УДК 621.39

О. М. Ткаченко;
О. Ф. Грійо Тукало

ДВОЕТАПНА СТРАТЕГІЯ ПОШУКУ У ВЕКТОРНИХ КОДОВИХ КНИГАХ ДЛЯ УЩІЛЬНЕННЯ МОВЛЕННЯ

Розроблено підхід, який дозволяє поєднати переваги методів швидкого пошуку найближчого сусіда у кодових книгах із застосуванням зваженої евклідової метрики. Запропонована двоетапна стратегія пошуку полягає в тому, що на першому етапі з кодової книги, структурованої на основі бінарного дерева, за евклідовою метрикою відбирається невелика кількість векторів (кандидатів), з яких на другому етапі із використанням ваг обирається вектор, найближчий до вхідного. Отримано аналітичні залежності кількості кандидатів та операцій, необхідних для пошуку найближчого вектора. Наведено результати експериментів, що підтверджують ефективність запропонованого підходу.

Вступ

Кодування мовлення з метою його ущільнення широко застосовується в системах цифрового зв'язку, оскільки дозволяє підвищити ефективність використання смуги частот у таких системах. Більшість сучасних методів кодування базуються на лінійному прогнозуванні (ЛП) параметрів, які описують спектральну інформацію, що міститься у мовленнєвому сигналі. ЛП-параметри звичайно перетворюють у лінійні спектральні частоти (LSF) для більш ефективного квантування й інтерполяції.

У багатьох роботах досліджувалися різні методи квантування LSF. Скалярне квантування (СК) LSF не потребує ані значних обчислювальних ресурсів, ані великих витрат пам'яті для своєї реалізації, проте останнім часом у стандартах ущільнення мови частіше використовують векторне квантування (ВК). Останнє дозволяє отримати менше спектральне спотворення, порівняно з СК, за умови кодування сигналу однаковою кількістю бітів. Цей виграш зумовлений тим, що ВК більш ефективно використовує кореляцію між окремими складовими вектора параметрів. Перевага зростає зі зростанням кількості компонентів, що квантується разом, тобто зі збільшенням розмірності вектора. Проте разом з цим зростають витрати пам'яті, а також час на пошук найближчого вектора в кодовій книзі (таблиці, що містить типові набори значень параметрів). Це призводить до неможливості практичного застосування безпосереднього квантування 10-вимірних векторів параметрів. Тому в діючих стандартах застосовують субоптимальне кодування, коли вектор параметрів розбивають на два або більше підвекторів меншої розмірності [1].

З ВК параметрів пов'язано ще одне питання — вибір метрики для вимірювання відстані між вхідним вектором і векторами кодової книги. Безпосереднє використання спектрального спотворення [1], за допомогою якого прийнято оцінювати похибку квантування, на практиці неможливо через надмірну обчислювальну складність. З іншого боку, застосування найпростішої евклідової метрики (ЕМ) часто призводить до вибору не кращого варіанту при кодуванні вхідного вектора. Тому для вимірювання відстані було запропоновано використовувати зважену евклідову метрику (ЗЕМ), а також досліджувалися різні варіанти формування ваг. Так, у [2] в формулу для вимірювання відстані введено вагові коефіцієнти, пропорційні спектральній потужності для цього компонента LSF. У [3] для формування ваг запропоновано використовувати спектральну чутливість відносно цього LSF. У [4] з цією ж метою введено так звану узагальнену вагову функцію. Експериментальні результати показали практичну доцільність застосування описаних вище методів.

Також у низка робіт розглядається зменшення обчислювальної складності в процесі вибору найближчого сусіда з кодової книги. Так, з метою зменшення часу пошуку в [5] запропоновано декілька підходів до впорядкування векторів у кодовій книзі, названих авторами методами швидкого векторного квантування. У [6] розроблено метод структуризації кодових книг на основі відношення мажорування. У [7] запропоновано метод прискорення пошуку найближчого сусіда на основі діаграми Вороного.

Спільним для всіх вищезгаданих методів є те, що зменшення часу пошуку в процесі кодування, що відбувається в реальному масштабі часу, досягається за рахунок попередньої обробки (структуризації) кодових книг. Такий підхід унеможливлює подальше застосування для пошуку найближчого сусіда ЗЕМ, оскільки ваги розрядів функціонально залежать від вхідного вектора. Застосування ж простої ЕМ під час швидкого пошуку призводить до втрати продуктивності, що виражається у збільшенні спектрального спотворення.

