

УДК 621.397:004.932

СІТКОВЕ КОДУВАННЯ ЗОБРАЖЕННЯ ЗА АЛГОРИТМОМ EDGEBREAKER

САМУСЬ Н., ОШАРОВСЬКА О.В.

Одеська національна академія зв'язку ім. О.С. Попова

IMAGE MESH CODING BY EDGEBREAKER ALGORITHM

SAMUS N., OSHAROVSKA O.V.

Odessa national academy of telecommunications n.a. A.S. Popov

Анотація. У даній статті розглядаються питання сіткового представлення 3D об'єкта та подальше його кодування за алгоритмом *edgebreaker*. Аналізується процес кодування міток вибраної області об'єкта. Обґрунтовується ефективність даного методу. Пропонується алгоритм кодування трикутної сітки 3D об'єкта. Використане одне із сучасних програмних забезпечень для створення сіткових моделей 3D об'єктів – *MeshLab*.

Abstract. In this article the questions of mesh representation of 3D object and subsequent its encoding process in the *edgebreaker* algorithm are considered. The process of encoding selected area's labels are analyzed. The efficiency of this method is justified. 3D triangular mesh encoding algorithm is proposed. One of modern software for creating 3D objects' mesh models – *MeshLab* – is used.

У загальному випадку створювана тривимірна модель об'єкта – це фізично представлений об'єкт, при проектуванні якого на відповідну координатну площину виходить фігура. Вона створює візуальне враження об'єкта, який може спочатку існувати тільки в уяві людини. Модель об'єкта повинна бути певним чином перебудована, щоб створити ефект реалістичності незалежно від точки огляду. Комп'ютер надає нам можливість, яка дозволяє безперервно змінювати тривимірну модель при зміні точки огляду таким чином, що проєкція тривимірної моделі на площину буде відповідати певним правилам [1].

Розроблено безліч програмних засобів візуального проектування. Використаємо одну із них – *MeshLab* – для побудови зображення собаки (рис. 1).

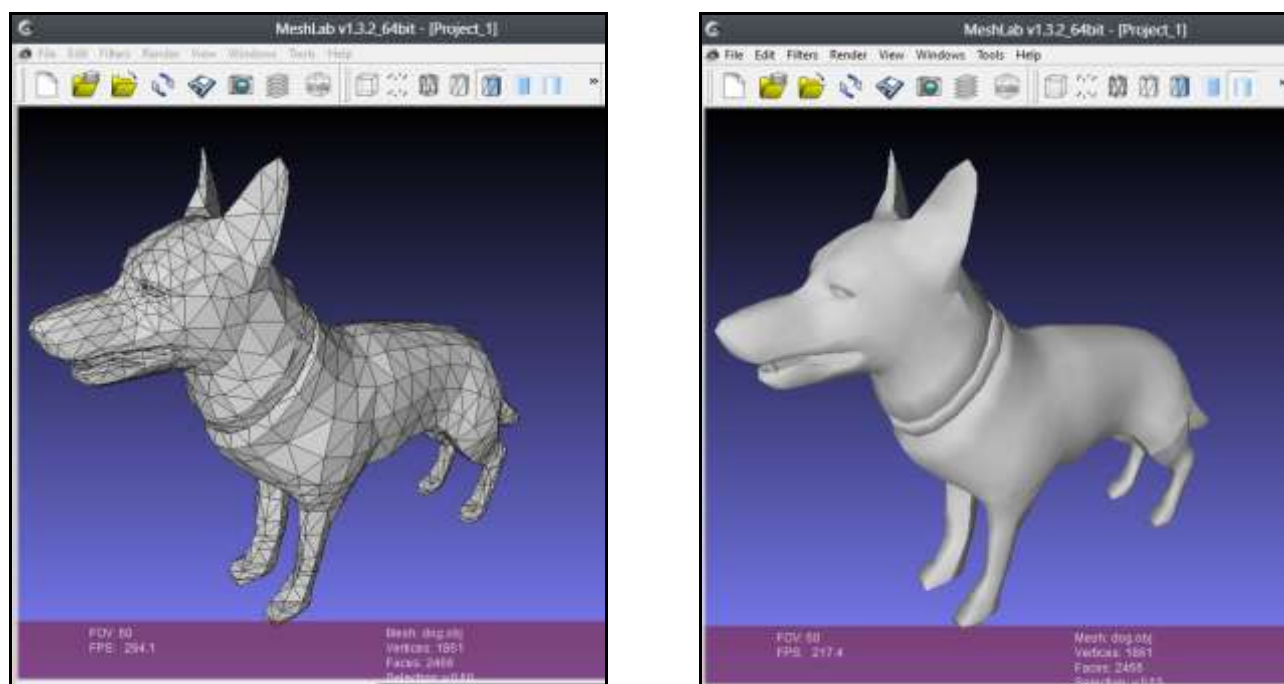


Рисунок 1 – Зображення собаки в програмі *MeshLab*: зліва – сіткове представлення; справа – отриманий результат

