

УДК 794:004+004.627+004.415+621.391

**Горєлов Віталій Олевтинович**

Кандидат технічних наук, доцент кафедри комп'ютерних наук та інформаційних систем,  
*orcid.org/0000-0002-2106-8704*

ДВНЗ «Прикарпатський національний університет імені Василя Стефаника», Івано-Франківськ

**Іляш Юрій Юрійович**

Кандидат технічних наук, доцент кафедри комп'ютерних наук та інформаційних систем,  
*orcid.org/0000-0003-3724-4183*

ДВНЗ «Прикарпатський національний університет імені Василя Стефаника», Івано-Франківськ

**Ровінський Віктор Анатолійович**

Кандидат технічних наук, доцент кафедри комп'ютерних наук та інформаційних систем,  
*orcid.org/0000-0001-8454-8580*

ДВНЗ «Прикарпатський національний університет імені Василя Стефаника», Івано-Франківськ

**МЕТОДИ УЩІЛЬНЕННЯ ДАНИХ ТА ПЕРЕТВОРЕННЯ ФОРМИ ІНФОРМАЦІЇ  
У КОМП'ЮТЕРНИХ ІГРАХ**

***Анотація.** Розглянуто питання, пов'язані з ущільненням даних та перетворенням форми інформації при розробленні комп'ютерних ігор. Показано методологічний підхід, що дає змогу ефективно реалізувати зменшення об'єму аудіоданих у порівнянні з використанням звичайних музичних файлів зі стисненням. Запропоновано метод та проведено експериментальні дослідження, результати яких можуть бути використані для розроблення ігрового програмного забезпечення. Визначення ефективності методів компресії важливе для подальшого вибору конкретного методу. Розглянуто систему числових характеристик, які дають змогу порівнювати методи між собою, запропоновано комплексний підхід до оцінки коефіцієнта стиснення та показник, який є об'єктивною кількісною характеристикою будь-якого квазізворотного методу стиснення даних. Розглянуто метод перетворення форми інформації при дослідженні подібності ламаних ліній, заданих набором пар координат. Запропоновано перетворення числових значень координат на площині у символічні рядки, які можуть оброблятися засобами систем управління базами даних, що допомагає отримати високу швидкість і усуває потребу проведення арифметичних обчислень у реальному часі.*

***Ключові слова:** звуковий файл; компресія; ефективність компресії; перетворення форми інформації; тайл; СУБД*

**Методологія зменшення об'єму  
аудіоданих у комп'ютерних іграх**

Звуковий супровід є важливою складовою комп'ютерних ігор, якій приділяють значну увагу [1-5]. Створення програмних блоків, призначених для відтворення звукового та музичного забезпечення в ігрових програмах, потребує дотримання певних вимог, серед яких основними є: робота в реальному часі та економія пам'яті обчислювального пристрою. Робота в реальному часі забезпечує паралельне відтворення звукових сигналів без пауз, які перевищують частоту дискретизації відтворюваного сигналу, під час проведення ігрового процесу. У переважній більшості випадків режим реального часу задовільно забезпечується засобами ігрових фреймворків, які використовують при розробленні ігор.

Економія пам'яті є особливо важливим завданням при виробництві ігор, розрахованих на використання в мобільних пристроях, оскільки такі пристрої містять меншу кількість оперативної та програмної пам'яті у порівнянні із ПК.

Просте використання фреймворків не забезпечує автоматичного ущільнення даних, потрібних для збереження ігрового музичного супроводу, що призводить до надмірного використання пам'яті обчислювального пристрою. Тому актуальним завданням є розроблення методології алгоритмічного стиснення звукових даних. Збереження звукових сигналів у пам'яті даних обчислювальних пристроїв, як правило, здійснюється у вигляді стиснених звукових файлів із втратами (.mp3, .ogg/Vorbis), які забезпечують прийнятну якість звукового відтворення. Типово, це забезпечує коефіцієнт стиснення форматних даних:

$$k_f = \frac{S_0}{S_c} \approx 10,$$

де  $S_0$  – об’єм початкових даних;  $S_c$  – об’єм скомпресованих даних у порівнянні із нестисненими форматами (.wav, .aif). Використання безвратного формату звукових даних (.flac), не є рекомендованим для ігрової індустрії, оскільки забезпечує форматний коефіцієнт стиснення даних тільки на рівні  $k_f \approx 2$ , а забезпечення покращення якості звуковідтворення не буде помітне. Використовуючи деякі методологічні прийоми, можна досягнути значно більшого зменшення об’єму використання даних, ніж просте використання музичних файлів зі стисненням. Загальне стиснення даних у такому разі визначається так:

$$k = k_f \cdot k_m,$$

де  $k_m$  – методологічний коефіцієнт стиснення даних.

Першим методологічним способом, який можна застосувати для зменшення об’єму збережених даних, є використання збережених MIDI-команд, які використовуються для керування музичними інструментами. У такому разі ігрова програма повинна мати вбудовані музичні синтезатори, семплери, звукові ефекти тощо. Проведені авторами дослідження показують, що коефіцієнт стиснення даних при цьому потрапляє в межі  $k_m \in [870...21340]$  на вибірці з 117 різножанрових музичних проектів. Проте такий спосіб стиснення має суттєві недоліки – низьку якість генерованого звуку, який викликає ілюзію «механістичного сприйняття», та суттєве зростання навантаження на центральний процесор, яке пропорційне до покращення звукових характеристик генерованого сигналу. Це пов’язане з тим, що ігрова програма має в реальному часі синтезувати різноміркові звуки у відповідь на надходження MIDI-команд «note-on», що вимагає значної кількості математичних обчислень.

Такі операції виконуються спеціалізованими музичними синтезаторами або потужними персональними комп’ютерами, і це може займати більше 80% їхніх обчислювальних потужностей.

Виходячи з цього, такий спосіб може бути рекомендовано тільки, якщо генерована музична композиція є дуже простою і це узгоджується із художнім задумом гри. Іншою методою, яка може бути застосована для зменшення використаного об’єму пам’яті, є врахування особливостей структури музичного матеріалу, який здебільшого має значну кількість циклічно повторюваних фрагментів. Для прикладу розглянемо типовий музичний проект, що має рефренну модель побудови і наведений на рис. 1.

Проект складається з 35 партій музичних інструментів, які представлені в часі у вигляді горизонтальних доріжок (Piano, Electric Piano Suitcase, тощо). Проект має спільні фрагменти, які повторюються, тому їх доцільно записати у вигляді окремих аудіофайлів і відтворювати декілька разів протягом всього часу звучання композиції. Таким чином може бути забезпечене додаткове стиснення музичних даних. Більшість партій музичних інструментів також містять значну кількість повторів у часі, як наведено для прикладу на рис. 2. Формування аудіофрагментів здійснюється після процесу створення музичної композиції у аудіоредакторі (Apple Logic Pro X, Steinberg Cubase, тощо). Практично визначено, що кількість фрагментів для поширених ігрових музичних композицій наближено варіюється від 4 до 32. Тому доцільно створювати музичні композиції рефренного типу, які містять багато циклічних фрагментів. Отже, програмна модель блоку звуковідтворення для ігрової платформи може мати вигляд, показаний на рис. 3. Генератор опорних відліків формує тактові проміжки на основі підпрограми формування сигналу частоти зміни кадрів, яка складає переважно 60 кадрів/сек.