В цій статті описується підхід, який поєднує переваги методів швидкого пошуку найближчого сусіда із застосуванням ЗЕМ. Пропонується двоетапна стратегія пошуку, суть якої полягає у тому, що на першому етапі на основі пошуку по бінарному дереву з кодової книги за ЕМ відбирається невелика кількість векторів (кандидатів), з числа яких на другому етапі із використанням ЗЕМ відбирається вектор, найближчий до вхідного. У наступних розділах статті виводиться залежність кількості кандидатів від ваг коефіцієнтів LSF, а також проводиться оцінювання кількості операцій для пошуку найближчого вектора з урахуванням додаткових витрат на другому етапі. Також наводяться експериментальні результати, що засвідчують переваги запропонованого підходу.

Метою роботи є зменшення складності обчислень при кодуванні мовленнєвих сигналів за рахунок скорочення часу пошуку найближчого вектора у кодовій книзі.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

— проаналізувати фактори, що впливають на ефективність пошуку;

— розробити двоетапну стратегію пошуку найближчого сусіда;

— отримати аналітичні залежності для кількості кандидатів і операцій в межах запропонованого підходу;

— провести експериментальну перевірку отриманих залежностей.

Пошук найближчого вектора в кодовій книзі на основі бінарного дерева

Векторні кодові книги (КК) містять набори шаблонів векторів параметрів (коефіцієнтів) для представлення мовленнєвого сигналу, що передається. Як параметри використовуються лінійні спектральні частоти (LSF), що отримують на основі ЛП-параметрів. У [2] доведено, що LSF-коефіцієнти є найефективнішими для подальшої квантизації та передачі, стійкими до канальних перешкод.

Нехай КК містить кінцеву множину векторів $\mathbf{Q} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$, $\mathbf{p}_i = (p_{i1}, p_{i2}, \ldots, p_{iM})$. Таким чином, з кожним вектором $\mathbf{p}_j$ у КК пов'язаний індекс, або кодове слово j, що може бути записано як n-розрядне ціле число. На вхід квантизатора надходить вектор $\mathbf{x} = (x_1, x_2, \ldots, x_M)$. В результаті кодування необхідно вибрати таке кодове слово j, що мінімізує спотворення $d(\mathbf{x}, \mathbf{p}_j)$ (правило вибору найближчого сусіднього вектора).

$$\mathbf{R}_j = \{\mathbf{x} : d(\mathbf{x}, \mathbf{p}_j) \le d(\mathbf{x}, \mathbf{p}_i);\ \ \forall i \in I, i \ne j\}, \tag{1}$$

де $I = \{1, 2, \ldots, n\}$ — множина індексів; n — кількість векторів у КК.

Бінарним деревом (БД) називається дерево $\mathbf{T} = (\mathbf{V}, \mathbf{E})$, де ($\mathbf{V} = \{v_1, v_2, \ldots, v_h\}$ — множина вершин, $\mathbf{E} = \{e_1, e_2, \ldots, e_m\}$ — множина пар елементів множини $\mathbf{V}$, або множина ребер, $m = h - 1$), яке має виокремлений корінь та два піддерева (праве та ліве), що не перетинаються, або є пустою множиною вершин: $\mathbf{T} \le \{\mathbf{T}_1, \mathbf{T}_2\}$; $\mathbf{T}_1 \not\subset \mathbf{T}_2$. Таким чином, бінарне дерево — це структура даних, в якій кожна вершина має не більше двох нащадків (правого та лівого).

В БД присутні два типи вершин: термінальні (листи) та нетермінальні (вузли). Кожен лист дерева може містити не більше одного вектора КК, тобто $l_к \subset \mathbf{p}_i$ або $l_к \subset \varnothing$, де $l_к$ — деякий лист БД (тобто в нашому випадку кількість листів $k \ge 4096$).

Використання БД дозволяє суттєво зменшити кількість необхідних вимірювань для пошуку найближчого сусіднього вектора ($\log_2 n$ замість n під час повного пошуку). Проте, при цьому зростає спектральне спотворення, оскільки пошук по БД не гарантує знаходження дійсно найближчого вектора відповідно до формули (1) та рис. 1.

