

## Висновки

Запропоновані пульсуючі інформаційні решітки дають можливість будувати високопродуктивні обчислювальні засоби, що зумовлено:

- високим ступенем паралелізму обчислень за рахунок матричної архітектури обчислювального поля;
- апаратною підтримкою реалізації алгоритму задачі, оскільки кожній арифметико-логічній операції відповідає власний процесор;
- однорідністю архітектури, що дає можливість перейти на ефективний конструктивно-технологічний рівень виготовлення – пластину.

Враховуючи сучасний рівень у розвитку мікроелектроніки, використання пульсірів дає можливість будувати дво- та тривимірні високопродуктивні обчислювальні системи різного призначення з малими габаритними розмірами.

1. Вишенчук И.М., Черкасский Н.В. Алгоритмические операционные устройства и суперЭВМ. – К.: Техника, 1990. – 197 с. 2. Кожемяко В.П., Тимченко Л.Я., Лысенко Г.Л., Кутаев Ю.Ф. Функциональные элементы и устройства оптоэлектроники. – К.: УМК ВО, 1990. – 250 с. 3. Мартинюк Т.Б., Кожемяко В.П., Павлов С.В., Заболотна Н.І. Оптоелектронні компютери. – Вінниця: ВДТУ, 1998. 4. Каляев А.В. Однородные коммутационные регистровые структуры. – М.: Сов. радио, 1978. – 336 с. 5. Кузьо М.М., Русин Б.П., Шмойлов В.І. Однородні обчислювальні середовища // Оптико-електронні інформаційно-енергетичні технології. – Вінниця, 2001. – № 2. – С. 19–37. 6. Русин Б.П., Кузьо М.Н., Шмойлов В.І. Реконфигурируемые высокопроизводительные системы на однородных средах // Автоматика и вычислительная техника. – Рига, 2000. – № 3. – С. 72–81. 7. Федотов Я.А. Проблемы интегральной электроники // Электронные компоненты. – М., 2000. – № 3. – С. 10–13. 8. Chip. – М., 2002. – № 5. – С. 76.

УДК 004.932

А.О. Мельник, О.І. Акимішин

Національний університет “Львівська політехніка”,  
кафедра електронних обчислювальних машин

## ПРОРІДЖЕННЯ ТРІАНГУЛЯЦІЙНИХ СІТОК ТРІВИМІРНИХ ОБ’ЄКТІВ КОМП’ЮТЕРНОЇ ТОМОГРАФІЇ

© Мельник А.О., Акимішин О.І., 2006

**Описано розроблений алгоритм для прорідження тріангуляційних сіток, отриманих за даними комп’ютерної томографії та досліджено його ефективність на різних типах тривимірних фігур.**

**The developed algorithm for triangle mesh decimation, created according to computed tomography data, is described and its efficiency for different types of 3-dimensional objects is analyzed.**

### Вступ

З розвитком технології отримання тривимірних моделей об’єктів (лазерні сканери, комп’ютерні томографи) все актуальнішою стає задача обробки об’ємних зображень за допомогою персональних комп’ютерів. В комп’ютерній графіці для обробки та відображення тривимірних об’єктів використовують полігональні моделі поверхонь. Як правило, такі моделі є фіксованими множинами полігонів, що забезпечують заданий рівень деталізації. Часто вони містять мільйони

полігонів, що для опису багатьох об'єктів є надлишковим. Ця задача є актуальною в комп'ютерній томографії твердих тіл, оскільки отримані зображення мають високу роздільну здатність за трьома координатами (X,Y,Z), зокрема роздільна здатність  $1024 \times 1024 \times 1024$  не є межею для сучасних томографів [1–4]. Під час відтворення об'єкта з воксельних зрізів (сканів) реальні моделі містять велику кількість даних. В зв'язку з цим виникають дві основні проблеми :

1. Як обробляти такі моделі, якщо для них не вистачає пам'яті комп'ютера;
2. Як забезпечити їх швидке відображення на екрані комп'ютера.

Друга проблема є складнішою, тому що часто додатково існує вимога роботи в реальному масштабі часу. Основний підхід для вирішення вказаних проблем полягає в побудові спрощених моделей поверхні об'єкта, які мають значно менші розміри.