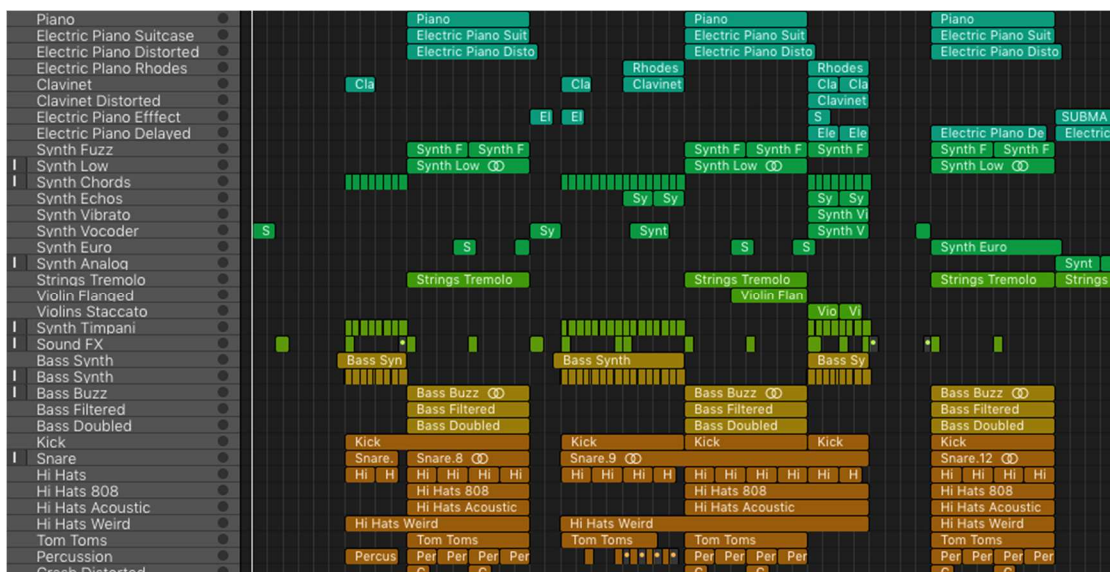


Рисунок 1 – Типовий музичний проект для Apple Logic Pro



Рисунок 2 – Типова музична партія ігрового проекту

Тактові відліки надходять до списків стартових позицій  $i$ , у випадку збігу номера поточного такту з тактом запуску заданого  $i$ -го аудіофрагмента, на вхід аудіоблока «Аудіо №  $i$ » подаються команди старту і амплітуди. Блок аудіовідтворення у відповідь на ці сигнали програв свій фрагмент із заданою амплітудою. При надходженні наступного запускомого сигналу, аудіофрагмент може бути заново відтворений, причому між циклічно повторюваними фрагментами пауза відсутня. Вихідні сигнали з багатьох блоків аудіовідтворення сумуються у вихідному суматорі і подаються на вхід додаткового аудіокомпресора, який вирівнює можливі амплітудні викиди вихідного сигналу. У найпростішому випадку – компресор може бути відсутній.

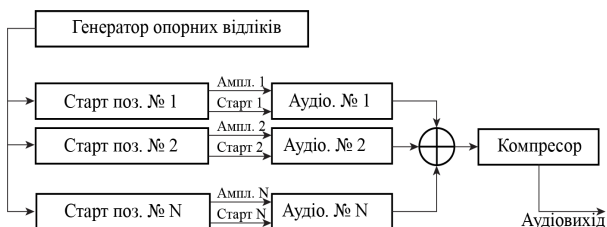


Рисунок 3 – Блок звуковідтворення музичної композиції

Формат файлу із втратами для запису аудіофрагментів доцільно вибирати .ogg/Vorbis або .ogg/Opus, оскільки він допускає вище стиснення даних ніж .mp3 при співставній якості. Крім того, зменшення бітрейту для .ogg сприймається акустично як спецефект, а не як погіршення якості, що притаманно для кодерів .mp3. Тому використання .mp3-стиснення доцільно тільки, якщо ігровий фреймворк не підтримує формат .ogg/Vorbis, як це є наприклад в SpriteKit для iOS. Іншою перевагою формату .ogg/Vorbis є відсутність зміщень в дискретизаціях, при варіативній дискретизації сигналу, та відсутність патентних обмежень – формат є вільним для використання.

Авторами експериментально визначено, що забезпечений метод для випадку 24 аудіофрагментів загальною довжиною 4 хв 28 с з використанням кодека .ogg/Vorbis дає змогу забезпечувати неперервне звучання музичної композиції тривалістю 58 хв. 20 с без суттєвого прояву ефекту одноманітності при прослуховуванні. Для даного випадку загальний коефіцієнт стиснення інформації при частоті дискретизації 44100 Гц буде становити  $k \approx 130,6$ , що доводить високу ефективність запропонованої методи.

### Визначення ефективності зменшення об'єму даних

Визначення ефективності зменшення об'єму даних в ігровій індустрії є одним з основних етапів розроблення ігрового програмного забезпечення на етапі його кінцевого тестування. В основному компресія даних застосовується до музичного супроводу відеогри. При компресії даних важливим етапом є визначити найефективніший метод, який можна застосувати в конкретному випадку.

Визначення ефективності зменшення об'єму даних можна проводити двома шляхами. По-перше, це можна зробити аналітично для відомої математичної моделі аудіофайлу; по-друге, – експериментально, застосувавши методи до конкретних аудіоданих. Кожний з цих шляхів має свої недоліки і переваги. Експериментальне дослідження ефективності методу стиснення здійснити простіше, але при цьому можна отримати тільки результат для даної реалізації процесу і нічого конкретного не можна сказати про результат повторного випробування. Знання математичної моделі, вхідного процесу дає можливість провести теоретичний аналіз вибраного методу і одержати вичерпні характеристики [6 – 8].

У роботах [9; 10] обґрунтовані деякі критерії оцінки, але кожний з критеріїв не дає змогу охарактеризувати повною мірою всі характеристики

методів стиснення даних. Для порівняння методів компресії між собою необхідно встановити систему єдиних числових характеристик. Такими характеристиками можуть бути:

- 1) коефіцієнт стиснення;
- 2) показник ефективності представлення повідомлень;
- 3) коефіцієнт складності;
- 4) час затримки повідомлення в кодері джерела;
- 5) показник завадостійкості;
- 6) порівняння з еталонною процедурою кодування;
- 7) складність алгоритму декодування, час декодування.

Ефективність процедури оброблення даних можна оцінити шляхом порівняння із заданою еталонною процедурою при заданій похибці відновлення аудіоряду.

