

МЕТОД СИНТЕЗУ БЕЗШОВНОЇ ТЕКСТУРОВАНОЇ СФЕРИ ДЛЯ ВІЗУАЛІЗАЦІЇ У ГЕОІНФОРМАЦІЙНИХ ТА КАРТОГРАФІЧНИХ СИСТЕМАХ

Анотація. У статті розглянуто метод створення безшовної текстурованої моделі сфери та видалення артефактів пов'язаних з особливостями розрахунку текстурних координат при циліндричному проєціюванні. **Метою** статті є розробка методу побудови та візуалізації сферичних текстурованих об'єктів який би не мав недоліків існуючих методів, які пов'язані з виникненням наочно помітних візуальних артефактів при накладанні текстури на поверхню сфери. Наведено метод який дозволяє коректувати текстурні координати як на геометричному рівні, так і в процесі проведення візуалізації, в якому сфера умовно розділяється на частини відповідно до відстані від полюсів. Для кожної частини використовується найбільш оптимальний алгоритм текстурування, що забезпечує швидкість методу. Обчислення за допомогою запропонованих алгоритмів можна повністю реалізувати за рахунок ресурсів GPU. За **результатами** дослідження визначено, що запропонований метод дозволяє видалити артефакти які виникають при розрахунку текстурних координат для циліндричного проєціювання як безпосередньо у процесі візуалізації, так і на етапі синтезу моделі сфери, що дозволяє значно покращити відображення текстур на поверхні сфери.

Ключові слова: теселяція, поділ рівної площини, GPU, MIP рівні, реалістичні візуальні сцени, текстурування.

Вступ

Картографічна проєкція - це систематичне перетворення широт і довгот місцевостей з поверхні сфери або еліпсоїда в місця на площині [1]. Будь які карти неможливо створити без застосування картографічних проєкцій. Проте поверхня сфери не може бути представлена на площині без спотворень, те саме стосується і інших опорних поверхонь таких як сплюснені сфероїди, еліпсоїди та геоїди. Для спрощення в цій статті припускається, що поверхня, яку потрібно відобразити, є сферою. Оскільки будь-яка картографічна проєкція є зображенням однієї з опорних поверхонь на площині, усі картографічні проєкції спотворюються, але кожна спотворюється певним чином. Залежно від призначення карти одні спотворення допустимі, а інші ні; тому існують різні картографічні проєкції, які зберігають одні властивості кулеподібного тіла за рахунок інших властивостей. У більш загальному плані поверхні планетних тіл можуть бути нанесені на карту, навіть якщо вони надто неправильні, щоб змоделювати їх за допомогою сфери чи еліпсоїда. Карти можуть бути кориснішими, ніж глобуси, у багатьох ситуаціях: вони компактніші та легші для зберігання, легко вміщують величезний діапазон масштабів та переглядаються на дисплеях комп'ютерів, можуть полегшити вимірювання властивостей регіону, що нанесено на карту. Також карти можуть показувати більші частини поверхні Землі одночасно. Але з розвитком комп'ютерної техніки, графічних прискорювачів та появою великої кількості різноманітних програм, зокрема геоінформаційних та картографічних систем, виникає і зворотна задача - як, маючи карту з відповідними спотвореннями, побудувати тривимірну модель відповідного тіла з якого ця карта була створена, наприклад модель земної кулі, або тіл сонячної системи, таким чином щоб мінімізувати кількість артефактів які виникають при модулюванні та візуалізації моделі.

Метою статті є розробка методу побудови та візуалізації сферичних текстурованих об'єктів який

би не мав недоліків існуючих методів, які пов'язані з виникненням наочно помітних візуальних артефактів при накладанні текстури на поверхню сфери. Процес візуалізації тривимірної моделі сфери може бути розділено на два етапи: побудова геометричної моделі та налагодження текстури на поверхню моделі. Проаналізуємо послідовно існуючі підходи та проблеми які виникають на кожному етапі.