### Огляд літератури

На практиці найчастіше подають полігональні сітки у вигляді трикутників, що дає можливість забезпечити точність, гнучкість та простоту представлення тривимірних даних [5]. Полігон довільної складності може бути представлений у вигляді трикутників, тому за допомогою сітки трикутників можна описати будь-яку тривимірну поверхню [6]. Існує кілька механізмів спрощення поверхонь [7], проте найчастіше використовують злиття двох і більше вершин в одну, яку надалі також можна злити з іншими вершинами, та прорідження сітки шляхом ітераційного видалення вершин з опису поверхні. Ці операції замінюють деяку групу суміжних трикутників на іншу, що покриває ту саму область, зменшуючи кількість трикутників в описі об'єкта. Механізм злиття двох вершин, що утворюють ребро (колапс ребра) триангуляційної сітки [8], дає можливість зберегти топологію моделі, простий в реалізації та прискорює обробку даних.

Відома досить велика кількість алгоритмів спрощення поверхонь [9–14]. Основні кроки цих алгоритмів можна описати так:

1. Ініціалізація – класифікація вершин.
2. Обчислення цін локальних модифікацій та створення черги виконання цих модифікацій, ключем якої є ціна локальної модифікації.
3. Поки не досягнуто заданої кількості трикутників:
  - 3.1. Вибрати з черги локальну модифікацію з найменшою ціною виконання;
  - 3.2. Виконати цю модифікацію;
  - 3.3. Перерахувати ціну та оновити чергу для всіх вершин в околі вершини, яку було видалено в результаті виконання локальної модифікації.

Відмінними для цих алгоритмів є: структура даних для представлення моделі в пам'яті комп'ютера, тип та спосіб обчислення ціни локальних модифікацій. Вибір структури для представлення триангуляції впливає на обчислювальну складність алгоритмів та швидкість конкретної реалізації [5]. Тип та спосіб обчислення ціни локальної модифікації впливають на швидкість виконання алгоритму та якість отриманої поверхні після оптимізації. Тому при проектуванні алгоритму три вказані параметри відіграють найважливішу роль.

Недоліком існуючих алгоритмів спрощення поверхонь є зменшення трикутників в описі моделі до наперед заданого значення, що є недопустимо під час автоматичного відтворення реальних об'єктів, наприклад, під час подальшої обробки та дослідження в CAD-системах. У цьому випадку критерієм проведення оптимізації повинна бути допустима величина відхилення спрощеного об'єкта від початкового. За такого підходу з опису поверхні видаляють всі вершини, що задовольняють вказаний критерій разом з інцидентними їм трикутниками, причому буде забезпечено потрібну точність представлення об'єкта.

### Мета досліджень

Ціллю роботи є розроблення та реалізація алгоритму автоматичного процесу прорідження триангуляційних сіток, якими описуються поверхні 3-вимірних об'єктів, до заданого рівня точності, тобто зменшення кількості трикутників в описі поверхні без втрати інформації про об'єкт або із втратою, яка відповідає заданій точності.

Маючи триангуляцію  $M_0$ , що містить  $N$  точок, отримати триангуляцію  $M_n$  таку, що:

- вона має відхилення  $d$  від  $M_0$  не більше ніж задана точність  $\varepsilon$   $d(M_0, M_n) < \varepsilon$ ,
- вона має мінімальну кількість вузлів (точок)  $n$ ,  $n = \min n(M_i)$ .

### Опис запропонованого алгоритму

Існує кілька способів задання триангуляційного опису поверхні, які відрізняються структурою даних для представлення сітки трикутників. Крім того, вибір структури залежить від мети подальшого використання триангуляції. Однією із найзручніших структур даних є *кільцева структура типу "Вузли і трикутники"*, яку можна описати так:

1) полігональну модель поверхні ділять на дві множини: множину точок та множину трикутників. Кожній окремій точці та трикутнику відповідає унікальний ідентифікатор – ціле число. Тоді множину точок та множину трикутників представляють динамічними масивами, де їх ідентифікатори є індексами цих масивів;

2) для кожної точки задають її координати в просторі  $(x, y, z)$  та список вказівників на інцидентні вершини трикутники. Трикутник представляється вказівниками на три точки, що є його вершинами (рис. 1).