Як еталон можна вибрати:

1) оптимальний метод кодування, який дає змогу реалізувати потенційні характеристики системи обробки;

2) стандартний метод, за якого дані рівномірно дискретизують в часі і в кожному дискреті зіставляють кодове слово фіксованої довжини.

У першому випадкові можна оцінити якість алгоритму по відношенню до максимального ефекту, який визначається ентропією джерела. Для цього як оптимальну процедуру слід використовувати універсально-оптимальний або адаптивний кодер з обмеженою затримкою.

У другому випадкові коефіцієнт стиснення має характеризувати зміну об'єму звукового файлу, оскільки компресія використовується для швидшого завантаження звукового ряду в оперативну пам'ять електронного пристрою. Природно, що оцінку коефіцієнта стиснення доцільно проводити шляхом порівняння об'єму повідомлень, отриманого в результаті компресії, тобто при вибраному способі представлення (кодування) з об'ємом, що отримується при деякому початковому формуванні даних, тобто

$$K_1 = \frac{J_0}{J}, \quad (1)$$

де  $J_0$  – об'єм даних до компресії;  $J$  – об'єм даних після компресії.

Коефіцієнт  $K_1$  задовольняє основні вимоги, які висуваються до показника якості компресії; він залежить від вибраного методу та порівняно просто обчислюється на основі математичної моделі.

У випадку, коли в процесі перетворення та обробки даних відбувається дискретизація і кількість рівнів квантування на кожний інформативний елемент (дискрет) стала і дорівнює  $A$ , коефіцієнт стиснення визначається:

$$K_2 = \frac{J_0}{J_\alpha} = \frac{N' \cdot \log_2 A}{N \cdot \log_2 A} = \frac{N'}{N}, \quad (2)$$

де  $N'$  – загальна кількість дискретів;  $N$  – кількість дискретів після застосування методу.

Досить часто цей коефіцієнт називають коефіцієнтом за кількістю відліків (рис. 4).

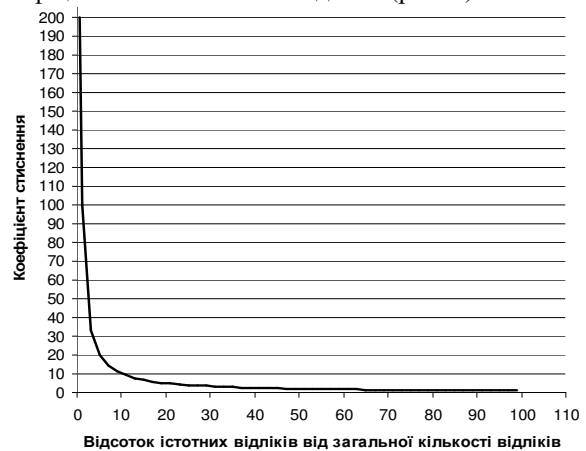


Рисунок 4 – Залежність коефіцієнта  $K_2$  від кількості дискретів після стиснення

Більш точну оцінку ефективності можна отримати, якщо  $J = J_\alpha + J_\beta$ , тобто враховується наявність службової інформації  $J_\beta$ . В цьому випадку коефіцієнт стиснення  $K_3$ , який називається коефіцієнтом стиснення за кількістю двійкових одиниць, визначається так:

$$K_3 = \frac{J_0}{J_\alpha + J_\beta} = \frac{J_0}{J_\alpha \left(1 + \frac{J_\beta}{J_\alpha}\right)} = \frac{K_2}{1 + \alpha}, \quad (2)$$

де  $J_\alpha$  та  $J_\beta$  – кількість двійкових одиниць (біт), які представляють інформаційну та службову частини цифрових даних, що передаються:

$$\alpha = \frac{J_\beta}{J_\alpha} \leq 1. \quad (3)$$

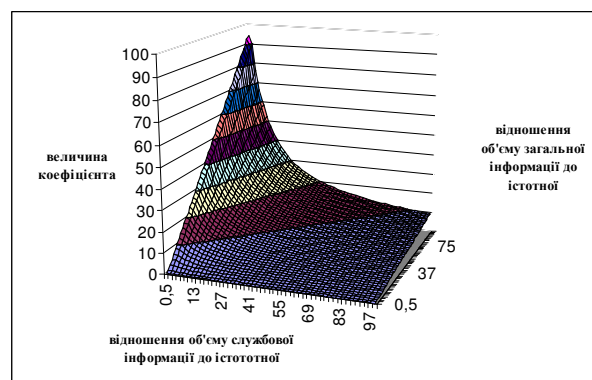


Рисунок 5 – Залежність коефіцієнта  $K_3$  від об'ємів службової та істотної інформації



Слід відзначити, що якщо  $K_2 \geq 1$ , то  $K_3$  може бути не менше одиниці, тобто можливий програш по  $K_3$  при малих  $K_2$ .

Використовуючи для порівняння в якості еталона рівномірно-часову циклічну дискретизацію, можна отримати ще один варіант оцінки коефіцієнта стиснення даних, а саме, по смузі частот

$$K_4 = F_0 / F_{CT}, \quad (4)$$

де  $F_0$  і  $F_{CT}$  – відповідно необхідна смуга частот до і після стиснення інформації.

Цей варіант оцінювання коефіцієнта стиснення достатньо точно характеризує ефективність конкретного алгоритму компресії. Це пояснюється тим, що він враховує заповненість смуги частот (отже, і споживаної потужності) та передачу стиснутих повідомлень в цілому, тобто корисну, часову, службову і корегуючу інформації.

При виборі одного з групи методів стиснення даних, рівноцінних з точки зору ефективності та інших показників, перевагу отримує менш складний метод. Складність системи може бути визначена як середня кількість обчислювальних операцій та об'єм пам'яті, необхідний для визначення однієї інформативної вибірки; як кількість команд в обчислювальній програмі, яка моделює роботу системи; як середня кількість елементів функціонально повної системи, яка дає змогу реалізувати систему оброблення даних іншими способами.

Для визначення ефективності того чи іншого методу стиснення даних можна використовувати комплексний коефіцієнт ефективності, який можна визначати як добуток або суму факторів, що впливають на ефективність методу стиснення даних [11]:

$$K_{ef1} = \prod_{i=1}^n t_i, \quad (5)$$

$$K_{ef2} = \sum_{i=1}^n t_i, \quad (6)$$

де  $n$  – кількість факторів;  $t_1 - t_3$  – апаратна, алгоритмічна, часова складність;  $t_4$  – коефіцієнт стиснення;  $t_5$  – складність алгоритму декодування, час декодування;  $t_6$  – чутливість до помилок;  $t_7$  – коефіцієнт розмноження помилок.

Коефіцієнт стиснення залежить від складності алгоритму обробки, і, як правило, чим вище необхідний коефіцієнт стиснення, тим складніше алгоритм обробки.

Як коефіцієнт складності раціонально прийняти середнє число обчислювальних операцій, що реалізується кодером стиснення для обчислення

однієї координати, при цьому обчислювальні операції повинні бути зведені до коротких операцій.

