УДК 004.627

ЗАВАДОСТІЙКЕ ЦІЛОЧИСЕЛЬНЕ КОДУВАННЯ ГЕОМЕТРІЇ СІТКОВИХ 3D ОБ'ЄКТІВ

Самусь Н.С., Ошаровська О.В.

Одеська національна академія зв'язку ім. О.С. Попова,
65029, Україна, м. Одеса, вул. Ковальська,1.

osharovskaya@mail.ru, natalia_samus@ukr.net

ПОМЕХОУСТОЙЧИВОЕ ЦЕЛОЧИСЛЕННОЕ КОДИРОВАНИЕ ГЕОМЕТРИИ СЕТОЧНЫХ 3D ОБЬЕКТОВ

Самусь Н.С., Ошаровская Е.В.

Одесская национальная академия связи им. А.С. Попова,
65029, Украина, г. Одесса, ул. Ковальская,1.

osharovskaya@mail.ru, natalia_samus@ukr.net

NOISE-RESISTANT INTEGER ENCODING 3D MESH GEOMETRY OBJECTS

Samus N.S., Osharovskaya E.V.

O.S. Popov Odessa national academy of telecommunications,

1 Kovalska St., Odessa 65029, Ukraine.

osharovskaya@mail.ru, natalia_samus@ukr.net

Анотація. У статті розглядане кодування вершин вибраної сіткової 3D моделі за допомогою двох методів: системи залишкових класів (СЗК) і чисел Фібоначчі. Представлено аналіз ефективності таких методів кодування, а також проведено порівняння зі звичайним двійковим представленням чисел. В результаті зроблено висновки щодо недоліків і переваг таких алгоритмів кодування вершин сітки.

Ключові слова: 3D трикутна сітка, кодування вершин, система залишкових класів, числа Фібоначчі

Аннотация. В статье рассматривается кодирование вершин выбранной сеточной 3D модели с помощью двух методов: системы остаточных классов (СОК) и чисел Фибоначчи. Анализируется эффективность таких методов кодирования, а также проводится сравнение с обычным двоичным представлением чисел. В результате сделаны выводы о недостатках и преимуществах таких алгоритмов кодирования вершин сетки.

Ключевые слова: 3D треугольная сетка, кодирование вершин, система остаточных классов, числа Фибоначчи

Abstract. In this article coding of vertices in selected 3D mesh model using two methods, namely residue number system (RNS) and Fibonacci numbers, is considered. The efficiency of these coding methods is analyzed, and also comparison with conventional binary representation of numbers is conducted. As a result, conclusions on the advantages and disadvantages of such coding algorithms of vertices are done.

Keywords: 3D triangular mesh, coding of vertices, Residue number system (RNS), Fibonacci numbers

ВСТУП

При цифровій обробці важливим фактором є вибір системи кодування. На сьогоднішній день ще широко застосовують методи і системи двійкового позиційного кодування, але все рівно ведеться впровадження нових непозиційних (чи змішаних) методів кодування, наприклад, таких як, система залишкових класів, послідовності Фібоначчі, поля Галуа та інші. Хоча вони і характеризуються надлишковістю, проте вони є досить завадостійкими та дозволяють виявляти помилки, в деяких випадках навіть виправляти їх.

У попередніх статтях було розглянуто кодування зв'язності ділянки сіткової 3D моделі за допомогою алгоритму Edgebreaker в поєднанні з кодом Хаффмана [1] чи арифметичним кодом [2]. Але також представляє інтерес кодування без втрат геометрії таких сіткових моделей.

КОДУВАННЯ ГЕОМЕТРІЇ

Тому розглянемо кодування вершин вибраної ділянки сіткової 3D моделі.

