УДК 621.39

О.М. ТКАЧЕНКО, О.Ф. ГРІЙО ТУКАЛО

ПОШУК ВЕКТОРІВ У КОДОВИХ КНИГАХ ПРИ УЩІЛЬНЕННІ МОВЛЕННЯ НА ОСНОВІ БІНАРНОГО ДЕРЕВА

Анотація: Передача мовленнєвих сигналів у неущільненому вигляді потребує швидкісних каналів зв’язку через велику надлишковість даних. Саме тому в сучасних системах цифрового зв’язку широко застосовуються алгоритми кодування мовлення, що реалізують високу ступінь ущільнення при збереженні достатнього рівня якості звучання, зокрема з використанням кодових книг. При цьому широко застосовується векторне квантування. Проте практичне застосування векторного квантування у реальному масштабі часу обмежено через різке зростання витрат пам’яті та часу на пошук кодованого вектора у кодових книгах. Для скорочення часу пошуку найближчого вектора у кодових книгах запропоновано метод структуризації кодових книг на основі бінарного дерева. Показано, що використання бінарного дерева дозволяє суттєво зменшити кількість вимірювань відстані, необхідних для пошуку найближчого сусіднього вектора. Проаналізовано основні фактори, що впливають на ефективність пошуку. Розглянуто кілька варіантів розбиття області параметрів при створенні дерева, а саме: midpt, коли комірка ділиться січною площиною посередині перпендикулярно найдовшій стороні; sl_midpt, що відрізняється від попереднього можливістю зсуву січної площини з метою зменшення кількості тривіальних листів; sl_fair, що поєднує методи розбиття по медіані та sl_midpt з врахуванням обмеження на відношення сторін створюваних комірок. Наведено результати порівняльного аналізу вказаних варіантів. Розглянуто властивості LSF як об’єкта квантування, які необхідно враховувати при побудові бінарного дерева Експериментально досліджено продуктивність методів. Оскільки всі вектори в кодових книгах займають лише верхню частину прямокутної області, було запропоновано повернути осі координат на 45 градусів проти годинникової стрілки, перерахувавши координати векторів. Це дало можливість дещо скоротити час пошуку за рахунок штучного досягнення більшої симетричності розбиття дерева. Оцінювання результатів проводилося за кількістю обчислень відстаней у кодових книгах, необхідною для пошуку найближчого вектора, відносно розміру тестової вибірки. Застосування методу при квантуванні мовлення дозволило зменшити час пошуку найближчого вектора до 5% від часу повного пошуку. Скорочення часу пошуку (≈ в 17 разів) в процесі передавання мовленнєвих сигналів досягається за рахунок додаткових обчислювальних витрат на підготовчому етапі, необхідних для побудови бінарного дерева, а також збільшення обсягів пам’яті, потрібної для його зберігання.

Ключові слова: ущільнення мовленнєвих сигналів, лінійні спектральні частоти, кодові книги, бінарні дерева.