За показники складності апаратної реалізації алгоритмів стиснення даних визначаються:

- середнє або максимальнє число обчислювальних операцій, які витрачаються для обробки одного параметра;
- кількість програм в машинному кодї алгоритму;
- об'єм оперативної і постійної пам'яті, необхідної для реалізації алгоритму на ЕОМ;
- середнє число стандартних елементів, які використовуються для створення спеціалізованого блоку стиснення інформації.

Складність системи, яка складається з елементів складності  $S_i$ ,  $i = \overline{1, n}$ , розраховується згідно виразу:

$$S_v = (1 + \nu \gamma_c) \sum_{i=1}^n S_i k_i, \quad (7)$$

де  $\gamma_c$  – кількість функціональних зв'язків системи;  $k_i$  – кількість елементів  $i$ -го типу, що входять до складу системи;  $\nu$  – ваговий коефіцієнт, який враховує складність зв'язків у порівнянні з складністю елементів системи.

Для оцінювання ефективності компресії традиційно розраховують ентропію даних до застосування методів зменшення надлишковості  $H(x)$ , ентропію після застосування алгоритмів компресії  $H_{CT}(x)$  і максимально можливу середню кількість інформації  $H_{max}(x)$ . Тоді визначити ефективність стиснення даних доцільно не порівнянням  $H_{CT}(x)$  і  $H(x)$ , як це по суті робиться в попередніх оцінках (наприклад, відношення кількості відліків у процесі, що вимірюється, до і після стиснення), а порівнянням  $H_{CT}(x)$  і  $H_{max}(x)$  як з границею, до досягнення якої направлено застосування процедури стиснення даних. З урахуванням цього зауваження відношення

$$\mu = \frac{H_{CT}(x)}{H_{max}(x)} \quad (8)$$

показує, наскільки той або інший алгоритм стискає вхідний файл. Цей показник є об'єктивною кількісною характеристикою будь-якого квазізворотного методу стиснення даних і може використовуватися як в абсолютному сенсі, так і у відносному при порівнянні різних методів стиснення даних.

#### Перетворення форми інформації при знаходженні ламаних, що збігаються

Переміщення об'єкта у вигляді послідовності пар довгота-широта у деякій системі координат описують за допомогою треків. Їх задають, зазвичай, у вигляді xml-файлів. Популярними на сьогодні форматами є kml (Google), gpx (Garmin) та інші.

Аналіз даних, що містять треки, актуальний як при вирішенні задач реального життя, так і у ігрових ситуаціях. Наприклад, при прогнозуванні можливої поведінки гравця, виходячи із досвіду його попередників.

Вихідні дані до задачі: набір із  $n$  треків, отриманих попередньо, та  $n+1$ -й новий трек. Координати точок треків  $\{lat_i, lon_i\}, i \in \mathbb{Z}$  отримані із деякою періодичністю у часі, проте середньоквадратичне відхилення інтервалу часу між двома точками – не нульове. На практиці інтервали можуть відрізнятися один від одного у декілька разів. Координати точок належать множині  $\mathbb{R}$  і не відповідають жодній сітці.

Мета: знайти серед згаданих  $n$  треків такі, що подібні до  $n+1$ -го. При цьому подібність між двома треками може бути як повною (100% однакова кількість пар координат та їхніх значень), так і частковою (частини треків збігаються за координатами або у деякій області знаходяться близько один від одного). Ймовірність існування першого випадку близька до нуля. Крім того, він поглинається другим, тому для практичного використання є сенс розглянути випадок часткової подібності треків. При цьому слід буде визначити, що мається на увазі під близькістю треків на координатній площині. Пошук має бути якомога швидшим і бути придатним до реалізації у режимі реального часу з великим навантаженням, що створюється значною кількістю користувачів.

Зміщення між двома наборами випадкових даних можна визначити за допомогою кореляційного аналізу. Проте у даному випадкові коефіцієнт кореляції – не достатньо чутливий інструмент. Розглянемо відому формулу:

$$r_{lat_1 lat_2} = \frac{\sum_{j=1}^n (lat_{1j} - \overline{lat_1})(lat_{2j} - \overline{lat_2})}{\sigma_{lat_1} \sigma_{lat_2}}, \quad (9)$$

де  $lat_{1j}$  та  $lat_{2j}$  – широти відповідних точок двох треків. Індекс  $j$  – не тотожний індексові  $i$ , використаному при постановці задачі. Це – деякий індекс точок у послідовностях  $\{lat_{1j}\}$  та  $\{lat_{2j}\}$ , отриманих за допомогою інтерполяції для відповідних значень  $\{lon_j\}$ . Останні при цьому можуть бути рівномірно розподілені на інтервалі дослідження.

Вираз добре підходить для лінійних випадкових величин, оскільки базується на аналізі відхилень випадкових величин від їхніх середніх значень. Проте у випадкові довільних кривих він здатен представити тільки усереднену схему залежності.

Використання коефіцієнта кореляції не коректне за наявності петель у маршруті (одному і тому ж значенню довготи відповідатимуть декілька

значень широти). При цьому можлива хибна інтерпретація понять «згори», «знизу», «справа» та «зліва» при утворенні петлі. Наприклад, якщо треки деякий час були горизонтальними і паралельними із деяким зміщенням один відносно одного, а потім за дуговою траєкторією повернули на  $\pi$ , верхній трек стає нижнім, а нижній – верхнім. При цьому можлива зміна знака коефіцієнта кореляції.

Ділення на добуток середньоквадратичних відхилень нівелює масштаб, що для даної задачі не припустиме.

Крім того, використання формули (9) передбачає однаковий інтервал визначення обох треків, наприклад, за довготою. На практиці всі треки мають власні інтервали. Ймовірність подібності інтервалів, на яких визначені два треки – мізерна.

Отже, застосування коефіцієнта кореляції для вирішення поставленої задачі – не виправдане.

Аналогічно, застосування взаємно-кореляційної функції у даному випадкові також не допоможе, оскільки немає жодних гарантій того, що трек із набору містить цілий новий трек. Треки можуть перетинатися, дотикатися, бути близькими випадковим чином і довільну кількість разів. Крім того, застосування інтерполяції сповільнить роботу механізму порівняння внаслідок появи додаткових обчислень.

Інакший підхід: знаходження відстаней між треками або інтеграла функції, яку обмежують два треки на спільній області визначення. Такі варіанти частково знімають проблему петель, оскільки у випадкові інтегрування у якості змінної інтегрування можна вибрати довжину. Тоді певні значення широти точок обох треків відповідатимуть однаковим значенням довжини пройденого шляху. При цьому виникне нова проблема: якщо область визначення треку із набору вже закінчилася, а досліджуваного треку – ще ні, то частина його не буде проаналізована. Також перехід до відстані як координати вимагатиме апроксимації набору даних (з обмеженнями за значеннями довготи), що вимагатиме значної кількості додаткових обчислень.

Отже, слід розробити метод порівняння, складність роботи якого не буде залежною від характеру подібності треків і який не потребуватиме додаткових обчислень у процесі функціонування програмного забезпечення.