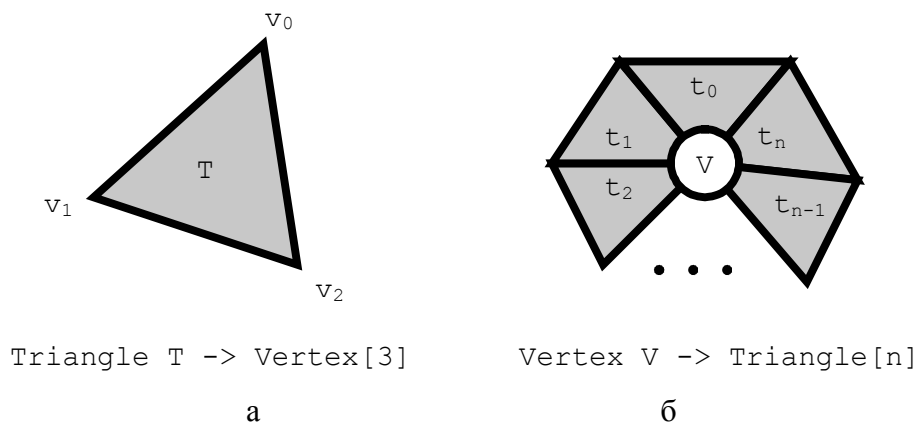


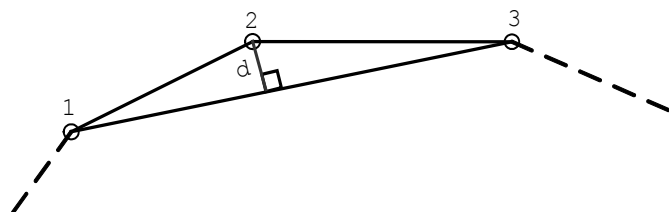
Рис. 1. Графічне представлення елементів триангуляції:

а – трикутник та інцидентні йому вершини; б – вершина та інцидентні їй трикутники

Вхідними даними для алгоритму є триангуляційний опис об'єкта та задана точність. Оскільки у разі відтворення об'єкта з воксельних зрізів (сканів) відстань між сусідніми точками дорівнює одиниці, то описаний об'єкт отримують сформованим з великої кількості трикутників розміром в один піксель. У такому випадку опис площини, наприклад, грані куба із розміром скану  $1024 \times 1024$  міститиме 1 Мб точок та 2 Мб трикутників. Для представлення такої площини достатньо 4 вершини (точки) та 2 трикутники.

#### Означення похибки на площині

Розглянемо випадок обчислення похибки для точки 2 на площині. Видаляючи точку 2 з опису моделі, точки 1 та 3 з'єднують прямою, як зображено на рис. 1. Якщо відстань від точки 2 до цієї прямої задовольняє рівність  $d \leq \varepsilon$  (1), то точку 2 можна видалити з триангуляції.



## Означення похибки в тривимірному просторі

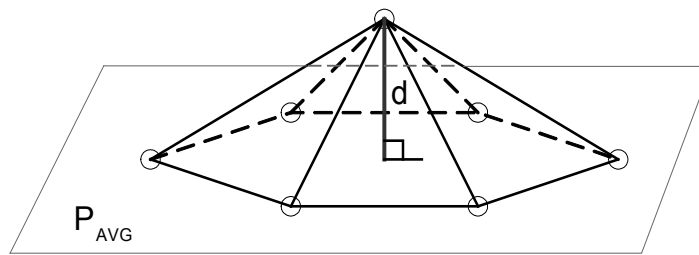


Рис. 2. Означення похибки в тривимірному просторі

У тривимірному просторі (рис. 2) похибкою для точки  $v_i$  вважатимемо відстань від цієї точки до площини  $P_{AVG}$  – середньої площини для точок, що лежать в околі точки  $v_i$ .

Визначаючи ціну видалення точки з триангуляції, перевіряють рівності (1). Якщо ця рівність виконується, то цю точку видаляють разом з інцидентними їй трикутниками.

### Базова операція спрощення триангуляції

Як базову операцію спрощення триангуляції (рис. 3, а) використовують операцію злиття вершин або об'єднання пар точок  $(v_i, v_j) \rightarrow v'$ . Цю операцію виконують у такій послідовності:

1. Вершини  $v_i$  та  $v_j$  переміщують на позицію  $v'$ .
2. Інцидентні обом вершинам трикутники видаляють з моделі.
3. Решту трикутників позначають як інцидентні вершині  $v_i$ .
4. Вершину  $v_j$  видаляють з моделі.