Таблиця 1 – Координати всіх вершин вибраної ділянки сіткової моделі

№ точки	Координата *X*	Координата *Y*	Координата *Z*
0	13.435	62.859	13.107
1	12.680	63.424	7.564
2	13.896	57.103	7.057
3	12.319	58.019	0.066
4	11.551	63.812	-0.719
5	10.761	57.371	-9.177
6	10.143	63.596	-9.984
7	10.631	55.474	-17.850
8	8.877	68.742	7.546
9	8.799	68.588	-0.597
10	4.420	71.392	7.741
11	8.936	68.805	13.746

Далі знайдемо максимальне і мінімальне значення кожної з координат *X, Y, Z,* які необхідні для наступних дій. Результат помістимо в таблицю 2.

Таблиця 2 – Максимальні та мінімальні значення кожної координати

	Координата *X*	Координата *Y*	Координата *Z*
Максимальне значення	13.896	71.392	13.746
Мінімальне значення	4.420	55.474	-17.850

Тепер пронормуємо всі значення і переведемо отримані числа в цілі числа. Результати розрахунків оформимо у вигляді таблиці 3.

СИСТЕМА ЗАЛИШКОВИХ КЛАСІВ

Для кодування вершин використаємо спочатку систему залишкових класів (СЗК).

В системі залишкових класів числа представляються залишками від ділення числа *A*, представленого в позиційній системі числення, на вибрану систему взаємно простих модулів $p_1, p_2, \dots, p_n$, при цьому діапазон представлення чисел [3]:

$$R = \prod_{i=1}^{n} p_i \tag{1}$$

Число в СЗК має вигляд:

$$A = (r_1, r_2, \dots, r_n), \tag{2}$$

Таблиця 3 – Приведення до цілочисельних значень всіх координат

№ точки	Координата *X*		Координата *Y*		Координата *Z*	
	пронормоване значення	значення ×1000	пронормоване значення	значення ×1000	пронормоване значення	значення ×1000
0	0.951	951	0.464	464	0.980	980
1	0.872	872	0.499	499	0.804	804
2	1	1000	0.102	102	0.788	788
3	0.834	834	0.160	160	0.567	567
4	0.753	753	0.524	524	0.542	542
5	0.669	669	0.119	119	0.274	274
6	0.604	604	0.510	510	0.249	249
7	0.655	655	0	0	0	0
8	0.470	470	0.834	834	0.804	804
9	0.462	462	0.824	824	0.546	546
10	0	0	1	1000	0.810	810
11	0.477	477	0.837	837	1	1000

Число в СЗК має вигляд:

$$A = (r_1, r_2, \ldots, r_n), \tag{3}$$

де $r_i = A(mod p_i)$ – залишки від ділення.

Візьмемо, наприклад, наступну систему взаємно простих чисел $(p_1, p_2, p_3) = (7, 11, 13)$. Отримаємо діапазон $R = 7 \times 11 \times 13 = 1001 \rightarrow [0; M) = [0; 1001)$.

Знайдемо представлення цілих значень координат кожної точки:

Точка 0

951 → (951mod7, 951mod11, 951mod13) → (6, 5, 2)

464 → (464mod7, 464mod11, 464mod13) → (2, 2, 9)

980 → (980mod7, 980mod11, 980mod13) → (0, 1, 5)

Точка 1

872 → (4, 3, 1)

499 → (2, 4, 5)

804 → (6, 1, 11)

Точка 2

1000→ (6, 10, 12)

102 → (4, 3, 11)

788 → (4, 7, 8)

Точка 3

834 → (1, 9, 2)

160 → (6, 6, 4)

567 → (0, 6, 8)

Точка 4

753 → (4, 5, 12)

524 → (6, 7, 4)

542 → (3, 3, 9)

Точка 5

669 → (4, 9, 6)

119 → (0, 9, 2)

274 → (1, 10, 1)

Точка 6

604 → (2, 10, 6)

510 → (6, 4, 3)

249 → (4, 7, 2)

Точка 7

655 → (4, 6, 5)

0 → (0, 0, 0)

0 → (0, 0, 0)

Точка 8

470 → (1, 8, 2)

834 → (1, 9, 2)

804 → (6, 1, 11)

Точка 9

462 → (0, 0, 7)

824 → (5, 10, 5)

546 → (0, 7, 0)

Точка 10

0 → (0, 0, 0)

1000 → (6, 10, 12)

810 → (5, 7, 4)

Точка 11

477 → (1, 4, 9)

837 → (4, 1, 5)

1000 → (6, 10, 12)

Тепер порівняємо застосування двійкових значень чисел координат і двійкових значень отриманих чисел після застосування системи залишкових класів (табл.4).