Можливий варіант вирішення задачі – перетворення форми інформації, яке усуне надлишковість і дасть змогу проводити аналіз шляхом порівнянь, а не обчислень. При цьому метод потребуватиме певної кількості початкових обчислень, кількість яких ростиме лінійно зі збільшенням кількості даних у трекові. Також збільшення продуктивності можна досягти за рахунок розподілу процесорного навантаження між різними обчислювальними вузлами.

У Google використали для сервісу Google maps підхід, що полягає у рекурсивному поділі проекції земної кулі на невеликі зображення розміром 256 x 256 px [12-14] (рис. 6). Отже, кожна ділянка отримує свій номер, що залежить від індекса масштабу.

При цьому використовують такі координатні системи:

1) довгота та широта, які однозначно ідентифікують положення точки на планеті – WGS84;

2) світові координати, які однозначно ідентифікують положення точки на планеті;

3) піксельні координати, що є посиланням на конкретний піксель на зображенні мапи при певному рівні збільшення;

4) координати тайла, що є посиланням на конкретний тайл на мапі при певному рівні збільшення.

Для даного підходу використано проекцію Меркатора. При цьому широта точки може перебувати у межах від -90 до 90 градусів, а довгота: від -180 до 180 градусів. Існує обмеження за значенням широти: 85.0511287798066 градуса.

Перетворивши координати із WGS-84 (чи якихось інших, придуманих, якщо ідеться про гру) у світові, можна знайти координати тайла, що міститиме піксель із даною точкою. Світові координати слід нормалізувати, поділивши  $x$  на 180, а  $y$  – на  $\pi$ :

$$\varphi = lat_i \cdot \pi / 180;$$

$$x_{norm} = lon_i / 180;$$

$$y_{norm} = \frac{1}{2\pi} \ln \left( \frac{1 + \sin(\varphi)}{1 - \sin(\varphi)} \right);$$

$$tileRow = 2^{zoom} \cdot (1 - y_{norm}) / 2;$$

$$tileColumn = 2^{zoom} (1 + x_{norm}) / 2,$$

де  $zoom$  – рівень збільшення;  $x$  – нормалізована довгота точки у світових координатах;  $y$  – нормалізована широта точки у світових координатах. Загальна кількість тайлів при певному рівні збільшення:

$$nrTiles = 2^{2zoom}.$$

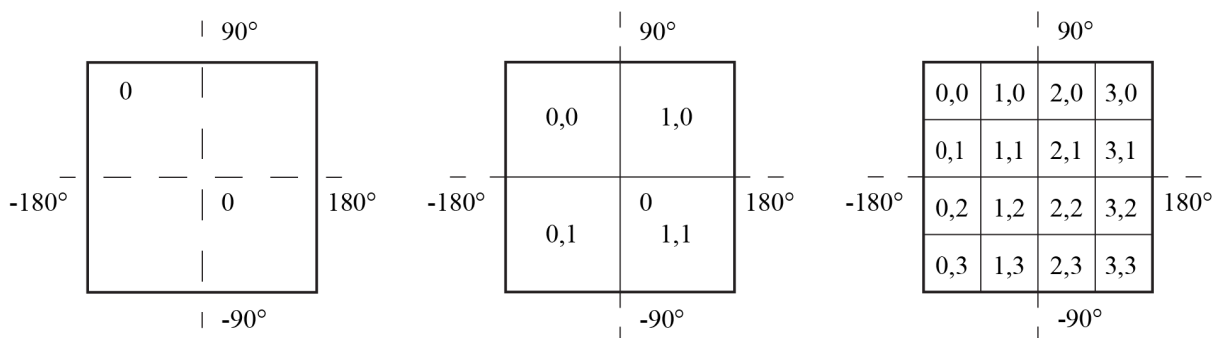


Рисунок 6 – Нумерація тайлів у Google Maps

Кількість рядків та стовпчиків визначається виразом

$$nrColumns = nrRows = 2^{zoom}. \quad (10)$$

Отже, на даному етапі відомі координати точки у просторі WGS-84, світові координати  $i$ , як наслідок – номер рядка та стовпчика, що містять тайл із зображенням, що напевне включає точку із заданими координатами.

Вище визначено, що порівнювати треки, працюючи із координатами напряму, не доцільно з різних причин. Перетворимо форму інформації.

Циліндрична проекція земної кулі на різних рівнях збільшення складається з різної кількості тайлів. Кожен із них має унікальне ім'я. Google maps використовує рівні 0-17 (за потреби можна зробити і більше).

Тайл із рівнем 0 має ім'я «0». На наступному рівні тайл ділять на 2 частини за широтою та 2 – за довготою. Таким чином, на першому рівні зліва-направо і згори-униз новоутворені тайли отримують імена «0,0», «0,1»; «1,0», «1,1». На другому рівні – відповідно: «0,0», «0,1», «0,2», «0,3» ... «3,0», «3,1», «3,2», «3,3». Фактично, імена тайлів можна трактувати як індекси елементи матриці. Кількість рядків та стовпчиків у матриці визначається виразом (10). Отже, знаючи рівень збільшення та координати точки на місцевості, можна визначити загальну кількість тайлів, а також номер рядка та стовпчика, який містить тайл, що відповідає заданим координатам.

Загалом трансформацію географічних координат у номер тайла можна було б використати вже для описаної вище задачі. Тоді опис трека складався б з пар номерів тайлів. При цьому певною мірою усунулася б надлишковість інформації, оскільки при порівнянні треків потрібна менша точність, ніж та, яку забезпечує трек.

Недоліком цього варіанта є неможливість здійснювати аналіз із різною точністю (даний термін у контексті задачі, що розглядається, буде розкритий далі). Наприклад, шукати подібні треки в околі 100 км чи 500 м від точок даного треку.

Модифікуємо перетворення (рис. 7).