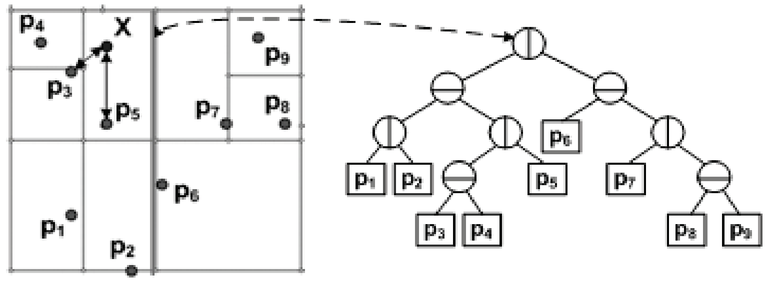

Рис. 1. Ілюстрація впорядкування КК на основі бінарного дерева

У роботах [8, 9] запропоновано вдосконалену процедуру пошуку по БД: вона має дві фази пошуку — пряму та зворотну. Під час прямого пошуку фіксуються всі відстані до вузлів d_i. Пряма фаза завершується обчисленням відстані до відповідного листа $D_k = D_{\min}$. Після цього починається зворотна фаза пошуку, при цьому обчислюються відстані D_k лише до тих листів дерева, які можуть забезпечити виконання $D_k < D_{\min}$. Якщо ця умова виконується, вважається $D_{\min} = D_k$. На зворотній фазі також можуть обчислюватися відстані до вузлів. Пошук найближчого вектора в КК, впорядкованій на основі БД, показано на рис. 2 для двовимірного випадку.

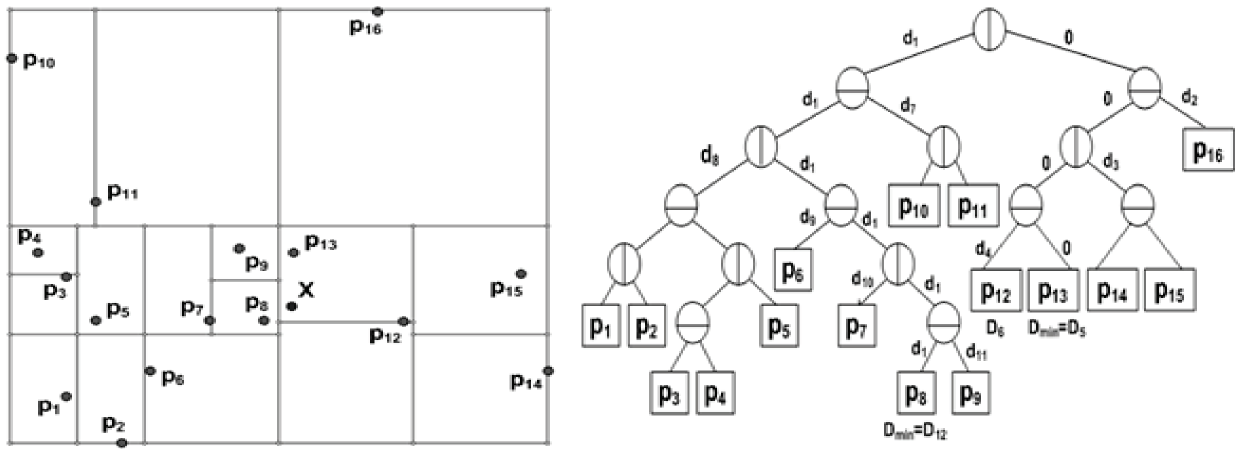

Рис. 2. Ілюстрація пошуку найближчого вектора по бінарному дереву

Розглянута процедура пошуку гарантує знаходження найближчого сусіднього вектора згідно з (1), проте потребує значно більшої кількості вимірювань відстані, ніж $\log_2 n$. З метою скорочення часу в [10] запропоновано для обчислення відстані до вузлів скористатися ефективнішою формулою

$$D_k^{\,2} = \left(D_k^{\,2}\right)' - \left(p_k' - x_k\right)^2 + \left(p_k - x_k\right)^2, \qquad (2)$$