Таблиця 4 – Порівняння двійкового представлення координат і представлення з застосуванням системи залишкових класів

Координата	Двійкове представлення координати	Координата, представлена за допомогою СЗК	Двійкове представлення координати
951 464 980	1110110111 111010000 1111010100	(6, 5, 2) (2, 2, 9) (0, 1, 5)	110 101 010 0010 0010 1001 000 001 101
872 499 804	1101101000 111110011 1100100100	(4, 3, 1) (2, 4, 5) (6, 1, 11)	100 011 001 010 100 101 0110 0001 1011
1000 102 788	1111101000 1100110 1100010100	(6, 10, 12) (4, 3, 11) (4, 7, 8)	0110 1010 1100 0100 0011 1011 0100 0111 1000
834 160 567	1101000010 10100000 1000110111	(1, 9, 2) (6, 6, 4) (0, 6, 8)	0001 1001 0001 110 110 100 0000 0110 1000
753 524 542	1011110001 1000001100 1000011110	(4, 5, 12) (6, 7, 4) (3, 3, 9)	0100 0101 1100 110 111 100 0011 0011 1001
669 119 274	1010011101 1110111 100010010	(4, 9, 6) (0, 9, 2) (1, 10, 1)	0100 1001 0110 0000 1001 0001 0001 1010 0001
604 510 249	1001011100 111111110 11111001	(2, 10, 6) (6, 4, 3) (4, 7, 2)	0010 1010 0110 110 100 011 100 111 010
655 0 0	1010001111 0 0	(4, 6, 5) (0, 0, 0) (0, 0, 0)	100 110 101 0 0 0 0 0 0
470 834 804	111010110 1101000010 1100100100	(1, 8, 2) (1, 9, 2) (6, 1, 11)	0001 1000 0010 0001 1001 0010 0110 0001 1011
462 824 546	111001110 1100111000 1000100010	(0, 0, 7) (5, 10, 5) (0, 7, 0)	000 000 111 0101 1010 0101 000 111 000
0 1000 810	0 1111101000 1100101010	(0, 0, 0) (6, 10, 12) (5, 7, 4)	0 0 0 0110 1010 1100 101 111 100
477 837 1000	111011101 1101000101 1111101000	(1, 4, 9) (4, 1, 5) (6, 10, 12)	0001 0100 1001 100 001 101 0110 1010 1100

Отже, в 6 випадках використання системи залишкових кодів дало позитивний результат, в 3 випадках – той же результат, в інших – було гіршим в порівнянні зі звичайним двійковим представленням.

Тобто виграшу в чистому вигляді ми не отримуємо, однак при розпаралелюванні короткі комбінації не будуть приводити до значних помилок.

Вирахуємо середнє значення біт на координату:

- для звичайного двійкового представлення координати

$$C_{bin} = \sum_{m=1}^{10} m \cdot \frac{l}{N_{max}}, \qquad (4)$$

де m – кількість розрядів, що потрібно для кодування вершин двійковим кодом змінної довжини; у нашому випадку найменшій розрядності відповідає число 0, яке кодується 1 бітом, а максимальна довжина в 10 біт відповідає числу 1000; l – кількість чисел з розрядністю m; N_{max} – загальна кількість координат; у нашому випадку 1001.

$$C_{bin} = 1\times 1/1001 + 2\times 3/1001 + 3\times 4/1001 + 4\times 8/1001 + 5\times 16/1001 +$$
$$+6\times 32/1001 + 7\times 64/1001 + 8\times 128/1001 + 9\times 256/1001 + 10\times 489/1001 =$$
$$= 8{,}98001999 \text{ біт/коорд.}$$