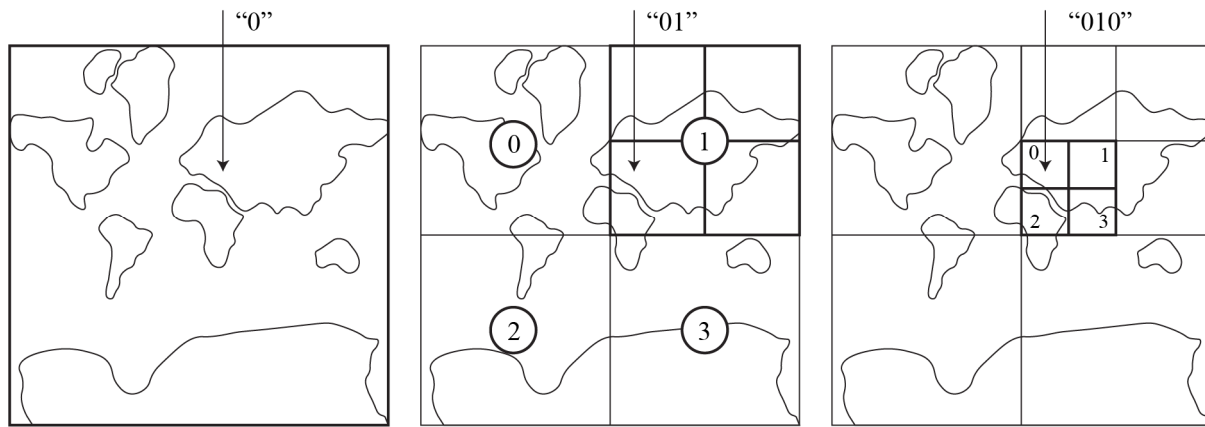


Рисунок 7 – Рекурсивний поділ поверхні кулі на квадранти

Визначаємо номер тайла, якому належить точка. На 0 рівні це – «0». Ділимо на наступному (1) рівні тайл рівня 0 на 4 частини (2 за широтою та 2 – за довготою) і присвоюємо їм імена «0», «1», «2» та «3». Кожне з імен отримує префікс – ім'я материнського тайла. Таким чином, повні імена цих тайлів такі: «00», «01», «02» та «03». Встановивши до якого із 4 тайлів рівня 1 належить точка із заданими координатами, ділимо його у свою чергу. Нехай, наприклад, це тайл із іменем «03». Тоді 4 тайли рівня 2 (один із них містить дану точку) матимуть імена «030», «031», «032» та «033». Продовжуємо процедуру до отримання максимальної роздільної здатності методу. У результаті перетворення точці із координатами  $lat_i, lon_i$  відповідатиме деякий рядок. Наприклад, «011232212313120120».

Здійснивши перетворення для усіх точок треку, переходимо від способу задання трека за допомогою координат до опису його у вигляді послідовності рядків, уникаючи дублювання.

Приклад перетворення наведено на рис. 8. Трек, позначений суцільною лінією, описує послідовність: 0000, 0002, 0003, 0012, 0030, 0031, 0120, 0122, 0300, 0302, 0303, 0312, 0330, 0331, 0333, 0332, 0323, 0322, 0320, 0231, 0230, 0212, 0203, 0202.

Трек (пунктирна лінія), що частково збігається із описаним: 0000, 0002, 0003, 0012, 0030, 0031, 0120, 0121, 0132, 0133.

Отриману інформацію додаємо до таблиці бази даних, що має структуру:  $idt$  – номер трека – рядок, що описує тайл,  $z1, z2, z3, z4$  – номер тайла на рівні  $n$ .

Описуємо вказаним чином всі наявні треки.

Для нового трека повторюємо те ж, тільки результат записуємо до тимчасової таблиці.

Скориставшись JOIN знаходимо перетин таблиць (рис. 9).

Підрахувавши скільки спільних точок має кожен з треків видачі і знайшовши співвідношення із загальними довжинами (кількостями записів) треків, знаходимо ступінь подібності. Точність порівняння

задаємо, вибравши довжину рядка для порівняння (правий символ відповідає найбільшій точності).

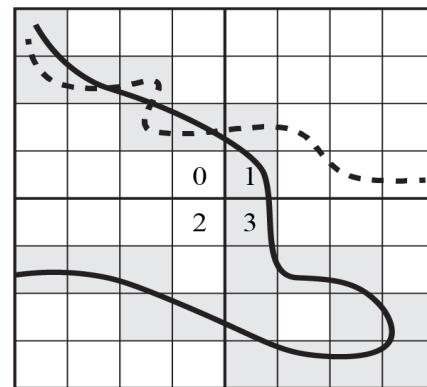


Рисунок 8 – Приклад часткового збігу треків

id	tid	z1	z2	z3	z4
1	1	0	0	0	0
2	1	0	0	0	2
3	1	0	0	0	3
4	1	0	0	1	2
...	1				
7	1	0	1	2	0
8	1	0	1	2	2
9	1	0	3	0	0

Рисунок 9 – Результат виконання JOIN

Звичайно, у цьому випадкові подібними вважатимуть точки, що знаходяться усередині одного тайла, а розмір його визначається глибиною перетворення.

Отже, здійснивши перетворення форми інформації, усуваємо потребу у проведенні арифметичних операцій і пошук подібних треків зводимо до роботи із символами, що відбувається значно швидше.

Нескладною, проте необхідною модифікацією методу є включення до аналізу імен тайлів, суміжних до тайлів треку (рис. 10).



+	+	+	+	+	+	+	+
+	0023	0032	0033	0122	0123	0032	+
+	+	+	+	+	+	+	+

Рисунок 10 – Результат виконання JOIN

Комірки, позначені знаком «+», – суміжні до комірок, яким належать точки треку. За їх рахунок можна два треки, позначені на рисунку як відрізки, трактувати як близькі. Інакше треки, що потрапляють до різних комірок на вищому рівні, не можуть бути трактовані як близькі на нижчому рівні. Водночас така ситуація є малоймовірною у реальних ігрових умовах. Довжину треків можна знаходити за теоремою Вінценті [15].

### Висновки

У статті розглянуто методи компресії даних та перетворення форми інформації, що можуть бути використані і в ігровій індустрії зокрема.

Зазвичай ігрове програмне забезпечення розробляють за допомогою фреймворків, які реалізують базовий набір операцій (організація ігрового простору, симуляція фізики тощо). Також розробники пропонують деякі операції для роботи зі звуковими ефектами та музичним супроводом. При цьому використовують цілі музичні композиції чи їхні фрагменти, що відтворюють у певній послідовності. Запропоновано програмну модель блоку звуковідтворення для ігрової платформи, що дозволяє формувати музичний супровід на основі рефренної моделі, яка передбачає циклічне відтворення не тільки фрагментів звукового супроводу, а й їхніх компонентів, що дає змогу одержати загальний коефіцієнт стиснення інформації порядку декількох десятків і більше.

Для здійснення оцінки ефективності методу стиснення даних запропоновано використовувати комплексний коефіцієнт ефективності, який може бути представлений як добуток чи сума факторів, що впливають на ефективність методу стиснення даних. Цей показник враховує кількість операцій, які витрачаються для обробки одного параметра, кількість програм в машинному коді алгоритму; об'єм оперативної і постійної пам'яті та кількість стандартних елементів, необхідних для

створення спеціалізованого блоку стиснення інформації.

Запропоновано метод перетворення форми інформації при знаходженні співпадаючих із заданою точністю ділянок довільних кривих, представлених у вигляді набору пар координат на площині. Перетворення реалізує перехід від числових координат до символічних рядків. Показано можливість побудови розподіленої системи з використанням сервера СУБД.

ДВНЗ «Прикарпатський національний університет імені Василя Стефаника» у складі групи українських та закордонних закладів вищої освіти та профільних компаній є членом консорціуму GameHub, що виконує грантовий проект Erasmus+ KA2 CBHE №561728-EPP-1-2015-1-ES-EPPKA2-CBHE-IP «Співробітництво між університетами та підприємствами у сфері ігрової індустрії в Україні – GameHub», який фінансується Європейською Комісією (проект триває упродовж 2016 – 2018 р.р.) [18].