Аннотация: Передача речевых сигналов в несжатом виде требует скоростных каналов связи через большую избыточность данных. Именно поэтому в современных системах цифровой связи широко применяются алгоритмы кодирования речи, реализующих высокую степень уплотнения при сохранении достаточного уровня качества звучания, в частности с использованием кодовых книг. При этом широко применяется векторное квантование. Однако практическое применение векторного квантования в реальном масштабе времени ограничено из-за резкого роста затрат памяти и времени на поиск кодированного вектора в кодовых книгах. Для сокращения времени поиска ближайшего вектора в кодовых книгах предложен метод структуризации кодовых книг на основе бинарного дерева. Показано, что использование бинарного дерева позволяет существенно уменьшить количество измерений расстояния, необходимых для поиска ближайшего соседнего вектора. Проанализированы основные факторы, влияющие на эффективность поиска. Рассмотрено несколько вариантов разбиения области параметров при создании дерева, а именно: midpt, когда ячейка делится секущей плоскостью посередине перпендикулярно самой длинной стороне; sl_midpt, отличающийся от предыдущего возможностью смещения секущей плоскости с целью уменьшения количества тривиальных листов; sl_fair, сочетающий методы разбиения по медиане и sl_midpt с учетом ограничения на отношение сторон создаваемых ячеек. Приведены результаты сравнительного анализа указанных вариантов. Рассмотрены свойства LSF как объекта квантования, которые необходимо учитывать при построении бинарного дерева. Экспериментально исследовано производительность методов. Поскольку все векторы в кодовых книгах занимают лишь верхнюю часть прямоугольной области, было предложено повернуть оси координат на 45 градусов против часовой стрелки, перечислив координаты векторов. Это позволило несколько сократить время поиска за счет искусственного достижения большей симметричности разбиения дерева. Оценка результатов проводилось по количеству вычислений расстояний в кодовых книгах, необходимых для поиска ближайшего вектора, относительно размера тестовой выборки. Применение метода при квантовании речи позволило уменьшить время поиска ближайшего вектора до 5% от времени полного поиска. Сокращения времени поиска (≈ в 17 раз) в процессе передачи речевых сигналов достигается за счет дополнительных вычислительных затрат на подготовительном этапе, необходимых для построения бинарного дерева, а также увеличение объемов памяти, необходимой для его хранения.

Ключевые слова: сжатие речевых сигналов, линейные спектральные частоты, кодовые книги, бинарные деревья.

Abstract: Speech signals transmission without coding requires high-speed communication channels because of a large redundancy of data. That is why in modern digital communication systems are widely used speech coding algorithms that implement a high degree of coding while maintaining a sufficient level of sound quality, in particular including codebooks using. It is widely used vector quantization. However, the practical use of vector quantization in real time is limited due to the sharp increase in costs of memory and coded vector search time in the codebooks. To reduce the nearest vector search time in codebooks the method of structuring the codebook is proposed. Shown that using of the binary tree allows to reduce the distance measurements number significantly that are needed to find the nearest neighbor vector. The main factors influencing search performance are analyzed. Several parameters partition options are considered when creating the tree, namely: midpt, which cuts the current cell by the splitting plane through its midpoint orthogonal to its longest side; sl_midpt, that differs from the previous by "sliding" the splitting plane, if a trivial split were to result in order to avoid this; sl_fair, that combines split at the median with sl_midpt taking into account the aspect ratio bound. The results of comparative analysis of the listed methods are presented. Properties LSF as an object of quantization are considered when building the binary tree. The methods performance are studied experimentally. As all vectors in codebooks cover only the top of the rectangular area it was proposed to rotate the coordinate axis 45 degrees counterclockwise, recalculating the vectors coordinates. This enabled us to reduce the search time due to the achieving more symmetrical tree partitioning. Results evaluation was based on distances calculations number in the codebooks needed to find the nearest vector, among the sample test. The method using for speech quantization reduced the nearest vector search time to 5% of full-time search. Search time reducing (≈ 17 times) when the transmission of speech signals is achieved by additional computational costs of the preparatory steps needed for building a binary tree, and also increase the amount of memory needed to store.

Keywords: speech signal coding, line spectral frequency, codebooks, binary trees.

Вступ

Розвиток сучасних технологій цифрового зв'язку, таких як стільникова та IP-телефонія, потребує розробки ефективних методів і алгоритмів кодування мовлення, що реалізують високу ступінь ущільнення при збереженні достатнього рівня якості звучання. У теперішній час найвищий рівень ущільнення (найменшу швидкість передачі) забезпечують методи на основі лінійного прогнозування LPC (Linear Prediction Coding), які при безпосередній реалізації дозволяють зменшити швидкість передачі даних до 4800 біт/с. На практиці лінійні коефіцієнти прогнозування представляють у вигляді LSF (лінійних спектральних частот), що надає додаткові можливості підвищення рівня ущільнення мовленнєвого сигналу.

