним регіоном.

Найпростішим методом пошуку в кодовій книзі вектора, найближчого до  $X$ , є повний пошук. При цьому для кодової книги розміром  $N$  необхідно  $N$  разів обчислити відстань згідно з (1), що у багатьох випадках неприйнятно для практичної реалізації. З іншого боку, за такого підходу потрібно виконання великого обсягу зайвої роботи, оскільки не використовуються результати попередніх обчислень. Метод сусіднього спуску [4] базується на використанні таблиць суміжності  $A_j$ , підготовлених заздалегідь для кожного регіону  $V_j$ . Основна ідея цього методу полягає у тому, що результати обчислення відстані використовуються для переходів до суміжних регіонів у напрямку оптимуму. Перехід від кодового слова  $j$  до кодового слова  $l$  відбувається за виконання умови

$$d(X, Y_l) \leq d(X, Y_j), l \in A_j \quad (2)$$

Було розглянуто дві основні модифікації методу сусіднього спуску. У першій з них (full neighbor descent, FND) спочатку обчислювалася відстань від вхідного вектора до усіх суміжних кодових слів для цієї діаграми Вороного, після чого приймалося рішення про перехід до кодового слова з мінімальною відстанню або про завершення пошуку (повний обхід сусідів).

У другій модифікації (random neighbor descent, RND) сусідні кодові слова перевірялися у випадковому порядку і перехід до сусіднього кодового слова відбувався відразу після виконання умови (2) (випадковий обхід сусідів).

Порівняння цих підходів проведено для кодових книг, що містять  $N = 4096$  векторів розмірністю  $M = 5$ . Найкращі результати отримані при випадковому обході сусідів [4]. В обох випадках як початкова гіпотеза вибиралося вихідне кодове слово, знайдене для попереднього фрейму, яке дозволяло використовувати кореляцію між сусідніми фрагментами мовленнєвого сигналу.

Наведені у [4] результати дозволяють зробити припущення, що ефективність пошуку тим вища, чим менше часу витрачається на пошук сусіда під час переходу згідно (2) для кожного кодового слова  $j$ . Фактично на цьому етапі виконується витратний лінійний пошук і його негативний вплив на загальні результати збільшується зі зростанням числа сусідів. Так, для згаданої вище кодової книги кількість сусідів коливається у межах 40 – 160 і становить у середньому 70 для кожного регіону. За таких умов лінійний пошук можна вважати неефективним. Випадковий обхід сусідів зменшує час лінійного пошуку і тим самим підвищує загальну продуктивність.

Ефективність такого підходу суттєво залежить від того, в якому порядку здійснюється обхід сусідніх кодових слів. При цьому логічно зробити припущення про те, що, упорядкувавши обхід векторів для кожного регіону, можна зменшити час лінійного пошуку і досягти подальшого скорочення обчислювальних витрат.

Один із можливих підходів до впорядкування кодових книг ґрунтується на використанні теорії мажоризації [5], застосування якої для прискорення пошуку найближчого вектора наведено у [3]. Нехай задано вектори  $Y = \{ y_1, y_2, \dots, y_n \}$ , та  $Y' = \{ y'_1, y'_2, \dots, y'_n \}$ ,  $Y, Y' \in Q$ , компоненти яких упорядковано за незростання. Вважаємо, що  $Y$  мажорується  $Y'$ , або  $Y'$  мажорує  $Y$  (позначається  $Y < Y'$ ), якщо виконується така умова [5]:

$$\sum_{i=1}^k y_i \leq \sum_{i=1}^k y'_i, k = 1, 2, \dots, M - 1,$$

$$\sum_{i=1}^M y_i = \sum_{i=1}^M y'_i \quad (3)$$

Упорядкування, яке задається відношенням мажорювання (3), називають мажоризацією. Якщо виконується нерівність

$$\sum_{i=1}^k y_i \leq \sum_{i=1}^k y'_i, k = 1, 2, \dots, M, \quad (4)$$

то це означає, що  $Y$  слабо мажорюється  $Y'$ , або  $Y'$  слабо мажоруює  $Y$  (позначається  $Y \prec_w Y'$ ).