У проекті задіяні як викладачі та студенти університету, так і ветерани АТО; безробітні; працівники центрів зайнятості; представники ІТ-компаній. На сьогодні проектом охоплено понад 90 студентів та 30 осіб вказаних вище категорій.

В університеті створено академічну групу [19], якою розроблено комплекси навчального та методичного забезпечення дисциплін, що викладаються у межах проекту, та проведено ряд тренінгів.

За фінансової підтримки Європейської Комісії на базі кафедри інформатики створено та обладнано сучасну лабораторію з розроблення ігрового програмного забезпечення GameLab. Лабораторія укомплектована PC/Mac-комп'ютерною технікою і дає змогу вирішувати задачі програмування ігрових додатків та створення їхнього музичного супроводу.

На кафедрі розроблено та викладаються спеціалізовані курси для студентів спеціальності «Інформатика»: 3 курс (бак.) – дисципліни «Розробка комп'ютерних ігор» та «Технології розробки комп'ютерних ігор», 1 курс (маг., «Комп'ютерні науки») – дисципліна «Створення звукового та музичного супроводу для комп'ютерних ігор».

Виконавці проекту взяли участь у ряді тренінгів в Україні та за її межами, а також виступили організаторами тренінгів «Особливості використання фреймворку SpriteKit для програмування ігор під iOS» (квітень 2017), «Розробка ігрових додатків з використанням фреймворка Cocos2d-x» (травень 2017 р.). Перебіг та результати проекту обговорено на засіданнях Міжнародної науково-технічної конференції «Інформаційні технології та комп'ютерне моделювання» (2017, 2018 р.р. [16; 17]).

До вирішення наукових задач проекту залучені також і студенти. Студенти-інформатики узяли участь у роботі Міжнародної науково-технічної конференції студентів, аспірантів та молодих вчених "Комп'ютерні науки, інформаційні технології та

системи управління" (CSYSC-2017, Івано-Франківськ, квітень 2017 р.), міжнародної студентської науково-технічної конференції Konferencja Studenckich Kół Naukowych (Краків, травень 2017, 2018 р.р.).

### Список літератури

1. Michael Sweet. *Writing interactive music for video games: a composer's Guide*. Addison-Wesley. 2015.
2. Aaron Marks. *The Complete Guide to Game Audio For Composers, Musicians, Sound Designers, and Game Developers*. Elsevier. 2009.
3. G. W. Childs. *Creating Music and Sound for Games*. Thomson. 2007.
4. Boulanger R., Lazzarini V. *The Audio Programming Book*. The MIT Press. 2011. 920 p.
5. Phillips W. *A Composer's Guide to Game Music*. The MIT Press. 2014. 288 p.
6. Орищенко В.И. Сжатие данных в системах сбора и передачи информации / В.И. Орищенко, В.А. Санников, В.А. Свириденко. – М.: Радио и связь, 1985. – 184 с.
7. Ольховський Ю.Б. Сжатие данных при телеизмерениях / Ю.Б. Ольховський, О.Н. Новоселов, А.П. Мановцев. – М.: Сов. радио, 1971. – 340 с.
8. Свириденко В.А. Анализ систем со сжатием данных / В.А. Свириденко. – М.: Связь, 1977. – 184 с.
9. Іляш Ю.Ю. Формування стиснутих повідомлень шляхом кодування рівнів квантування / Ю.Ю. Іляш // Інформаційні технології та комп'ютерна інженерія: матеріали статей п'ятої міжнар. наук.-практ. конф., м. Івано-Франківськ, 27-29 травня 2015 р. – Івано-Франківськ: Супрун В.П., 2015. – С. 100 – 101.
10. Іляш Ю.Ю. Критерії та оцінка ефективності зменшення надлишковості інформаційних потоків/Іляш Ю.Ю. // «Восточно-Европейский журнал передовых технологий» № 1/2(55), 2012. – С. 54 – 58.
11. Яцків Н.Г. (2003). Методи та засоби стиснення даних в розподілених комп'ютерних системах на основі кодів поля Галуа (Автореф. дис. канд. техн. наук: 05.13.13). Нац. ун-т "Львів. Політехніка".
12. <https://developers.google.com/maps/documentation/javascript/coordinates>
13. Dincer, A., Uraz. B. *Google Maps JavaScript API Cookbook*. Birmingham – Mumbai: Packt Publishing. 2013.
14. Svennerberg G. *Beginning Google Maps API 3*. Apress. 2010.
15. Vincenty, Thaddeus. *Geodetic inverse solution between antipodal points. (Technical report)*. DMAAC Geodetic Survey Squadron. doi: 10.5281/zenodo. 32999.
16. Горелов В. О. Гейміфікація навчання / Горелов В. О., Sala Dariusz.// Інформаційні технології та комп'ютерне моделювання: матеріали Міжнародної науково-практичної конференції, м. Івано-Франківськ, 23-28 травня 2017 р. – Івано-Франківськ: Супрун В.П., 2017. – С. 136 – 139.
17. Горелов В. О. Використання методів машинного навчання для ігрового моделювання розподілених систем управління / В.О. Горелов, Л.Б. Петришин // Інформаційні технології та комп'ютерне моделювання. Матеріали міжнародної наукової конференції конференції. Івано-Франківськ. Голіней., 2018. – С. 101 – 104.
18. Співробітництво між університетами та підприємствами в сфері ігрової індустрії в Україні — GameHub [Електронний ресурс]. – Режим доступу: <http://gamehub-cbhe.eu/ua/>
19. GameHub-команда ДВНЗ "Прикарпатський національний університет імені Василя Стефаника" [Електронний ресурс]. – Режим доступу: <http://gamehub-cbhe.eu/ua/project-consortium/vasyl-stefanyk-precarpathian-national-university-team/>

Стаття надійшла до редколегії 06.09.2018

**Рецензент:** д-р техн. наук, проф. С.Д. Бушуєв, Київський національний університет будівництва і архітектури, Київ.

#### Горелов Виталий Олевтинович

Кандидат технических наук, доцент кафедры компьютерных наук и информационных систем, [orcid.org/0000-0002-2106-8704](https://orcid.org/0000-0002-2106-8704) ГВУЗ «Прикарпатский национальный университет имени Василия Стефаника», Івано-Франковск

#### Іляш Юрій Юрьевич

Кандидат технических наук, доцент кафедры компьютерных наук и информационных систем, [orcid.org/0000-0003-3724-4183](https://orcid.org/0000-0003-3724-4183) ГВУЗ «Прикарпатский национальный университет имени Василия Стефаника», Івано-Франковск

#### Ровинский Виктор Анатольевич

Кандидат технических наук, доцент кафедры компьютерных наук и информационных систем, [orcid.org/0000-0001-8454-8580](https://orcid.org/0000-0001-8454-8580) ГВУЗ «Прикарпатский национальный университет имени Василия Стефаника», Івано-Франковск