- з використанням системи залишкових кодів

$$C_{\text{СЗК}} = \sum_{m=1}^{4} 3m \cdot \frac{l}{K_{\max}}, \qquad (5)$$

де $K_{\max} = p_{\max} \cdot p_{\max} \cdot p_{\max}$ – кількість всіх можливих комбінацій з залишків;

$p_{\max}$ – максимальній із взаємно простих модулів, який відповідає максимальній кількості можливих комбінацій на одному місці.

$$C_{bin} = 3\times 1/2197 + 6\times 63/2197 + 9\times 448/2197 + 12\times 1685/2197 =$$
$$= 11{,}21210742 \text{ біт/коорд.}$$

Хоча збільшення ефективності кодування в порівнянні з простим двійковим представленням числа отримано не було, але з'явилися значні переваги:

- можливість паралельного оброблення кожного з трьох масивів залишків, що підвищує швидкодію роботи алгоритму стиснення;
- малорозрядність залишків (в даному випадку максимум 4 розряди);
- реалізація принципу конвеєрної обробки інформації;
- висока точність, надійність, спроможність до самокорекції.

ПОСЛІДОВНІСТЬ ФІБОНАЧЧІ

Далі для кодування вершин використаємо числа Фібоначчі. Будь яке натуральне число однозначно можна представити у вигляді суми чисел Фібоначчі [4]:

$$N = a_1 f_1 + a_2 f_2 + \dots a_i f_i, \qquad (6)$$

де $a_i = \{0, 1\}$ – бінарне представлення кожного розряду заданого числа, в якому 0 відповідає відсутності даного числа Фібоначчі в сумі заданого числа, а 1 – його присутності; $f_i = f_{i-1} + f_{i-2}$ – значення ваги розряду.

Замітимо також, що якщо в розкладанні числа n присутнє f_j, то в цьому розкладанні не може бути числа f_{j+1}. А тому можна до кінця коду використовувати додаткову одиницю. *Тоді дві одиниці підряд будуть однозначно визначати кінець кодування поточного числа. Це є дуже суттєвою перевагою цього методу.*

Нам знадобляться перші 15 чисел Фібоначчі:

1	2	3	5	8	13	21	34	55	89	144	233	377	610	987

Представимо всі вершини вибраної ділянки числами Фібоначчі.

Точка 0
951→100101000101011
464→00101010100011
980→100001010101011

Точка 1
872→000010100001011
499→10101010010011
804→001001010010011

Точка 2
1000→0000010000000011
102→00000100011
788→000000010010011

Точка 3
834→101000101010011
160→001001000011
567→10101001001011

Точка 4
753→010101010100011
524→00100000001011
542→00000010001011

Точка 5
669→101000001000011
119→10001010011
274→0101000100011

Точка 6
604→01010010101011
510→01001001010011
249→0010010000011

Точка 7
655→001010010000011
0→11
0→11

Точка 8
470→10100000010011
834→101000101010011
804→001001010010011

Точка 9
462→10001010100011
824→010001001010011
546→10100010001011

Точка 10
0→11
1000→0000010000000011
810→100000001010011

Точка 11
477→00101000010011
837→010100101010011
1000→0000010000000011

Для кодування 14 координат потребувалося 15 біт, ще 14 – 14 біт, на 3 – 16 біт, ще на 3 – 2 біта, на 2 – 13 біт, на наступні 2 – 11 біт і на 1 зі всіх координат – 12 біт.

ПОЄДНАННЯ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ І ПОСЛІДОВНОСТІ ФІБОНАЧЧІ

Далі для кодування вершин використаємо числа Фібоначчі до отриманих значень координат, після використання системи залишкових класів.

Нам знадобляться перші 5 чисел Фібоначчі:

1	2	3	5	8

Так як отримані за допомогою СЗК числа не перевищують 12, то представимо числа від 1 до 13 числами Фібоначчі (табл.5).