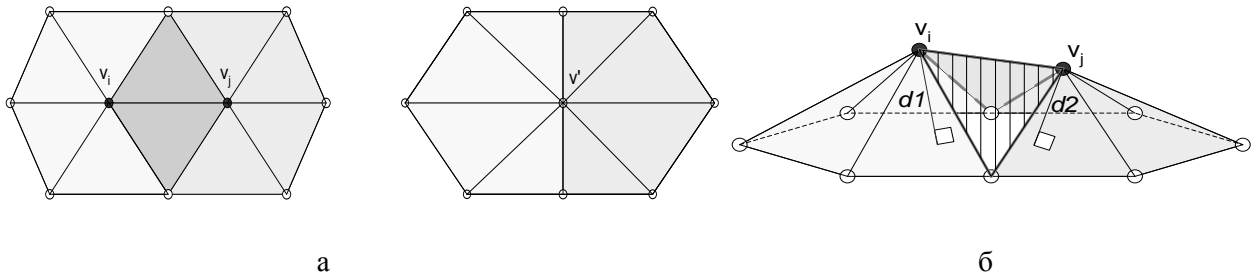


Рис. 3. Базова операція спрощення триангуляції (а) та означення похибки відносно базової операції спрощення опису об'єкта (б)

Внаслідок виконання цієї операції з моделі видаляють одну вершину та трикутники, спільні для обох вершин.

Розроблений алгоритм оптимізації оперує ребрами триангуляції. Для точок, що формують кожне ребро, обчислюються відстані до  $P_{AVG}$ . Ребра заносяться в чергу на видалення. Ключем черги є  $e_{min} = d1 + d2$ .

### Алгоритм прорідження триангуляційної сітки

1. Вибір всіх пар точок  $(v_i, v_j)$  таких, що  $(v_i, v_j)$  є ребром триангуляційного опису поверхні.
2. Для кожної вершини обчислюють відстань до середньої площини.
3. Для кожної пари  $(v_i, v_j)$  :
  - 3.1. Вибирають позицію  $v'$ ;
  - 3.2. Заносять пари в чергу, ключем якої є  $e_{min}$ ;
4. Повторювати, поки не досягнуто заданої точності:
  - 4.1. З черги вибирається пара  $(v_i, v_j)$  з мінімальною ціною;

- 4.2. Виконують операцію  $(v_i, v_j) \rightarrow v'$ ;
- 4.3. Для всіх пар  $(v_i, v_k)$ , що залишились, виконують крок 3;
- 4.4. Оновлюють чергу.

### Результати виконання алгоритму

Оскільки межею прорідження опису об'єктів у цьому випадку є забезпечення заданої точності, а не кількість трикутників, що повинні залишитись після спрощення опису поверхні, то очевидним є те, що кращими будуть результати на плоских фігурах, ніж на випуклих. Тому необхідним є відпрацювання алгоритму на різних типах фігур для визначення ефективності його роботи та доведення доцільності його використання. На рис. 5 зображено кілька типів фігур з приблизно однаковими кількостями трикутників. У табл. 1 наведено характеристики цих об'єктів до прорідження їхнього опису. В табл. 2 вказано часові параметри обробки тестових поверхонь, зокрема з розділенням часових затрат на кожному з етапів обробки даних. Оскільки тестові зображення мають значно менший розмір ніж реальні, то для роботи з реальними зображеннями необхідні значно більші часові затрати. З табл. 2 видно, що найбільш трудомісткою є задача прорідження сітки, крім того витрати часу на етапах вводу/виводу даних можна істотно скоротити за умови переходу від текстового формату даних до бінарного. Опрацьовані фігури (рис. 6) повністю зберігають інформацію про об'єкт, проте описуються іншою сіткою трикутників, характеристики якої наведено у табл. 3. Порівнюючи їх з характеристиками вхідних даних (табл. 1), отримуємо відношення між вхідними та вихідними тріангуляціями (табл. 4), на основі чого можемо зробити висновки про доцільність виконання прорідження сіток та ефективність розробленого алгоритму для різних типів фігур. Краща ефективність на плоских фігурах обумовлена наявністю більшої кількості точок, які задовольняють заданій точності. Наприклад, для опису поверхні правильного куба (об'єкт 2) сіткою трикутників достатньо вісім вершин та дванадцять трикутників, тоді як для опису поверхні кулі ця кількість елементів є недостатньою.