## МЕТОДЫ СЖАТИЯ ДАННЫХ И ПРЕОБРАЗОВАНИЯ ФОРМЫ ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ ИГРАХ

**Аннотация.** Рассмотрены вопросы, связанные с уплотнением данных и преобразованием формы информации при разработке компьютерных игр. Показан методологический подход, позволяющий эффективно реализовать уменьшение объема аудиоданных по сравнению с использованием обычных музыкальных файлов со сжатием. Предложен метод и проведены экспериментальные исследования, результаты которых могут быть использованы при разработке игрового программного обеспечения. Определение эффективности методов компрессии важно для дальнейшего выбора конкретного метода. Рассмотрена система числовых характеристик, которые позволяют сравнивать методы между собой, предложен комплексный подход к оценке коэффициента сжатия и показатель, который является объективной количественной характеристикой любого квазиобратимого метода сжатия данных. Рассмотрен метод преобразования формы информации при исследовании сходства ломаных линий, заданных набором пар координат. Предложено преобразование числовых значений координат на плоскости в символьные строки, которые могут обрабатываться средствами управления базами данных, что позволяет получить высокое быстродействие и устраняет необходимость проведения арифметических вычислений в реальном времени.

**Ключевые слова:** звуковой файл; компрессия; эффективность компрессии; преобразование формы информации; тайл; СУБД

### **Horielov Vitalii**

PhD (Eng.), assistant professor of Computer Science and Information Systems Department, [orcid.org/0000-0002-2106-8704](https://orcid.org/0000-0002-2106-8704)  
State Higher Education Institution «Vasyl Stefanyk Precarpathian National University», Ivano-Frankivsk

### **Iliash Yurii**

PhD (Eng.), assistant professor of Computer Science and Information Systems Department, [orcid.org/0000-0003-3724-4183](https://orcid.org/0000-0003-3724-4183)  
State Higher Education Institution «Vasyl Stefanyk Precarpathian National University», Ivano-Frankivsk

### **Rovinskyi Viktor**

PhD (Eng.), assistant professor of Computer Science and Information Systems Department, [orcid.org/0000-0001-8454-8580](https://orcid.org/0000-0001-8454-8580)  
State Higher Education Institution «Vasyl Stefanyk Precarpathian National University», Ivano-Frankivsk

## DATA COMPRESSION AND TRANSFORMATION OF THE FORM OF INFORMATION METHODS IN COMPUTER GAMES

**Abstract.** The article deals with the computer game development issues connected with the data compression and transformation of the form of information. The methodological approach which allows effective reducing of audio file size in comparison with compressed audio files has been proposed. The method has been proposed and the research has been carried out. The results obtained may be used in the development of computer game software. Determining the effectiveness of compression methods is important for a specific method selection. The system of numerical characteristics that allows to compare the compression methods is considered, the complex approach to estimating the coefficient of compression and the index which is an objective quantitative characteristic of any quasi-reversible data compression method has been proposed. The method of transformation of the form of information for the detection of similarity of polylines given by a set of coordinates is considered. It is proposed to convert the numeric coordinates to character lines that can be processed by means of database management systems, which allows to obtain high speed and eliminates the need for real time arithmetic calculations.

**Keywords:** sound file; compression; compression efficiency; transformation of information form; tile; DMS

### References

1. Sweet, Michael. (2015). *Writing interactive music for video games: a composer's Guide*. Addison-Wesley.
2. Marks, Aaron. (2009). *The Complete Guide to Game Audio For Composers, Musicians, Sound Designers, and Game Developers*. Elsevier.
3. Childs, G. (2007). *Creating Music and Sound for Games*. Thomson.
4. Boulanger, R., Lazzarini, V. (2011). *The Audio Programming Book*. The MIT Press, 920.
5. Phillips W. (2014). *A Composer's Guide to Game Music*. The MIT Press, 288.
6. Orishchenko, V. (1985). *Data compression in information collection and transmission systems*. *Radio and communication*, 184. [in Russian]
7. Olkhovski, Yu. (1971). *Data compression in telemetry*. *Sov. radio*, 340. [in Russian]
8. Sviridenko, V. (1977). *Analysis of systems with data compression*. *Communication*, 184. [in Russian]

9. Iliash, Yu. (2015). Creation of compressed messages by quantization levels encoding. "Information technologies and computer engineering" proceedings of the 5<sup>th</sup> International Scientific Conference (p. p. 100-101). Ivano-Frankivsk, Suprun. [in Ukrainian]
10. Iliash, Yu. (2012). Criteria and evaluation of information flows redundancy efficiency. *East European journal of enterprise technologies*, 54–58. [in Ukrainian]
11. Yackiv, N. (2003). *Methods and means of data compression based on Galois codes in distributed computer systems* (Doctoral dissertation 05.13.13). Lviv Politechnical University. [in Ukrainian]
12. <https://developers.google.com/maps/documentation/javascript/coordinates>
13. Dincer, A., Uraz, B. (2013). *Google Maps JavaScript API Cookbook*. Birmingham – Mumbai: Packt Publishing.
14. Svennerberg, G. (2010). *Beginning Google Maps API 3*. Apress.
15. Vincenty, Thaddeus. (1975). *Geodetic inverse solution between antipodal points*. (Technical report). DMAAC Geodetic Survey Squadron. doi: 10.5281/zenodo.32999.
16. Goryelov, V., Sala, D. (2017). Gamification of education. "Information technologies and computer modelling" proceedings of the International Scientific Conference 2017, May, 15th to 20th. Ivano-Frankivsk: Suprun. 136–139. [in Ukrainian]
17. Goryelov, V., Petryshyn, L. (2018). Use of Machine Learning Methods for Gaming-Based Modelling of Distributed Management Systems. "Information technologies and computer modelling" proceedings of the International Scientific Conference 2018, May, 14th to 19th. Ivano-Frankivsk: Holinei. 101–104. [in Ukrainian]
18. University-enterprises cooperation in game industry in Ukraine – GameHub [Online]. Available: <http://gamehub-cbhe.eu/ua/>
19. GameHub team of Vasyl Stefanyk Precarpathian National University [Online]. Available: <http://gamehub-cbhe.eu/ua/project-consortium/vasyl-stefanyk-precarpathian-national-university-team/>

---

#### Посилання на публікацію

APA Horielov, V., Iliash Y. & Rovinskyi, V. (2018). Data compression and transformation of the form of information methods in computer games. *Management of development of complex systems*, 35, 93 – 104.

ДСТУ Горєлов В. О. Методи ущільнення даних та перетворення форми інформації у комп'ютерних іграх [Текст] // В.О. Горєлов, Ю.Ю. Іляш, В.А. Ровінський / Управління розвитком складних систем. – 2018. – № 35. – С. 93 – 104.

**Робота виконана у рамках проекту програми Еразмус+ КА2 – Розвиток потенціалу вищої освіти. №561728-EPP-1-2015-1-ES-EPPKA2-CBHE-JP-"GameHub: Співробітництво між університетами та підприємствами в сфері ігрової індустрії в Україні".**