У роботі [3] показано, як можна застосувати впорядкування, що задається відношенням мажорювання (4), для побудови структурованих векторних кодових книг. У цьому випадку використовуємо (4) для впорядкування сусідів кожного кодового слова  $j$ . При цьому таблицю суміжності  $A_j$  необхідно розбити на окремі класи, що формуються згідно з рівнями мажоризації, які формуються за таким правилом.

Вважатимемо, що рівень мажоризації  $L_i$  мажорюється рівнем мажоризації  $L_k$ , якщо для кожного вектора  $Y$ , що належить  $L_i$ , на рівні  $L_k$  знайдеться вектор  $Y'$ , що слабо мажоруює  $Y$ , або формально:

$$\forall Y, Y \in L_i, \exists Y', Y' \in L_k, Y \prec_w Y' \Rightarrow L_i \prec L_k; i, k \in A_j \quad (5)$$

Розбиття таблиці суміжності  $A_j$  на класи згідно із заданим критерієм (5) називатимемо структуризацією. Структуризація виконується для скорочення обчислювальних витрат на пошук під час переходу від кодового слова  $j$  до кодового слова  $l$  згідно з (2).

Спрямований пошук у межах запропонованого підходу відбувається так:

1. Визначається рівень мажоризації  $L_k$ , на який потрапляє вхідний вектор  $X$ .
2. З цього рівня  $L_k$  відбувається пошук вектора  $Y_l$  для переходу до кодового слова  $l$  згідно з (2).
3. Якщо такий вектор не знаходиться на рівні  $L_k$ , пошук продовжується на сусідніх рівнях  $L_{k-1}, L_{k+1}, L_{k-2}, L_{k+2}, \dots$ .
4. Якщо для жодного вектора  $Y_i, i \in A_j$  умова (2) не виконується, пошук завершується та вектор  $Y_j$  вважається найближчим до вхідного вектора  $X$ . Інакше відбувається перехід та пункти 1 – 4 знову повторюються для кодового слова  $l$ .

Слід зауважити, що пошук рівня мажоризації для вхідного вектора  $X$  не потребує значних витрат часу та зводиться до простої процедури перевірки виконання умови (4). Так само, як у [4], за початкову гіпотезу вибирається вихідне кодове слово, знайдене для попереднього кадру. Запропонований підхід фактично є модифікацією методу сусіднього спуску, але отримав назву метода спрямованого пошуку з мажоризацією (СПМ), що, на наш погляд, точніше відповідає суті методу.

### Моделювання пошуку найближчого вектора методом СПМ

З метою експериментальної перевірки запропонованого підходу в середовищі Visual Studio розроблено програмне забезпечення для побудови таблиць суміжності та спрямованого пошуку найближчого вектора. Побудову векторних кодових книг виконано за умов, описаних у [6]. Для того, щоб забезпечити можливість порівняння з результатами, наведеними у [4], 10-вимірний вектор LSF-параметрів було розбито на дві частини  $5 \times 5$ , кожна кодова книга містила по 4096 підвекторів. Таким чином, обсяг даних, необхідних для представлення спектральної інформації у межах кадру, становив 24 біти.

Для тестування обраний текст, що є уривком із повісті Оксани Забужко та складається з 20000 векторів. Результати пошуку оцінювалися за кількістю пропущених векторів, Наукові праці ВНТУ, 2025, № 1

найбільш близьких до вхідного, вираженого у відсотках, а також сумарного спектрального спотворення  $SD$  [1, 6].

У табл. 1 наведено значення середньої складності обчислень  $C$ , середньої кількості обчислень відстаней (КОВ)  $n$  та глибини пошуку  $Tr$ , отримані різними методами. Глибиною пошуку називатимемо число переходів між сусідніми регіонами, необхідне для досягнення оптимуму. Як видно з табл. 1, метод СПМ має перевагу перед іншими внаслідок істотного скорочення середньої КОВ  $n$  для кожного регіону, що виконуються при переході до сусіднього регіону.

Таблиця 1

Метод	C		Tr		$\bar{n}$
	Середнє	Максимальне	Середнє	Максимальне	
FND	140,76	432	2,47	7	56,99
RND	104,46	270	4,94	19	21,15
СПМ	89,26	214	4,60	20	19,40

На рис. 1 наведено залежність числа знайдених найближчих векторів  $N$  від глибини пошуку  $Tr$ . Як бачимо, глибина пошуку при RND та СПМ дещо зростає. Тим не менш, середня складність обчислень  $C$ , що визначається КОВ до векторів у кодовій книзі, зменшується:

$$C = n \cdot Tr,$$

де  $n$  – середня КОВ для кожного регіону;  $Tr$  – середня кількість переходів між сусідніми регіонами.