де $\left(D_k^{\,2}\right)'$ — раніше обчислена відстань до певного вузла дерева. Застосування (2) можливе завдяки тому, що на кожному кроці просування по дереву змінюється лише одна з координат. Обчислення відстані згідно (2) потребує лише одного множення та трьох додавань, оскільки $\left(p_k' - x_k\right)^2$ можна зберігати у структурі даних дерева. Обчислення відстані до листів виконується за формулою

$$d^{(E)}\left(\mathbf{x},\, \mathbf{p}_i\right) = \sum_{k=1}^{M} \left(x_k - p_{ik}\right)^2 . \qquad (3)$$

Також забезпечено можливість отримання не одного найближчого вектора, а деякої множини $\mathbf{C}$, впорядкованих за зростанням відстані [10]:

$$\mathbf{C} \subseteq \mathbf{Q} = \{\mathbf{p}_1,\ \mathbf{p}_2, \ldots, \mathbf{p}_t\},\ t \le n, |\mathbf{C}| = t, \tag{4}$$

де $|\mathbf{C}|$ — потужність множини $\mathbf{C}$, або кількість найближчих векторів.

Це можливо за рахунок того, що дані про відстань до вже пройдених листів зберігаються і одночасно здійснюється їх впорядкування за зростанням відстані від вхідного вектора. Завдяки цьому додаткове знаходження декількох найближчих векторів не потребує багато часу.

Крім того, якщо погодитись з тим, що знайдений вектор не обов'язково найближчий згідно з (1), але є досить близьким (з певною похибкою $e > 0$), пошук можна ще прискорити. Вектор КК є $(1+e)$-найближчим вектором до $\mathbf{x}$, якщо виконується:

$$d(\mathbf{x},\ \mathbf{p}) \le \frac{d(\mathbf{x},\ \mathbf{p}^*)}{(1+e)}; \quad \mathbf{p}, \mathbf{p}^* \in \mathbf{Q}, e > 0 \Big\}, \tag{5}$$

де $\mathbf{p}^*$ — найближчий вектор КК, знайдений до цього моменту.

Алгоритм пошуку $(1+e)$-найближчого вектора такий: спочатку здійснюється визначення листа, якому належить вхідний вектор, здійснюючи спуск по дереву (на прямій стадії пошуку), після цього здійснюється запам'ятовування листових вузлів у черзі в порядку збільшення відстані до $\mathbf{x}$ та визначається відстань до векторів КК, що відповідають листам. Як тільки відстань від $\mathbf{x}$ до поточного листа відповідає умові (5), пошук зупиняється [11]. Якщо задати $e = 0$, буде здійснюватись пошук дійсно найближчих векторів згідно з (1).

Отже, впорядкування векторів КК на основі БД дозволяє:

— досягти суттєвого скорочення часу пошуку;

— отримати додатково кілька кандидатів на найближчий вектор, впорядкованих за зростанням відстані від вхідного вектора.

Розробка двоетапної стратегії пошуку

Щоб кількісно оцінити спотворення, що вноситься за рахунок квантизації сигналу, прийнято використовувати спектральне спотворення SD (Spectral Distortion), яке вимірюється в децибелах, враховуючи особливості людського слуху [2]. SD обчислюється для кожного фрейму (фрагменту довжиною 20 мс) тестових даних та знаходиться його середнє значення SD_{av}:

$$SD_i^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[10\log_{10}\left(P_i(\omega)\right) - 10\log_{10}\left(\widehat{P}_i(\omega)\right)\right]^2 d\omega, \tag{6}$$

де $P_i(\omega)$, $\widehat{P}_i(\omega)$ — спектральні щільності відповідно початкового і квантованого векторів для i-го фрейму; ω — радіальна частота.

Вважається, що квантизація не вносить помітних спотворень, якщо $SD_{av} \approx 1dB$.

Описаний раніше варіант структуризації КК з метою прискорення пошуку (як і інші методи швидкого пошуку) базується на використанні ЕМ і не дозволяє застосувати ЗЕМ (при якій можна отримати менше SD), оскільки в цьому випадку здійснюється попередня обробка векторів КК. Відстань за ЗЕМ обчислюється за формулою:

$$d^{(WE)}(\mathbf{x},\ \mathbf{p}_i) = (\mathbf{x} - \mathbf{p}_i)^T \begin{bmatrix} w_1 & & 0 \\ & w_2 & \\ & & \ddots & \\ 0 & & w_M \end{bmatrix} (\mathbf{x} - \mathbf{p}_i) = \sum_{k=1}^{M} w_k \left(x_k - p_{ik}\right)^2, \tag{7}$$

де w_k — k-й ваговий коефіцієнт, $k = 1, \ldots, M$.