Таблиця 5 – Представлення послідовністю Фібоначчі чисел від 1 до 13

Звичайне число	Число Фібоначчі
1	11
2	011
3	0011
4	1011
5	00011
6	10011
7	01011
8	000011
9	100011
10	010011
11	001011
12	101011
13	0000011

Так як нуль ніяк не представляється за допомогою чисел Фібоначчі, то ми приймемо його за умовну «одиницю». А потім всі значення здвинемо на одне, тобто 0 будемо представляти як 1, 1 як 2 і так далі. Але пам'ятаємо, що декодер після декодування повинен буде всі числа здвинути назад.

Далі закодуємо всі точки за допомогою чисел Фібоначчі. Також представимо початкові значення координат числами Фібоначчі і порівняємо результати (табл. 6).

Таблиця 6 – Порівняння кодування числами Фібоначчі вибраних координат і кодування числами Фібоначчі результату системи залишкових класів

Координата, представлена за допомогою СЗК	Двійкове представлення координати (уже зі здвигом)	Координата	Двійкове представлення координати (без здвигу)
(6, 5, 2) (2, 2, 9) (0, 1, 5)	01011 10011 0011 0011 0011 010011 11 011 10011	951 464 980	100101000101011 00101010100011 100001010101011
(4, 3, 1) (2, 4, 5) (6, 1, 11)	00011 1011 011 0011 00011 10011 01011 011 101011	872 499 804	000010100001011 10101010010011 001001010010011
(6, 10, 12) (4, 3, 11) (4, 7, 8)	01011 001011 0000011 00011 1011 101011 00011 000011 100011	1000 102 788	0000010000000011 00000100011 000000010010011
(1, 9, 2) (6, 6, 4) (0, 6, 8)	011 010011 0011 01011 01011 00011 11 01011 100011	834 160 567	101000101010011 001001000011 10101001001011
(4, 5, 12) (6, 7, 4) (3, 3, 9)	00011 10011 0000011 01011 000011 00011 1011 1011 010011	753 524 542	010101010100011 00100000001011 00000010001011
(4, 9, 6) (0, 9, 2) (1, 10, 1)	00011 010011 01011 11 010011 0011 011 001011 011	669 119 274	101000001000011 10001010011 0101000100011
(2, 10, 6) (6, 4, 3) (4, 7, 2)	0011 001011 01011 01011 00011 1011 00011 000011 0011	604 510 249	01010010101011 01001001010011 0010010000011
(4, 6, 5) (0, 0, 0) (0, 0, 0)	00011 01011 10011 11 11 11 11 11 11	655 0 0	001010010000011 11 11
(1, 8, 2) (1, 9, 2) (6, 1, 11)	011 100011 0011 011 010011 0011 01011 011 101011	470 834 804	10100000010011 101000101010011 001001010010011
(0, 0, 7) (5, 10, 5) (0, 7, 0)	11 11 000011 10011 001011 10011 11 000011 11	462 824 546	10001010100011 010001001010011 10100010001011
(0, 0, 0) (6, 10, 12) (5, 7, 4)	111111 010110010110000011 1001100001100011	0 1000 810	11 0000010000000011 100000001010011

(1, 4, 9)	011 00011 010011	477	00101000010011
(4, 1, 5)	00011 011 10011	837	010100101010011
(6, 10, 12)	01011 001011 0000011	1000	0000010000000011

Отже, в 13 випадках використання кодів Фібоначчі після СЗК дало позитивний результат, в 6 випадках – той же результат, в інших – було гіршим в порівнянні із звичайним представленням координат числами Фібоначчі.

Тобто, на відміну від коду Хаффмана, чи арифметичного коду, чи просто двійкового представлення ми не зменшили кількість біт (при збільшенні точності) кодування, однак отримали можливість виправляти помилки.