Побудова моделі сфери

Існує декілька методів побудови моделі сфери. Найбільш простіший з них, це теселяція (розбиття на трикутники) за широтою та довготою, але вона має відчутні артефакти на полюсах, та нерівномірну щільність трикутників. Інші методи, такі як геодезична теселяція, не містять цих недоліків. Серед них можна виділити дві групи методів: теселяція опуклого багатогранника (наприклад, ікосаедра) [2] та згладжування куба [3].

У першому методі, спочатку будується модель ікосаедру (тетраедру або октаедру) з трикутників, а далі кожен трикутник моделі додатково розділяється на менші трикутники, з відповідною нормалізацією координат нових вершин. Процес повторюється до досягнення заданої щільності сітки трикутників [4].

Інший метод, який базується на розділенні куба, працює схожим чином, але в якості базової сітки обирається більш проста фігура - куб, кожна грань якого теселюється окремо і незалежно від інших. Це дозволяє спростити та прискорити процес отримання нових вершин, використавши паралельні обчислювачі, кожен з яких буде опрацьовувати відповідну грань.

Процес отримання координат вершини **P** виглядає наступним чином:

$$\mathbf{P} = \mathbf{N} + s(i\mathbf{T} + j\mathbf{B}), \quad (1)$$

де **N** - нормаль до поверхні грані, $s = 1/n$, де n - кількість розділень, i та j - локальні координати на грані (від 0 до $n - 1$), **T** - дотична до грані, $\mathbf{B} = \mathbf{N} \otimes \mathbf{T}$ - бінормаль (\otimes - векторний добуток).

Після розрахунку вектору \mathbf{P} , отримане значення нормалізується і запам'ятовується як нормаль у вершині, а після масштабування висотою зберігається як актуальна позиція вершини. Наступною задачею є отримання сітки трикутників. Вона також може бути отримана незалежно для кожної грані. Сітка трикутників будується шляхом формування чотирикутників з вершин отриманих за (1) з їх подальшою тесесяцією, в залежності від того до якого квадранту грані вони належать: другий та четвертий квадранти тесесяються в прямому напрямку, а перший та третій

в зворотному. Індекси вершин чотирикутника a, b, c, d отримуються наступним чином

$$a = [f(n + 1) + j](n + 1) + i, b = a + 1,$$

$$c = [f(n + 1) + j + 1](n + 1) + i, d = c + 1, \quad (2)$$

де f - номер грані, i та j - координати чотирикутника на грані (від 0 до $n - 1$).

Для прямого напрямку будуть сформовані трикутники: (a, b, c) та (a, c, d) ; для зворотного (a, b, d) та (b, c, d) . На рис. 1 показана відповідна сітка трикутників, яка отримана за наведеним методом.

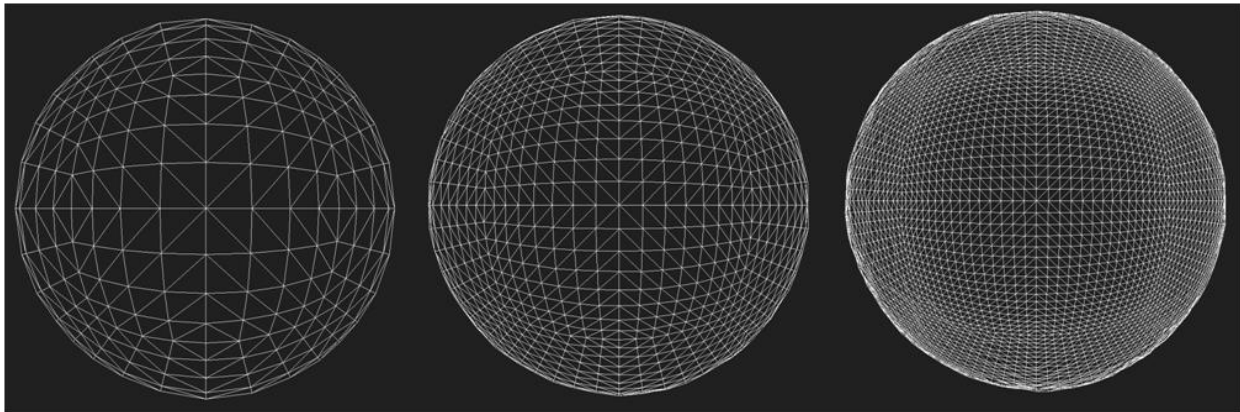


Рис. 1. Сітка трикутників як отримана після тесесяції та нормалізації куба, зліва направо, для 8, 16 та 32 розділень, відповідно