Для подальшого ущільнення використовуються квантування LSF за допомогою певних наборів значень, які називають кодовими книгами (КК). 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 із 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 із зростанням кількості компонентів, що квантується разом, тобто із збільшенням розмірності вектора. Проте разом з цим зростають витрати пам'яті, а також час на пошук кодованого вектора у кодовій книзі, що зумовлює суттєві обмеження на практичне застосування ВК у реальному масштабі часу. Тому в діючих стандартах застосовують субоптимальне кодування, коли 10-вимірний вектор параметрів розбивають на 2 або більше підвекторів меншої розмірності [1]. Це дозволяє зменшити швидкість передачі даних до 2400 біт/с.

З метою зменшення часу пошуку в [2] було запропоновано декілька підходів до впорядкування векторів у кодовій книзі, названих авторами методами швидкого векторного квантування (fast vector quantization methods). Було показано, що складність обчислень при застосуванні наведених методів складає порядка 25% від складності обчислень при повному пошуку без суттєвої втрати продуктивності, що оцінювалася по спектральному спотворенню.

В [3] було розроблено метод структуризації КК на основі відношення мажорування. Це дозволило зменшити складність обчислень до 16-20% від відповідної складності при повному пошуку. У даній статті пропонується метод пошуку по бінарному дереву, вперше введений при кодуванні зображень [4]. Застосування метода при квантуванні мовлення дозволяє зменшити час пошуку найближчого вектора до 5% від часу повного пошуку.

Мета та задачі статті

Метою роботи є зменшення складності обчислень при кодуванні мовленнєвих сигналів за рахунок пошуку найближчого вектора у кодовій книзі по бінарному дереву.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- розробити модель і структуру даних, що зберігаються у КК;
- розробити метод пошуку найближчого вектора у створеній КК;
- оцінити ефективність розробленого методу.

Розробка математичної моделі та метода пошуку по бінарному дереву

Нехай КК містить кінцеву множину векторів $Q=\{Y_1,Y_2,\ldots,Y_N\}$, $Y_i=(y_{i1},y_{i2},\ldots,y_{iM})$. Таким чином, з кожним вектором Y_j у КК пов'язаний індекс, або кодове слово j, що може бути записано як N- розрядне ціле число. На вхід квантизатора поступає вектор $X=(x_1,x_2,\ldots,x_M)$. В результаті кодування необхідно вибрати таке кодове слово j, що мінімізує спотворення $d(X,Y_j)$ (правило вибору найближчого сусіднього вектора).

$$R_j=\{X: d(X,Y_j)\le d(X,Y_i);\ \forall i\in I, i\ne j\}, \tag{1}$$

де $I=\{1,2,\ldots,N\}$ - множина індексів.

У даній роботі для вимірювання спотворення використовувалася Евклідова відстань:

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

Ациклічний зв'язний граф $T=(V,E)$, де $V=\{v_1,v_2,\ldots,v_n\}$ - множина вершин, $E=\{e_1,e_2,\ldots,e_m\}$ - множина пар елементів множини V, або множина ребер ($m=n-1$), називається

деревом. Якщо з деякої вершини v_i веде ребро у вершину v_j ($v_i, v_j \in V$), то v_i називається предком v_j, а v_j - нащадком v_i.

Вершини дерева поділяють на такі три групи:

- корінь – одна виокремлена вершина, в яку не входить жодного ребра, тобто вона не має предків;
- вузли – вершини, в які входить одне ребро і виходить одне чи більше ребер;
- листи або кінцеві (термінальні) вершини – вершини, в які входить одне ребро і не виходить жодного ребра (не мають нащадків).