Оскільки вагові коефіцієнти w_k обчислюються на основі вхідного вектора, то в даному випадку попередня обробка виконуватися не може. Тому необхідно поєднати швидкий пошук із застосуванням ЗЕМ, щоб досягти виграшу в швидкодії і зберегти при цьому мінімально можливе спектральне спотворення. Для цього пропонується двоетапна стратегія пошуку, яка реалізується таким чином:

1. За методом пошуку по бінарному дереву з КК за евклідовою мірою відбирається $|\mathbf{C}|$ найближчих сусідів до вхідного вектора з похибкою e.

2. Застосовуючи зважену евклідову міру, з множини кандидатів $\mathbf{C}$ обирається один вектор, найближчий до вхідного.

При цьому постає питання: якої кількості кандидатів $|\mathbf{C}|$ достатньо для знаходження на другому етапі пошуку за ЗЕМ дійсно найближчого вектора.

Оцінювання кількості кандидатів

Оскільки за ЗЕМ різні розмірності мають різну вагу, вектор КК, що є найближчим за ЕМ, може виявитися не найкращим кандидатом при врахуванні ваг. Таку ситуацію для двовимірного випадку зображено на рис. 3, де за ЕМ (див. формулу (3)) точка (вектор в двомірному просторі) $\mathbf{p}_n$ є найближчою згідно (1), але в результаті використання ЗЕМ (див. формулу (7)) найближчою виявляється точка $\mathbf{p}'_n$ за рахунок того, що

$$\frac{r_1}{r_2} = \frac{w_{\min}}{w_{\max}}. \tag{8}$$

Відповідно до (8) необхідно охопити множину кандидатів, що обмежується колом радіуса r_2.

Отже, визначення на другому етапі пошуку необхідної і достатньої кількості кандидатів для знаходження одного найближчого вектора КК за ЗЕМ вимагає здійснення таких етапів (див. рис. 3):

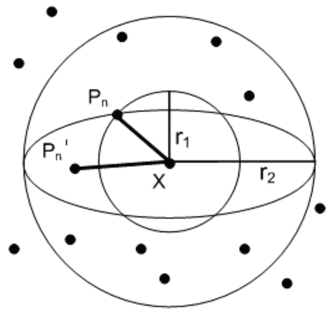

Рис. 3. Пошук найближчого сусіда у двовимірному випадку

1. Визначення вагових коефіцієнтів вхідного вектора $w_k,\ k = 1,\dots,M$.

2. Визначення евклідової відстані $d_n^{(E)}$ до найближчого вектора $\mathbf{p_n}$:

$$d_{nk}^{(E)} = \left(x_k - p_{nk}\right)^2, \quad d_n^{(E)} = \sum_{k=1}^{M} d_k^{(E)} = r_1^2. \tag{9}$$

3. Визначення зваженої евклідової відстані $d_n^{(WE)}$:

$$d_n^{(WE)} = \sum_{k=1}^{M} w_k \left(x_k - p_{nk}\right)^2. \tag{10}$$

4. Визначення граничної відстані пошуку (оцінка для найгіршого випадку):

$$d_{\lim}^{(E)} = \frac{d_n^{(WE)}}{w_{\min}^2} = r_2^2, \quad w_{\min} = \min_k w_k. \tag{11}$$

5. Визначення кількості кандидатів $|\mathbf{C}|$:

$$\mathbf{C} \subset \mathbf{P} = \left\{\mathbf{p}_i : d\left(\mathbf{x},\ \mathbf{p}_i\right) < d_{\lim}^{(WE)}\right\}. \tag{12}$$

Варто зауважити, що виведена вище гранична відстань $d_{\lim}^{(E)}$, що обмежує область пошуку кандидатів, є верхньою межею.

Оцінювання кількості операцій в процесі пошуку

Зазначимо, що надалі буде йтися про кількість операцій, обмежену зверху, тобто максимально необхідну. Дійсна кількість операцій може бути значно меншою.