Вирахуємо середнє значення біт на координату:

- без використання СЗК

$$C_{\Phi} = \sum_{m=2}^{16} m \cdot \frac{1}{N_{\max}}; \tag{7}$$

$$C_{\Phi} = 2\times {}^{2}\!/_{1001} + 3\times {}^{1}\!/_{1001} + 4\times {}^{2}\!/_{1001} + 5\times {}^{3}\!/_{1001} + 6\times {}^{5}\!/_{1001} +$$
$$+7\times {}^{8}\!/_{1001} + 8\times {}^{13}\!/_{1001} + 9\times {}^{21}\!/_{1001} + 10\times {}^{34}\!/_{1001} + 11\times {}^{55}\!/_{1001} +$$
$$+12\times {}^{89}\!/_{1001} + 13\times {}^{144}\!/_{1001} + 14\times {}^{233}\!/_{1001} + 15\times {}^{377}\!/_{1001} +$$
$$+16\times {}^{14}\!/_{1001} = 13{,}42158 \text{ біт/коорд.}$$

- з використанням СЗК

$$C_{\Phi} = 3 \cdot \sum_{m=2}^{7} m \cdot \frac{w}{p_{\max}}, , \tag{8}$$

де w – кількість чисел, які мають однакову відповідну розрядність (див. табл. 5).

$$C_{СЗК+Ф} = \left(7\times {}^{1}\!/_{13} + 6\times {}^{5}\!/_{13} + 5\times {}^{3}\!/_{13} + 4\times {}^{2}\!/_{13} + 3\times {}^{1}\!/_{13} + 2\times {}^{1}\!/_{13}\right)\times 3 =$$
$$= 15 \text{ біт/коорд.}$$

Збільшення ефективності кодування в порівнянні з простим двійковим представленням числа знову таки отримано не було. Але можна виділити дуже важливі факти в використанні такого алгоритму кодування:

- висока завадостійкість;
- однозначне визначення закінчення кодування кожного числа.

ВИСНОВКИ

Отже, в даній роботі були розглянуті нехарактерні для кодування зображень, відео і власне сіткових моделей методи – використання залишкових класів та чисел Фібоначчі. В чистому вигляді виграш отриманий не був, проте при розпаралелюванні кодів системи залишкових класів короткі комбінації не будуть призводити до значних помилок, а кодування числами Фібоначчі дозволило отримати можливість знаходити та виправляти помилки. Це все в значній мірі підвищує завадостійкість таких методів кодування геометрії.

ЛІТЕРАТУРА

1. Самусь Н.С. Сіткове кодування зображення за алгоритмом Edgebreaker / Н.С.Самусь, О.В. Ошаровська // Цифрові технології: Збірник. – Одеса: Одес. нац. академія зв'язку ім. О.С. Попова. – 2014. – №15. – С.119–124.

2. Samus N.S., Osharovskaya E.V., 3D image mesh entropy coding, Proceedings of the O.S. Popov ONAT, 2014'2, pp. 214–220.

3. Яцків В.В., Яцків Н.Г. Метод кодування зображень в системі залишкових класів. // Матеріали 14-ї міжнародної науково-практичної конференції «Сучасні інформаційні та електронні технології». 27–31 травня 2013. – Одеса: ОНПУ. – 2013. – С. 44–46.

4. Петришин Л.Б. Фибоначчи-подобный метод кодирования сообщений и полибоначчи способ перехода к двоичному исчислению // Вісник східноукраїнського національного університету ім. Володимира Даля: Збірник. – Луганськ: СНУ ім. В.Даля. – 2013. – №15(204). – С.158-165.

REFERENCES

1. Samus N., Osharovskaya O.V. “Image mesh coding by Edgebreaker algorithm”. Digital Technologies 15 (2014): 119-124.

2. Samus N.S., Osharovskaya E.V. “3D image mesh entropy coding”. Proceedings of the O.S. Popov ONAT, 2014’2: 214-220.

3. Yatskiv V.V., Yatskiv N.G. “The image coding method based on the residue number system”. Proceeding of the XIVth International scientific-practical conference “Modern information and electronic technologies”, 2013: 44-46.

4. Petryshyn L.B. “Fibonacci-similar method of data coding and polibonacci method transition to binary numeral system”. Visnik of the Volodymyr Dal East Ukrainian National University 15 (2013): 158-165.