MeshLab – це переносима і розширювана система з відкритим вихідним кодом для обробки та редагування неструктурованих тривимірних трикутних сіток. Система сприяє обробці типових великих неструктурованих моделей, створених шляхом тривимірного сканування, і надає набір інструментів для редагування, очищення, виправлення, перевірки, візуалізації та конвертації таких сіток. MeshLab читає файли наступних форматів: PLY, STL, OFF, OBJ, 3DS, COLLADA і PTX. А також проводить запис у форматах PLY, STL, OFF, OBJ, 3DS, COLLADA, VRML і DXF.

Далі виберемо довільну ділянку для подальшої її обробки. Результат цієї дії можна побачити на рис. 2 і рис. 3.

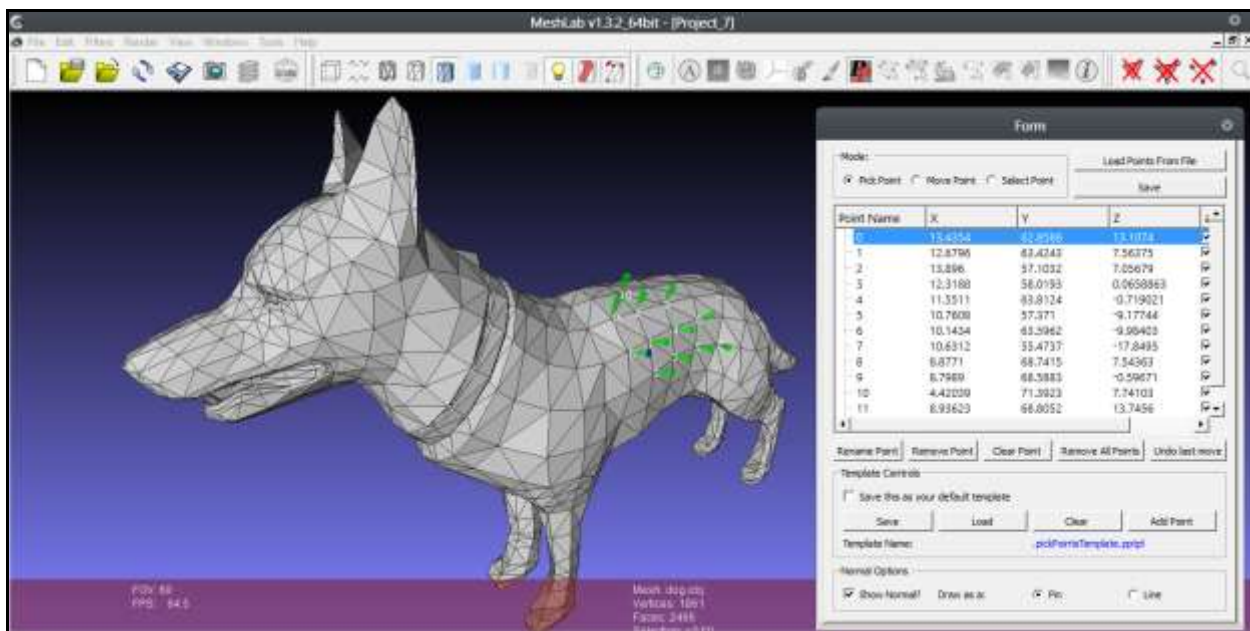


Рисунок 2 – Сіткове представлення зображення і вибрана ділянка з представленими у вікні точками в програмі MeshLab