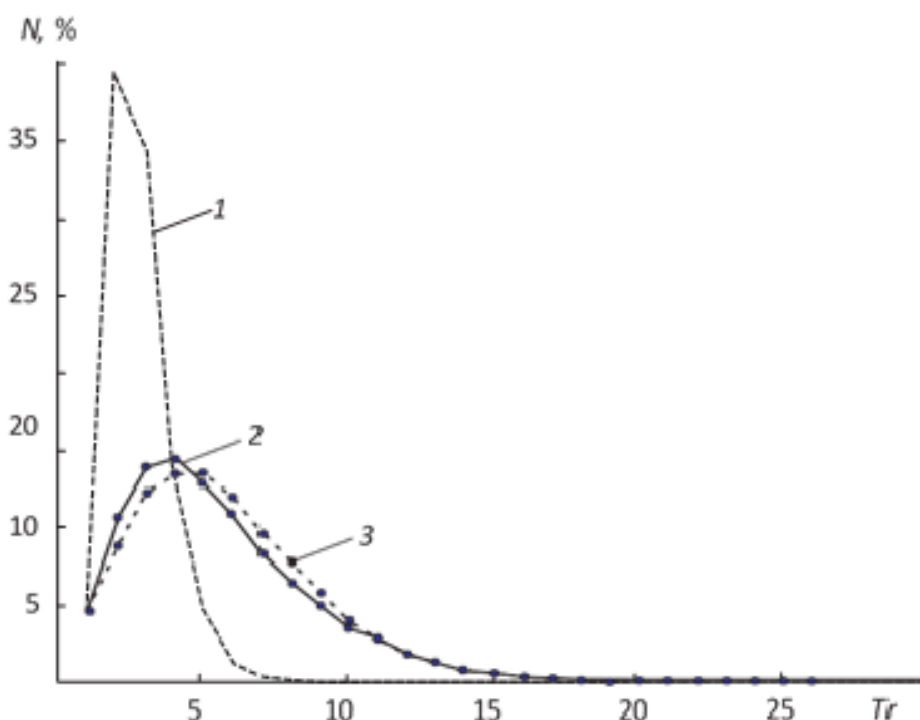


Рис. 1. Залежність числа знайдених векторів від глибини пошуку:  
1 – FND,  $n = 56,99$ ; 2 – СПМ,  $n = 19,40$ ; 3 – RND,  $n = 27,15$

Табл. 1 показує, що максимальне та середнє значення КОВ істотно відрізняються. Якщо для оцінювання складності розрахунків слід зосередити увагу на значенні середньої, а не на максимальній КОВ, можна скоротити час, виділений на пошук, у 2,5 – 3 рази. Однак середнє значення КОВ не є інформативною характеристикою для ущільнення мовленнєвих сигналів

у режимі реального часу, оскільки час, заощаджений на розрахунках в одному фреймі, не може бути використаний в іншому. Тому складність розрахунків повинна бути константою для різних фреймів, яка визначається часовим інтервалом (вікном), виділеним для пошуку найближчого вектора у кодовій книзі. Логічно припустити, що цей інтервал є значенням, пропорційним КОВ, і оцінити кількість пропущених векторів та спектральне спотворення для різних вікон. Результати такого дослідження наведено у табл. 2 та на рис. 2.

Таблиця 2

КОВ	Пропущені вектори MV, %		Спектральне спотворення SD, дБ	
	RND	СПМ	RND	СПМ
40	64,50	25,93	1,028	0,915
60	42,60	8,43	0,940	0,866
80	23,08	1,88	0,893	0,857
100	10,88	0,25	0,871	0,854
120	4,73	0,05	0,862	0,854
140	1,68	0,00	0,857	0,854