Для $\forall v_i \in V$, підграф дерева $T=(V,E)$, що включає всі досяжні з v_i вершини і з'єднуючі їх ребра з E, утворює піддерево $T_{v_i} \in T$ з коренем v_i. З цього випливає, що кожна вершина є в свою чергу коренем деякого піддерева. Тобто піддерево – це $\forall v_i \in V$ з усіма її нащадками. Якщо існує відносний порядок на піддеревах $T = \{T_1, T_2 ..., T_N\}$, то таке дерево називається впорядкованим. Висотою дерева називається максимальна із довжин усіх можливих шляхів від кореня дерева до термінальних вершин $H \geq h_k; k \in L$ (L - кількість листів).

Важливим окремим випадком кореневих дерев є бінарні дерева (БД), які визначаються як множина вершин, яка має виокремлений корінь та два піддерева (праве та ліве), що не перетинаються, або є пустою множиною вершин (на відміну від звичайного дерева, яке не може бути пустим): $T \leq \{T_1, T_2\}; T_1 \not\subset T_2$. Таким чином, бінарне дерево — це структура даних, в якій кожна вершина має не більше двох нащадків (правого та лівого).

Бінарне дерево пошуку (англ. binary search tree, BST) — БД, в якому кожній вершині $v_i \in V$ співставлене певне значення $Val[v_i]$. При цьому такі значення повинні задовольняти умові впорядкованості: нехай v_i — довільна вершина БД пошуку. Якщо вершина v_j знаходиться в лівому піддереві вершини v_i, то $Val[v_j] \leq Val[v_i]$.

Якщо v_j знаходиться у правому піддереві x, то $Val[v_j] \geq Val[v_i]$. Використання БД дозволяє суттєво зменшити кількість вимірювань відстані K, необхідних для пошуку найближчого сусіднього вектора ($K \approx log_2 N$ замість $K = N$ при повному пошуку, де N - кількість векторів у КК). Проте при цьому зростає спектральне спотворення, оскільки пошук по БД не гарантує знаходження дійсно найближчого вектора відповідно до формули (1) та рисунку 1.

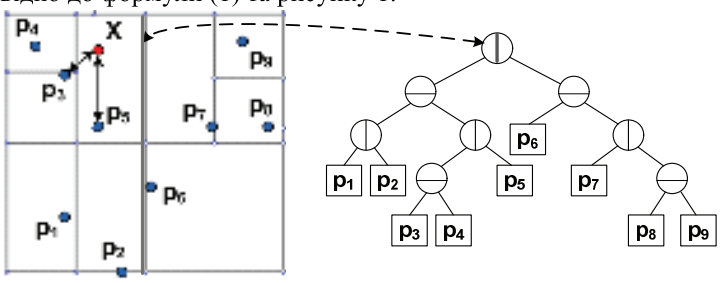


Рисунок 1. Ілюстрація впорядкування КК на основі бінарного дерева

У роботі [5, 6] запропоновано вдосконалену процедуру пошуку по БД. Як вище було зазначено, в БД (крім кореневої) присутні два типи вершин: термінальні (листи) та нетермінальні (вузли). Кожен лист дерева може містити не більше одного вектора КК, тобто $C_K \subset Y_i$ або $C_K \subset \varnothing$, де C_K - деякий лист БД (тобто в нашому випадку кількість листів $k \geq 4096$).

Розрізняють дві фази пошуку - пряму та зворотню. 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 зворотня 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 для двовимірного випадку наведено на рисунку 2. Розглянута процедура гарантує знаходження найближчого сусіднього вектора згідно (1), проте потребує значно більшої кількості вимірювань відстані, ніж $K \approx log_2 N$.