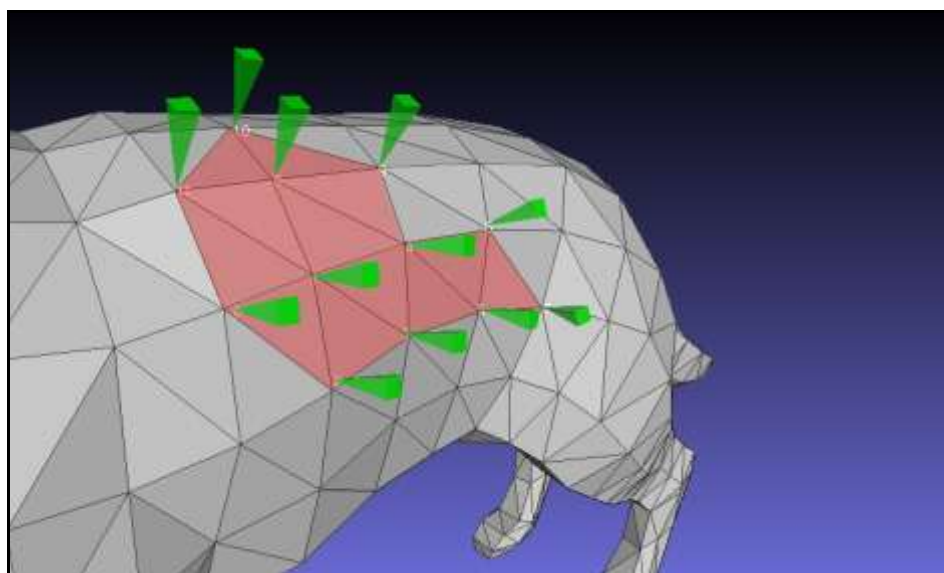


Рисунок 3 – Виділена область зображення для кодування за допомогою програми MeshLab

При побудові будь-якого тривимірного об'єкта використовується система координат. У загальному випадку це тривимірна система координат X , Y , Z . Однак у різних програмних пакетах тривимірної графіки розміщення координатних осей і відповідних координатних площин різні [2].

Оформимо в табл. 1 всі значення точок на вибраній ділянці.

Таблиця 1 – Координати вершин на вибраній області зображення

№ 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 із всіх вершин

	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.

Таблиця 3 – Пронормовані та цілі значення кожної координати

№ точки	Координата X		Координата Y		Координата Z	
	Пронормоване значення	Ціле значення	Пронормоване значення	Ціле значення	Пронормоване значення	Ціле значення
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

Для даного випадку можна використати алгоритм edgebreaker, запропонований Росігнаком [3]. Тому слід розглянути детальніше принципи, якими ми будемо користуватися при кодуванні даної вибраної області зображення.

Алгоритм edgebreaker відвідує трикутники по спіралі («в глибину») трикутного протоколу сполучного дерева і створює рядки міток – clers; одна мітка на кожен трикутник, яка вказує на декодування, тобто як може бути відновлена сітка шляхом приєднання нових трикутників до раніше відновлених.

Спочатку згадаємо 5 можливих ситуацій, розглядаючи на рис. 4 [3] стани «до» і «після» для кожної з них:

- випадок C: і справа, і зліва – не відвідані трикутники, розділення на частини не відбувається, додається мітка C в рядок clers, ітерація рухається на право;
- випадок L: зліва уже відвідано, справа – ні, додається мітка L в рядок clers, ітерація рухається на право;
- випадок R: справа уже відвідано, зліва – ні, додається мітка R в рядок clers, ітерація рухається на ліво;
- випадок S: і справа, і зліва – не відвідані трикутники, відбувається розділення на дві частини

зі спільною лише однією точкою, додається мітка S в рядок clefts, починається рекурсивний процес направо, а потім продовжується ітерація на ліво;

– випадок E: і справа, і зліва – відвідані трикутники, додається мітка E в рядок clefts, відбувається повернення з рекурсивного процесу, або отримуємо кінець стиснення.