Для пошуку найближчого сусіда за ЕМ по бінарному дереву необхідна кількість операцій визначається як:

$$N_1 = O\left(3c_{M,e} M \cdot \log n\right), \quad c_{M,e} = \left(1 + {}^{M}\!/_{e}\right)^{M}\Bigg), \tag{13}$$

де O — верхня межа обчислювальної складності; M — розмірність простору, $M = 10$.

Щоб додатково отримати $|\mathbf{C}|$ кандидатів, необхідна кількість операцій становить:

$$N_2 = O\left(3 \cdot |\mathbf{C}| \cdot M \cdot \log n\right). \quad (14)$$

Для визначення одного найближчого вектора за ЗЕМ на другому етапі необхідна кількість операцій:

$$N_3 = O\left(4 \cdot |\mathbf{C}| \cdot M\right). \quad (15)$$

Таким чином, загальна кількість операцій, необхідна для пошуку найближчого вектора, має три складові: $N = N_1 + N_2 + N_3$, що визначаються за формулами (13)—(15).

Експериментальні дослідження і результати

Для дослідження ефективності запропонованого підходу було використано базу даних з 90000 фреймів мовленнєвих даних 90 дикторів (45 жінок і 45 чоловіків). З них 80000 фреймів мовлення були використані для тренування, решта (10000) — для проведення дослідження. Частота дискретизації становила 8 кГц, тривалість фреймів — 20 мс.

Ефективність запропонованого в даній статті методу пошуку оцінювалася за рівнем спектрального спотворення (яким визначається якість відтворення мовлення, що передається) та кількістю здійснених під час пошуку операцій.

Табл. 1 дозволяє показати як змінюється спектральне спотворення залежно від кількості кандидатів (1, 5—100) та заданої похибки.

Таблиця 1

Спектральне спотворення

Похибка вибору найближчого вектора з КК, *e*	Оцінювання спектрального спотворення, *SD* (дБ) при визначенні найближчого вектора, використовуючи					
	евклідову метрику	зважену евклідову метрику				
	1	5	10	20	50	100
0	1,33	1,25	1,24	1,24	1,24	1,24
1	1,38	1,26	1,24	1,24	1,24	1,24
2	1,47	1,29	1,26	1,24	1,24	1,24
3	1,59	1,34	1,29	1,26	1,24	1,24
4	1,68	1,37	1,32	1,29	1,25	1,24
5	1,75	1,40	1,35	1,30	1,26	1,25
6	1,82	1,43	1,37	1,32	1,27	1,68
8	1,91	1,47	1,40	1,34	1,28	1,71
10	1,98	1,49	1,42	1,36	1,30	2,28

Примітка. Світло-сірим кольором виділена частина таблиця, де містяться результати, що не приводять до збільшення порогового значення спектрального спотворення $SD_{\min}$ = 1,24. Дещо темнішим кольором показані результати, за яких SD збільшується, але незначно.

В табл. 2 показано кількість операцій за різних значень кількості кандидатів і похибки. Зауважимо, що під час повного пошуку за ЗЕМ необхідно здійснити 160000 операцій.

З урахуванням результатів, поданих в табл. 2, були вибрані ті, що вимагають мінімальної кількості операцій: при $|\mathbf{C}|$ = 10, e = 1, коли спектральне спотворення взагалі не збільшилось $\left(SD = SD_{\min}\right)$, та $|\mathbf{C}|$ = 5, e = 1 — результат з мінімальною кількістю необхідних операцій.

Таблиця 2

Кількість операцій

Похибка вибору найближчого вектора з КК, *e*	Оцінювання кількості операцій *N* при визначенні найближчого вектора, використовуючи					
	евклідову метрику	зважену евклідову метрику				
	1	5	10	20	50	100
0	3936	5674	7014	8956	13320	26982
1	3032	3750	4358	5424	8384	13974
2	2892	3570	4112	4866	7570	12334
3	2844	3312	3772	4660	7260	11742
4	2816	3258	3698	4558	7100	11466
5	2802	3226	3654	4496	7010	11314
6	2790	3204	3626	4460	6956	11220
8	2774	3180	3596	4418	6890	11108
10	2764	3164	3576	4398	6856	11054

На рис. 4 показано графіки залежності спектрального спотворення від кількості кандидатів та значення заданої похибки.