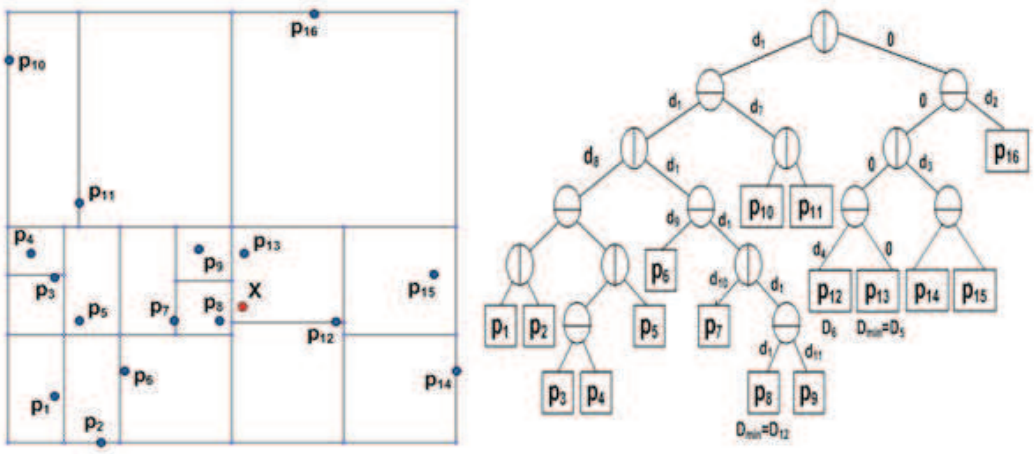


Рисунок 2. Ілюстрація пошуку найближчого вектора по бінарному дереву

З метою скорочення часу в [4] запропоновано для обчислення відстані до вузлів скористатися більш ефективною формулою

$$D_k{}^2 = (D_k{}^2)' - (y_k' - x_k)^2 + (y_k - x_k)^2, \quad (3)$$

де $(D_k{}^2)'$ - раніше обчислена відстань до певного вузла дерева. Застосування (3) можливе завдяки тому, що на кожному кроці просування по дереву змінюється лише одна з координат. Обчислення відстані згідно (3) потребує лише одного множення та трьох додавань, оскільки $(y_k' - x_k)^2$ можна зберігати у структурі даних дерева. Обчислення відстані до листів доводиться виконувати за формулою (2).

Застосування БД при кодуванні мовлення

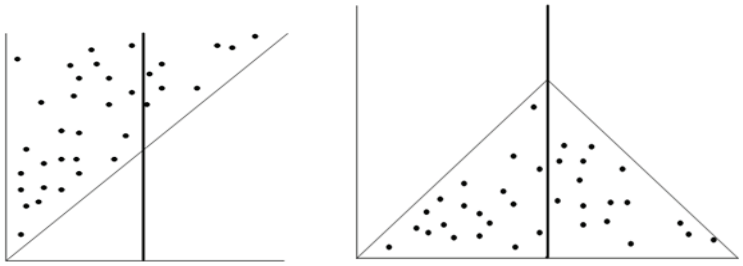
Розглядалися різні варіанти побудови дерева, застосування яких призводить до різної кількості обчислень відстані під час пошуку:

- midpt - комірка ділиться січною площиною посередині перпендикулярно найдовшій стороні. Відношення сторін є обмеженим. Хоча цей метод простий, його неефективно використовувати для великих розмірностей, оскільки він може генерувати велику кількість тривіальних листів (тобто, які не містять векторів КК), а також незбалансовані розбиття;
- sl_midpt - працює як і попередній, якщо по обидві сторони січної площини є точки. Інакше січну площину рухають, поки вона не досягне деякого вектора КК, що забезпечує відсутність тривіальних листів і відповідно БД має менше вузлів;
- sl_fair - різновид останнього; поєднує методи розбиття по медіані та sl_midpt, враховується обмеження на відношення сторін створюваних комірок, забезпечує більш збалансоване розбиття.

Найкращим варіантом дерева слід вважати такий, коли всі комірки мають кубічну форму (січною площиною ділиться найбільша сторона комірки) з кодовим вектором точно всередині. Проте таке дерево містить багато тривіальних листів і має занадто великий розмір. Тим не менш як критерій для обрання того чи іншого варіанту побудови дерева можна використати співвідношення максимальної та мінімальної сторін комірки або максимальну відстань між двома точками комірки.