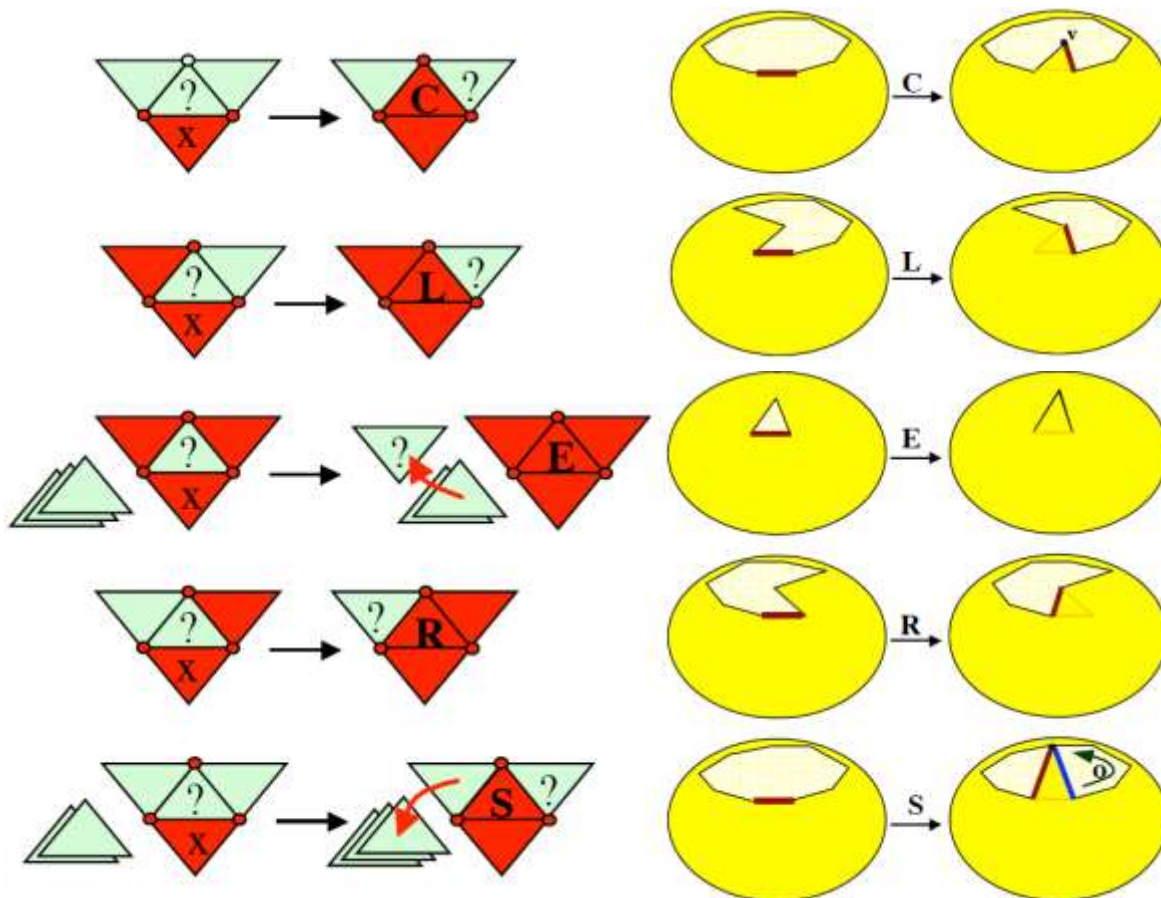


Рисунок 4 – П'ять ситуації (C, L, E, R, S) проілюстровані зверху донизу, зліва – стан “до”, справа – “після”. Нинішній трикутник відзначений “?”. Раніше відвідані трикутники і вершини червоного кольору. X знаменує трикутник, від якого рухаємося.

Далі використовуємо цей алгоритм для вибраної області зображення (рис.5).

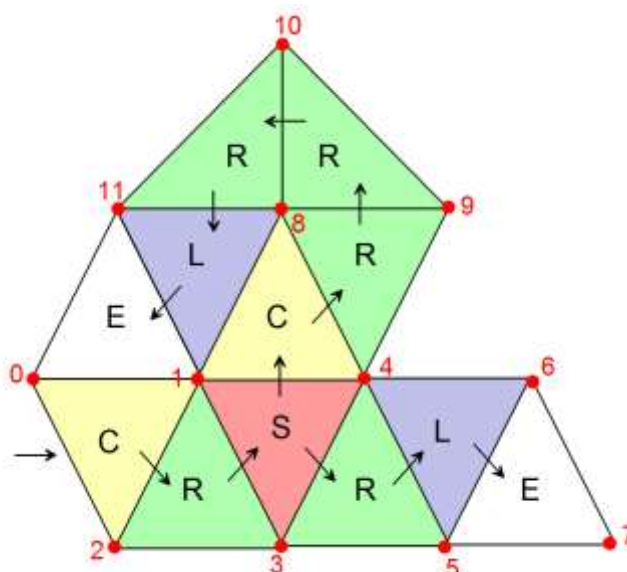


Рисунок 5 – Процес кодування вибраної області зображення

Отже, отримуємо послідовність для кодування – CRSRLECRRRLE.

Росігнак встановив, що через те що існує однозначна відповідність, за винятком перших двох вершин, між кожним C трикутником і кожною вершиною, то кількість C трикутників $v-2$. Отже, кількість не C трикутників у простій сітці становить $t-(v-2)$, що також $v-2$. Таким чином рівно половина з трикутників мають тип C . Отже, алгоритм *edgebreaker* гарантує, що стиснене уявлення зв'язності простої сітки із трикутників ніколи не перевищує $2t$ бітів, якщо для міток використовуються прості двійкові коди [3].

Їх можна отримати за допомогою використання коду Хаффмана і за умови, що ймовірність трикутників типу C така сама, як сума ймовірностей всіх інших, які у свою чергу мають однакову ймовірність (рис. 6).