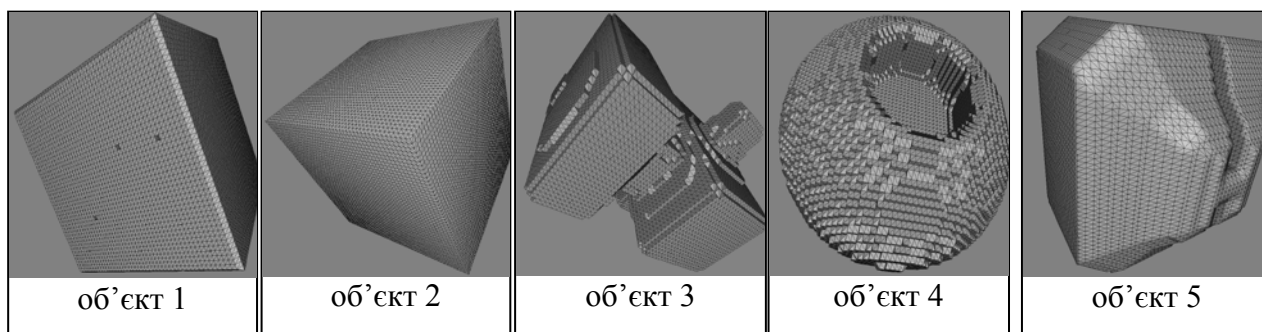


Рис. 5. Об'єкти перед обробкою

Таблиця 1

#### Характеристики об'єктів перед обробкою

Об'єкти	Об'єм пам'яті	Кількість трикутників	Кількість вершин	Кількість ребер	Розмір файла, Kb	Розмір * STL, Mb
Об'єкт 1	814,19	27796	13894	39513	670	3,21
Об'єкт 2	776,67	26508	13256	39750	643	3,09
Об'єкт 3	625,42	20840	10426	24819	498	2,48
Об'єкт 4	804,79	27468	13736	13399	671	3,29
Об'єкт 5	791376	26372	13192	32042	643	3,22

\* STL – стандартний формат для зберігання об'єктів, описаних сітками трикутників та їх імпорту у пакети обробки тривимірних зображень, наприклад 3D Studio Max.

## Часові затрати для обробки об'єктів

Об'єкти	Вчитка об'єкта, с	Ініціалізація, с	Обробка, с	Вивід у файл, с	Загальний, с
Об'єкт 1	1,125	3,422	8,45	0,02	13,02
Об'єкт 2	1,03	3,375	8,265	0,02	12,69
Об'єкт 3	0,84	2,61	5,48	0,15	9,09
Об'єкт 4	1,125	3,625	3,48	0,73	8,97
Об'єкт 5	1,08	3,30	7,28	0,14	11,80

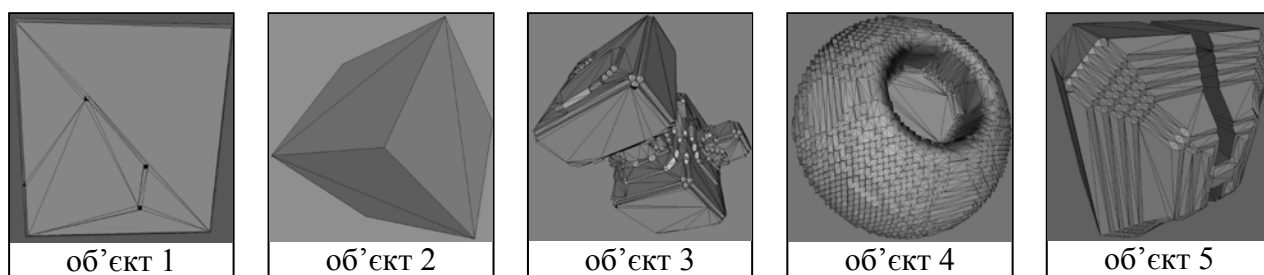


Рис. 6. Об'єкти після відпрацювання

## Характеристики об'єктів після обробки

Об'єкти	Об'єм пам'яті	Кількість трикутників	Кількість вершин	Розмір файла, Кб	Розмір STL, Мб
Об'єкт 1	2,93	104	48	2	0,013
Об'єкт 2	0,42	12	8	1	0,002
Об'єкт 3	92,32	3144	1578	70	0,393
Об'єкт 4	467,47	15906	7995	374	1,964
Об'єкт 5	82,13	2796	1404	61	0,366

Для оцінки алгоритму можна порівняти розміри вхідних файлів з вихідними та початкову кількість вершин та трикутників з отриманою після обробки даних. Результати порівняння окремо для кожної з фігур наведено в табл. 4. В графі “кількість трикутників” та “кількість вершин” вказано початкове та кінцеве значення елементів триангуляції. В графі “% ефективності” наведено процентне зменшення кількості трикутників між початковою та кінцевою триангуляціями для кожного об'єкта.