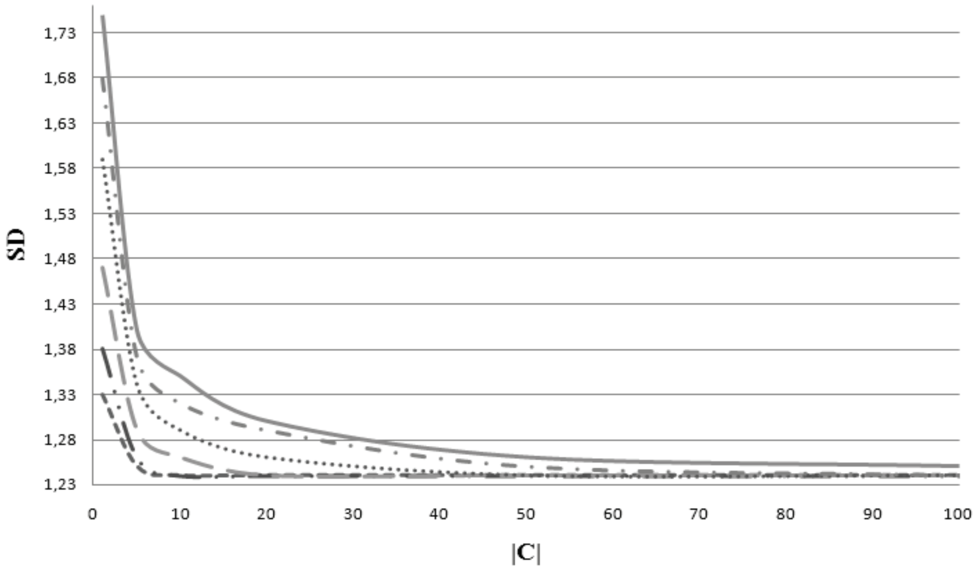

Рис. 4. Залежність спектрального спотворення від кількості кандидатів і похибки

Рис. 5 дозволяє проілюструвати, як змінюється кількість операцій в залежності від кількості кандидатів та похибки.

Отримані результати є очікуваними: зі збільшенням $|\mathbf{C}|$ спектральне спотворення зменшується, а кількість необхідних під час пошуку операцій зростає; зі збільшенням e — SD збільшується, N зменшується і навпаки.

Отже, показники, за якими здійснювалось оцінювання ефективності запропонованого підходу мають обернено пропорційну залежність. Результати, за яких досягається оптимум для SD і N, отримані при $|\mathbf{C}|$ = 5, *e* = 1 та $|\mathbf{C}|$ = 10, *e* = 1. При цьому кількість операцій зменшилась в 35 та 40 разів відповідно порівняно з повним пошуком, що дозволяє реалізувати систему ущільнення на DSP (Digital Signal Processor) з обмеженою швидкодією.