Властивості LSF як об’єкта квантування необхідно враховувати при побудові БД. Оскільки для будь-якого вектора LSF параметрів виконується $y_{ik} < y_{ik+1}\ \forall i \in I$, всі вектори в КК займають лише верхню частину прямокутної області (рисунок 3, а). Враховуючи це, було запропоновано повернути осі координат на 45 градусів проти годинникової стрілки, перерахувавши координати векторів за формулою 4 (рисунок 3, б). Це дало можливість дещо скоротити час пошуку за рахунок штучного досягнення більшої симетричності розбиття дерева (проте і в цьому випадку правильним можна вважати лише початкове розбиття).

$$\begin{aligned} y_{1k}' &= y_{1k} \cdot \cos 45 + y_{2k}' \cdot \sin 45 = \frac{\sqrt{2}}{2}(y_{1k}' + y_{2k}'); \\ y_{2k}' &= y_{2k} \cdot \cos 45 - y_{1k}' \cdot \sin 45 = \frac{\sqrt{2}}{2}(y_{2k}' - y_{1k}'). \end{aligned} \quad (4)$$



а) б)

Рисунок 3. Ілюстрація специфіки розміщення векторів в КК в 2-мірному просторі:

а) до повороту; б) після повороту

Експериментальне дослідження метода пошуку по бінарному дереву

Для побудови бінарного дерева та пошуку найближчого вектора було розроблено програмне забезпечення у середовищі Visual Studio 2008. 10-вимірний вектор LSF-параметрів було розбито на дві частини розмірністю $M = 5$, розмір кожної частини КК складав $N = 4096$ підвекторів. Таким чином, обсяг даних, необхідних для представлення короткочасної спектральної інформації, становив 24 біти. Для тестування було відібрано текст (уривок з роману І. Нечуй-Левицького), розмір тестової вибірки складав $V = 20000$ векторів.

У таблиці 1 наведено основні характеристики отриманого БД для кожного метода.

Таблиця 1 – Характеристики БД для КК розміром $M = 5, \quad N = 4096$

Метод	Відношення сторін листа	Глибина БД	К-сть листів	К-сть вузлів	К-сть тривіальних листів
Midpt	1.74445	50	7308	7307	3212
sl_midpt	2.26011	23	4096	4095	0
sl_fair	3.00671	18	4096	4095	0

Оцінювання результатів проводилося за кількістю обчислень відстаней (КОВ) у КК C, необхідною для пошуку найближчого вектора, відносно розміру тестової вибірки:

$$КОВ = \frac{C}{V}. \qquad (5)$$

Якщо для оцінювання складності обчислень орієнтуватися на значення середньої, а не максимальної КОВ, можна значно зменшити час пошуку (в 3-4 рази). Проте середня КОВ не є інформативною характеристикою для ущільнення мовленнєвих сигналів у реальному часі, оскільки час, заощаджений на обчислення в одному фреймі, неможливо використати в іншому, тому у таблицях наведено значення максимальної КОВ.

Отримані результати при використанні різних методів побудови дерева до і після повороту осі координат на 45 градусів проти годинникової стрілки наведено в таблиці 2 та на рисунку 4.

Таблиця 2 – КОВ для різних методів побудови дерева

Метод	КОВ									
	Середня				Максимальна (% від повного пошуку)					
	До повороту		Після повороту		До повороту			Після повороту		
	до листів	До вузлів	до листів	до вузлів	до листів	до вузлів	сумарна	до листів	до вузлів	сумарна
midpt	28	114	27	115	111 (2.7%)	422 (2,27%)	4,97%	113 (2.75%)	457 (2,46%)	5,21%
sl_midpt	37	91	36	85	153 (3.7%)	329 (1,77%)	5,47%	146 (3.6%)	295 (1,58%)	5,18%
sl_fair	39	95	36	85	162 (3.95%)	333 (1,79%)	5,74%	131 (3.2%)	316 (1,71)	4,91%