Обчислення текстурних координат

Важливою частиною для будь якої системи візуалізації є накладання матеріалів на поверхню моделей об'єктів, що дозволяє придати їм необхідний (реалістичний вигляд). Цей процес на практиці виконується за допомогою накладання текстур, для чого потрібно обчислити текстурні координати для кожної вершини моделі.

Якщо сфера використовується як основа для якогось тіла сонячної системи, то доцільно застосувати якусь картографічну проекцію [1, 5], для того,

щоб можна було нанести існуючі картографічні зображення на модель сфери. Найбільш відомою є циліндрична проекція Ламберта, яка є основою сімейства циліндричних рівноплощинних проекцій. У цій проекції екватор обрано як паралель без спотворень, таким чином, помноживши висоту проекції на певний коефіцієнт і поділивши ширину на той самий коефіцієнт, області відсутності спотворень можна перемістити до будь-якої бажаної пари паралелей на північ і південь від екватора. На рис. 2 наведені спотворення які надає циліндрична рівноплощинна проекція Ламберта.

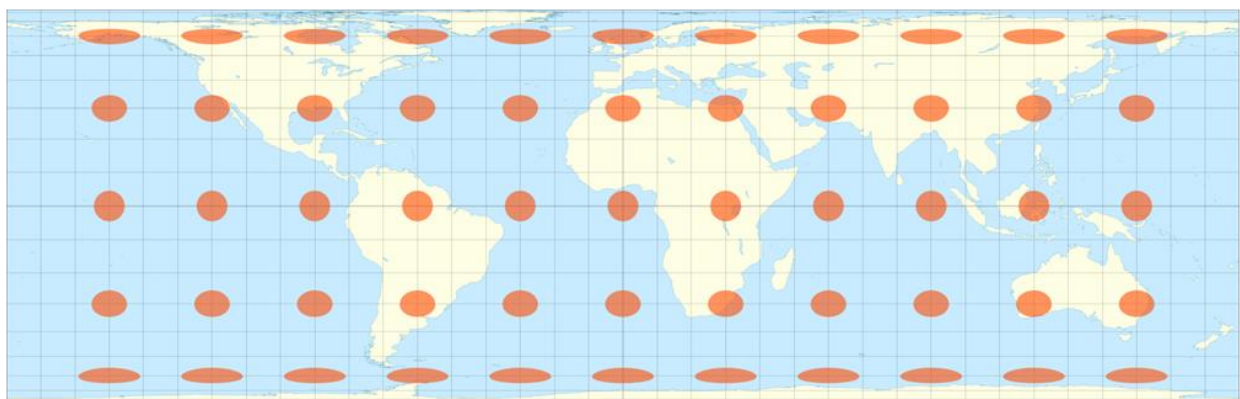


Рис. 2. Циліндрична рівноплощинна проекція Ламберта з індикатрисою деформації Тіссо (© Eric Gaba)

У картографії індикатриса Тіссо [6] є результатом проєціювання кола нескінченно малого радіуса з вигнутої геометричної моделі, такої як глобус, на карту. Одна індикатриса описує спотворення в якійсь одній точці, але оскільки викривлення різняться

на площині карти, зазвичай індикатриса Тіссо розміщуються по всій карті, щоб проілюструвати просторову зміну викривлення. Оскільки всі еліпси на карті мають одну й ту саму площу, спотворення яке внесене проекцією карти, є очевидним.

На рис. 3 показано процес формування відповідного зображення для циліндричної проекції.