## Оцінка ефективності алгоритму для різних об'єктів

Об'єкти	Розмір вх. файла, Кб	Розмір вих. файлу, Кб	Кількість трикутників	Кількість вершин	% ефективності
Об'єкт 1	670	2	27796/104	13894/48	≈ 99
Об'єкт 2	643	1	26508/12	13256/8	≈ 99,9
Об'єкт 3	498	70	20840/3144	10426/1578	≈ 85
Об'єкт 4	671	374	27468/15906	13736/7995	≈ 43
Об'єкт 5	643	61	26372/2796	13192/1404	≈ 89

## Висновки

Розроблено алгоритм прорідження триангуляційних сіток, що використовуються для опису поверхонь тривимірних об'єктів, відповідно до заданої точності. Програмну реалізацію описаного

алгоритму виконано мовою програмування C++. Оскільки алгоритм обробляє дані до заданого рівня точності, то було проведено тестування для визначення ефективності його виконання на різних типах об'ємних фігур. Моделі фігур було описано рівномірними триангуляційними сітками з приблизно однаковими кількостями трикутників. Після відпрацювання алгоритму було порівняно початкову та кінцеву кількість трикутників для різних типів фігур, на основі чого отримано: мінімальний відсоток прорідження триангуляції на випуклій фігурі становить 43 % (об'єкт 4), для плоских правильних фігур 99 % (об'єкти 1–2) та для фігур, що містять як плоскі, так і випуклі ділянки поверхні 89 % (об'єкт 5). Для визначення середнього значення ефективності за різними типами фігур потрібно виконати  $(99 + 85 + 43 + 89)/4 = 79 \%$ , що є достатньо високим показником. Надалі планується скоротити часові затрати на обробку даних та дослідити витрати пам'яті та процесорного часу на різних етапах обробки для реальних об'ємних зображень.

1. HYTEC Sensors & Imaging Group (HYSIG) FlashCT™ Industrial Computed Tomography [www.hytecinc.com](http://www.hytecinc.com). 2. BIR, Inc. 2006 [www.bio-imaging.com](http://www.bio-imaging.com). 3. Toshiba IT & Control systems corporation, Industrial X-ray CT TOSCANER-20000AV Series, [www.toshiba-itc.com/cat](http://www.toshiba-itc.com/cat). 4. Dr. Hartmut Runge Computer tomography exposes electronic products, Siemens AG Corporate Communications Press department, Munich, 11 November 2002. 5. Скворцов А.В. Триангуляция Делоне и её применение. – Томск: Изд-во Том. ун-та, 2002. – 128 с. 6. Игнатенко А. Геометрическое моделирование сплошных тел. [www.graphics.cs.msu.ru](http://www.graphics.cs.msu.ru). 7. Luebke, David P. A Developer's Survey of Polygonal Simplification Algorithms // IEEE Computer Graphics & Applications, 2001. 8. Hoppe H. Progressive meshes. Proceedings of SIGGRAPH96, pages 99–108, August 1996. ISBN 0-201-94800-1. 9. Garland M. and Heckbert P.S. Surface simplification using quadric error metrics, Proceedings of SIGGRAPH 97, pages 209–216, August 1997. ISBN 0-89791-896-7. Held in Los Angeles, California. 10. Heckbert P. and Garland M., editors. Surface Simplification, chapter Survey of Polygonal Surface Simplification Algorithms. Course notes of Siggraph 97. ACM SIGGRAPH, July 1997. 11. Hoppe H. Efficient Implementation of Progressive Meshes., Computer & Graphics. – 1998. – Vol. 22, No. 1. – P. 27–36. 12. Schroeder W.J., Zarge J.A. and Lorensen W.E. Decimation of triangle meshes. Computer Graphics (SIGGRAPH '92 Proc.), 26(2): 65–70, July 1992. 13. Hussain M., Okada Y., Nijima K. Efficient and Feature-Preserving Triangular Mesh Decimation Journal of WSCG, Vol. 12, No. 1–3., ISSN 1213-6972 WSCG'2004, February 2–6, 2004, Plzen, Czech Republic. 14. Hussain, M., Okada, Y. and Nijima, K. Fast, simple, feature-preserving and memory efficient simplification of triangle meshes. International Journal of Image and Graphics, 3(4): 1–18, 2003.