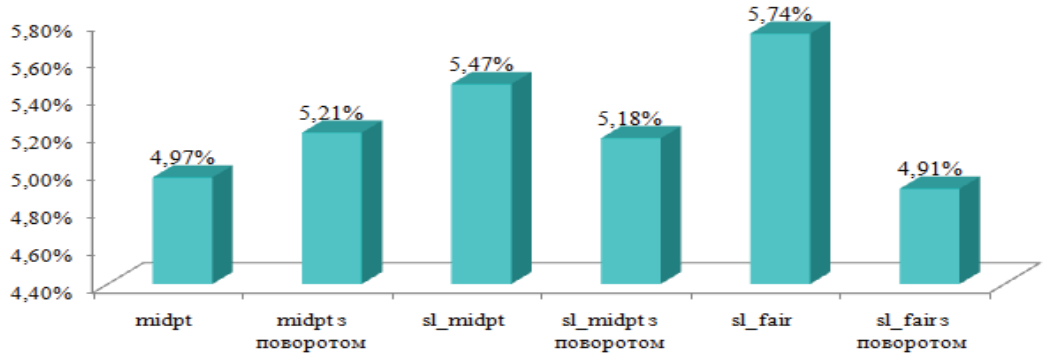


Рисунок 4. Максимальна КОВ для різних методів побудови дерева

Отримані експериментальні дані, наведені у таблиці 2, свідчать, що найпростіший метод побудови дерева midpt після повороту потребує найбільшої сумарної КОВ. Це пояснюється тим, що кількість вузлів в 1,8 разів більша порівняно з двома іншими методами (див. таблиця 1), незважаючи на те, що при обчисленні відносних значень КОВ до вузлів кількість необхідних операцій зменшується в 4,5 рази.

Висновки

Складність обчислень при пошуку найближчого вектора у КК із застосуванням бінарного дерева склала 5 – 6% від складності обчислень при повному пошуку. Витрати пам'яті при цьому в середньому збільшилися в 3 рази.

Таким чином, скорочення часу пошуку (≈ в 17 разів) в процесі передавання мовленнєвих сигналів досягається за рахунок додаткових обчислювальних витрат на підготовчому етапі, необхідних для побудови бінарного дерева, а також збільшення обсягів пам'яті, потрібної для його зберігання.

Література

1. Paliwal K. K., Atal B. S. Efficient vector quantization of LPC parameters at 24 bits/frame — IEEE Transaction on Speech and Audio Processing, 1993.—No. 2, vol. 1.—P. 3—14.
2. Zhou J., Shoham Y., Akansu A. Simple Fast Vector Quantization of the Line Spectral Frequencies // Image Compression and Encryption Technologies. — 2001. — Vol. 4551. — P. 274—282.
3. Біліченко Н.О., Ткаченко О. М., Феферман О. Д., Хрущак С. В. Швидкий пошук при векторному квантуванні лінійних спектральних частот // Реєстрація, зберігання і обробка даних. — 2008. — №2, т. 10. — С. 37—47.
4. S. Arya and D. M. Mount. Algorithms for fast vector quantization. In J. A. Storer and M. Cohn, editors, Proc. of DCC '93: Data Compression Conference, pages 381 — 390. IEEE Press, 1993.
5. J. L. Bentley. K-d trees for semidynamic point sets. In Proc. 6th Ann. ACM Sympos. Comput. Geom., pages 187 — 197, 1990.
6. J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. ACM Transactions on Mathematical Software, 3(3):209 — 226, 1977.

Відомості про авторів

Ткаченко Олександр Миколайович – к.т.н., доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, Хмельницьке шосе, 95, м. Вінниця, 21021, тел. 59-84-13, alextk1960@gmail.com.

Грійо Тукало Оксана Франсисківна – студентка кафедри обчислювальної техніки Вінницького національного технічного університету, Хмельницьке шосе, 95, м. Вінниця, 21021, xxmargoxx@gmail.com.