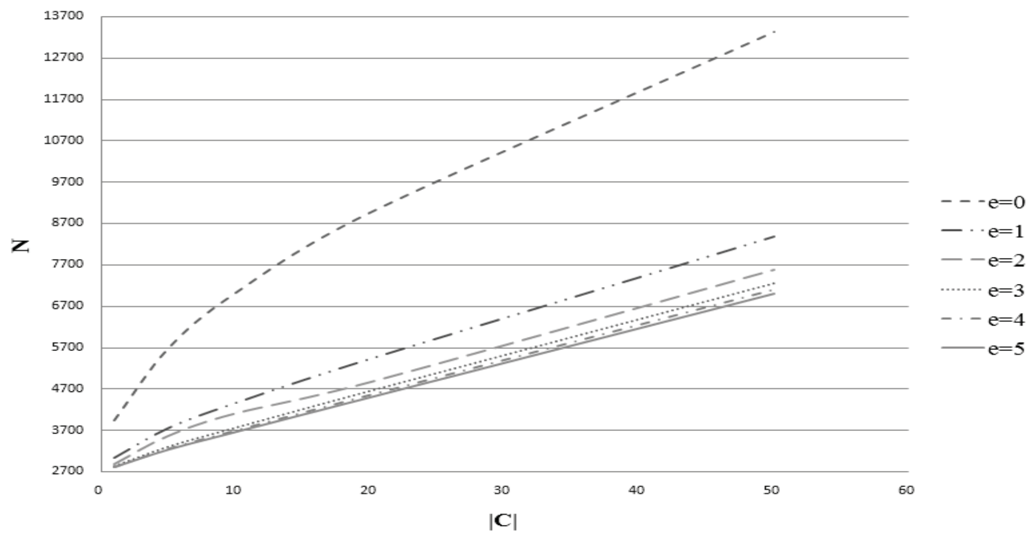

Рис. 5. Залежність кількості операцій від кількості кандидатів і похибки

Висновки

Розроблено двоетапну стратегію пошуку найближчого сусіда, яка полягає в тому, що (I) серед вибраних за звичайною евклідовою мірою найближчих до вхідного вектора $|\mathbf{C}|$ кандидатів, (II) знаходимо один (найближчий) за зваженою евклідовою мірою. Отримано аналітичні залежності для кількості кандидатів і кількості операцій під час пошуку найближчого вектора. Експериментальна перевірка показала, що результати, отримані для $|\mathbf{C}| = 10$, $e = 1$, дозволили досягти зменшення кількості операцій в 35 разів порівняно з повним перебором зі збереженням порогового значення $SD = 1{,}24$.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Chu W. C. Speech Coding Algorithms: Foundation and Evolution of Standardized Coders / Wai C. Chu // NY. : John Wiley & Sons, Inc. — 2003. — 558 p. — ISBN 0-471-37312-5.
2. Paliwal K. K. Efficient vector quantization of LPC parameters at 24 bits/frame / K. K. Paliwal, B. S. Atal // IEEE Transaction on Speech and Audio Processing. — 1993. — No. 2, vol. 1. — P. 3—14.
3. Gardner W. R. Theoretical analysis of the high-rate vector quantization of LPC parameters / W. R. Gardner, B. D. Rao // IEEE Transaction Speech Audio Processing. — Sep. 1995. — Vol. 3. — P. 367—381.
4. Hai Le Vu and Laszlo Lois. Efficient Distance Measure for Quantization of LSF and Its Karhunen–Loeve Transformed Parameters / Hai Le Vu and Laszlo Lois // IEEE Transactions on speech and audio processing. — Nov. 2000. — No. 6, vol. 8.
5. Zhou J. Simple Fast Vector Quantization of the Line Spectral Frequencies / J. Zhou, Y. Shoham, A. Akansu // Image Compression and Encryption Technologies. — 2001. — Vol. 4551. — P. 274—282.
6. Біліченко Н. О. Швидкий пошук при векторному квантуванні лінійних спектральних частот / Н. О. Біліченко, О. М. Ткаченко, О. Д. Феферман, С. В. Хрущак // Реєстрація, зберігання і обробка даних. — 2008. — Т. 10, № 2. — С. 37—47.
7. Ткаченко О. М. Спрямований пошук при квантуванні лінійних спектральних частот / О. М. Ткаченко, О. Ф. Грійо Тукало // Вісник Вінницького політехнічного інституту. — 2009. — № 5. — С. 64—69.
8. Bentley J. L. K-d trees for semidynamic point sets / J. L. Bentley // In Proc. 6th Ann. ACM Sympos. Comput. Geom. — 1990. — P. 187—197.
9. Friedman J. H. An algorithm for finding best matches in logarithmic expected time / J. H. Friedman, J. L. Bentley, and R. A. Finkel // ACM Transactions on Mathematical Software. — 1977. — 3(3). — P. 209–226.
10. Arya S. Algorithms for fast vector quantization / S. Arya and D. M. Mount // In J. A. Storer and M. Cohn, editors, Proc. of DCC '93: Data Compression Conference, IEEE Press. — 1993. — P. 381—390.
11. Approximate nearest neighbor queries in fixed dimensions / S. Arya and D. M. Mount // In Proc. 4th ACM-SIAM Sympos. Discrete Algorithms. — 1993. — P. 271—280.

Рекомендована кафедрою обчислювальної техніки

Стаття надійшла до редакції 8.06.11
Рекомендована до друку 21.06.11

Ткаченко Олександр Миколайович — доцент кафедри обчислювальної техніки;
Грійо Тукало Оксана Франсисківна — студентка Інституту інформаційних технологій та комп'ютерної інженерії.

Вінницький національний технічний університет, Вінниця