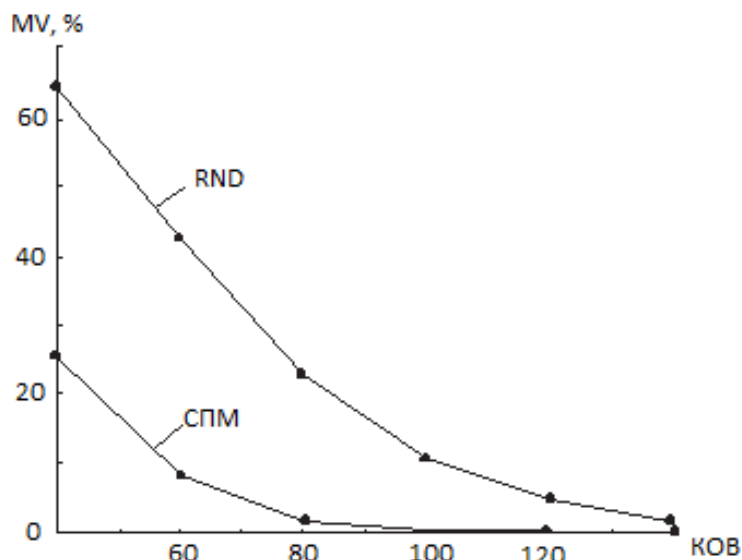


Рис. 2. Число пропущених векторів для різних вікон

Як можна побачити, результати, прийнятні для практичного використання, досягаються для вікон, значно менших за розміром, ніж максимальні значення КОВ, наведені у табл. 1. Це пов'язано з тим, що найбільша КОВ припадає на перевірку останньої діаграми  $V_j$ , коли фактично вже знайдено найближчий вектор і перехід не здійснюється. Слід також зазначити, що метод СПМ забезпечує зменшення кількості пропущених векторів та спектрального спотворення порівняно з методом RND для вікон однакового розміру.

### Висновки

Запропонований метод СПМ є вдосконаленим варіантом методу сусіднього спуску RND. Як і метод RND, він дозволяє прискорити процес вибору необхідних векторів з кодових книг у процесі передачі мовлення в результаті створення таблиць суміжності на підготовчій стадії. У той же час, метод СПМ забезпечує скорочення часу пошуку найближчого сусіднього вектора на 30 % порівняно з методом RND через попереднє впорядкування векторів у таблицях суміжності відповідно до відношення мажоризації. При цьому загальне спектральне спотворення не збільшується і становить 0,854 дБ у розглянутому прикладі.

## СПИСОК ЛІТЕРАТУРИ

1. Gersho A., Gray R. Vector Quantization and Signal Compression. *Springer*. 2012. Vol. 159. New York. URL: <https://doi.org/10.1007/978-1-4615-3626-0>.
2. CompGS: Smaller and Faster Gaussian Splatting with Vector Quantization / Navaneet K. L. et al. *Computer Vision – ECCV 2024*. ECCV 2024. Lecture Notes in Computer Science. 2024. Vol 15090, Springer, Cham. P. 330–349. [https://doi.org/10.1007/978-3-031-73411-3\\_19](https://doi.org/10.1007/978-3-031-73411-3_19).
3. Швидкий пошук при векторному квантуванні лінійних спектральних частот / Біліченко Н. О. та ін. *Реєстрація, зберігання і обробка даних*. 2008. Т. 10, № 2. С. 37–47.
4. Agrell E. Spectral coding by fast vector quantization. *Proc. IEEE Workshop on Speech Coding for Telecommunications*. Sainte-Adel'e, Quebec, Canada. 1993. P. 61–62.
5. Barry C. A. Majorization: Here, There and Everywhere. *Statistical Science Journal*. 2007. Vol. 22, №3. P. 407–413.
6. LSF-вокодер на основі векторного квантування / Біліченко Н. О. та ін. *Реєстрація, зберігання і обробка даних*. 2007. Т. 9, № 1. С. 35–41.

Стаття надійшла до редакції 14.02.2025.

Стаття пройшла рецензування 11.03.2025.

**Ткаченко Олександр Миколайович** – канд. тех. наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет.

**Арсенюк Ігор Ростиславович** – канд. тех. наук, доцент кафедри комп'ютерних наук, e-mail: [air@vntu.edu.ua](mailto:air@vntu.edu.ua).

Вінницький національний технічний університет.

**Хрущак Сергій Вікторович** – канд. тех. наук, старший викладач кафедри комп'ютерних наук.

Вінницький національний аграрний університет.

**Куліш Сергій Павлович** – аспірант кафедри програмного забезпечення.

Вінницький національний технічний університет.