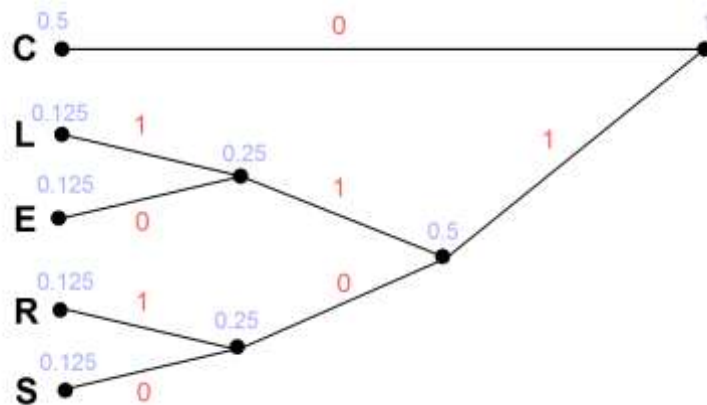


Рисунок 6 – Алгоритм побудови коду Хаффмана

Помістимо отриманий результат в табл. 4.

Таблиця 4 – Код Хаффмана

Знак (мітка), α_i	Ймовірність, ρ_i	Кодова комбінація
C	0,5	0
L	0,125	110
E	0,125	111
R	0,125	101
S	0,125	100

Далі, використовуючи отримані кодові комбінації, закодуємо отриману раніше послідовність:

CRSRLECRRRLE = 01011001011101110101101101110111

Ефективність представлення інформації цілком і повністю залежить від властивостей самої інформації. Для кількісної оцінки цих властивостей були введені інформаційні характеристики. Однією із найважливіших являється ентропія. Ентропія джерела $H(A)$ – це середня кількість інформації в одному знаку. Таким чином, середня кількість інформації визначається як математичне очікування для M_A незалежних знаків [4].

Розрахуємо ентропію нашої отриманої послідовності за формулою [4]:

$$H(A) = - \sum_{i=1}^{M_A} \rho_i \log_2 \rho_i \quad (1)$$

$$H(A) = -(0,5 \cdot \log_2 0,5 + 0,125 \cdot \log_2 0,125 + 0,125 \cdot \log_2 0,125 + 0,125 \cdot \log_2 0,125 + 0,125 \cdot \log_2 0,125) = 0,5 + 0,375 + 0,375 + 0,375 + 0,375 = 2 \text{ біта}$$

Для оцінки ефективності вибраного методу кодування вводиться поняття ефективності кодування [4]:

$$\mu = \frac{H(A)}{\bar{n}} \quad (2)$$

де \bar{n} – середня довжина кодових комбінацій і визначається як [4]:

$$\bar{n} = \sum_{i=1}^{M_A} \rho_i(\alpha_i) \cdot n_i \quad (3)$$

\bar{n}_i – кількість цифр в i -й кодовій комбінації,

$$\bar{n}_i = 1 \cdot 0,5 + 3 \cdot 0,5 + 3 \cdot 0,5 + 3 \cdot 0,5 + 3 \cdot 0,5 = 2 \text{ біта} \quad \mu = \frac{2}{2} = 1$$

Також можна розрахувати коефіцієнт стиснення в порівнянні з рівномірним кодом [4]:

$$\eta = \frac{n}{\bar{n}} \quad (4)$$

де n – мінімальне ціле число, при якому виконується нерівність $n \geq \log_2 M$

M – кількість кодових комбінацій рівномірного коду,

$$n = \log_2 12 \approx 4 \text{ біта}; \quad \eta = \frac{4}{2} = 2.$$

Проте, нас цікавить кодування не тільки міток для даної вибраної області зображення, а й кодування координат всіх її точок. А тому можна запропонувати наступний алгоритм кодування (рис. 7):



Рисунок 7 – Алгоритм кодування трикутної сітки зображення

ЛІТЕРАТУРА

1. Дж. Ли Трёхмерная графика и анимация / Дж. Ли, Б. Уэр. – [2-е изд.] – М.: Вильямс, 2002. – 640 с.
2. Гасов В.М. Трёхмерная графика в медиаиндустрии : учебник / В.М. Гасов, А.М. Цыганенко. – М. : МГУП, 2010. – 524 с.
3. J. Rossignac, 3D mesh compression, GIT – GVU Technical Report, 2003.
4. Дирда В.Ю., Івашенко П.В. Теорія електричного зв'язку. Модуль 2. Передавання інформації в телекомунікаційних системах / Навч. посіб. для бакалаврів за напрямом вищої освіти 6.050903 – Телекомунікації / В.Ю.Дирда, П.В. Івашенко – Одеса: ОНАЗ ім. О.С. Попова, 2010.