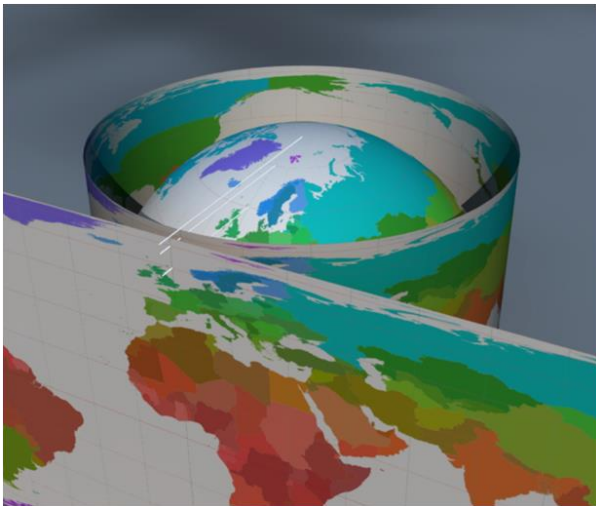


Рис. 3. Циліндрична проекція Землі (Public Domain)

Однією з найбільш поширених видів циліндричних проекцій є циліндрична рівнопрямокутна проекція. Така проекція відображає меридіани на вертикальні прямі лінії постійного інтервалу, а широти на горизонтальні прямі лінії постійного інтервалу. Така проекція не використовується в навігації чи кадастровому картографуванні, але, через простий зв'язок між положенням пікселя зображення на карті та його відповідним географічним розташуванням на Землі чи інших сферичних тілах Сонячної системи, знаходить своє використання в картографуванні. Вона є стандартом для глобальних наборів растрових даних: Celestia, NASA World Wind, USGS Astrogeology Research Program і Natural Earth. Крім того, така проекція часто використовується у панорамній фотографії та відеозйомці для представлення сферичного панорамного зображення [7], а також для формування зображень у віртуальних середовищах, які складаються з усюди направлених зображень, що відображалися у віртуальній реальності [8]. Для того щоб розрахувати відповідні текстурні координати для рівноплощинної проекції Ламберта для кожної вершини, необхідно використати наступні вирази:

$$\begin{aligned} \mathbf{N}_1 &= \|\mathbf{P} - \mathbf{P}_0\|, \\ u &= 0,5 + \frac{\text{atan2}(x_n, z_n)}{2\pi}, \\ v &= 0,5 - \frac{\arcsin(y_n)}{\pi}, \end{aligned} \quad (3)$$

де u та v - відповідні текстурні координати, $\mathbf{N}_1 = (x_n, y_n, z_n)$ - напрям від центру сфери \mathbf{P}_0 на точку \mathbf{P} на поверхні сфери, координати якої обчислені відповідно до (1), функція $\text{atan2}(x_n, z_n)$ обчислює значення арктангенсу для x_n/z_n в діапазоні від $-\pi$ до π , $\|\cdot\|$ - операція нормалізації векторів.

Для рівнопрямокутної проекції текстурна координата по вертикалі обчислюється як

$$v = 0,5 - 0,5y_n. \quad (4)$$

Отже, маючи список вершин сфери, можна обчислити текстурні координати для будь якої з них.

Видалення прошарків які виникають при налагодженні текстур на модель сфери

При налагодженні текстур виникає дві завади, які породжують прошарки на моделі сфери. Перша завада виникає при перетині горизонтальною текстурною координатою межі між 1 та 0. Фактично вираз (3) для u дає дробову частину від координати, тому на стику двох граней остання вертикальна смуга прямокутників буде мати текстурні координати які пов'язують лівий та правий кути текстури і фактично вся текстура в зворотному напрямку буде накладатися на трикутники які входять до цієї смуги (рис. 4).



Рис. 4. Смуга чотирикутників з текстурними координатами в зворотному напрямку: a - на моделі Землі, b - на хмарному покриві

Для того щоб прибрати цю заваду, потрібно при теселяції чотирикутника аналізувати горизонтальну складову його текстурних координат, і якщо ця складова для будь якої вершини буде близької до 0, а для протилежної - близької до 1, потрібно додати до меншої 1, та продублювати відповідну вершину з новою текстурною координатою.

Це дозволяє повністю прибрати спотворення текстури показані на рис. 4.

Ще один тип візуальних артефактів виникає в полярних областях і обумовлений тим як будується проекція та інтерполяцією текстурних координат вздовж трикутника (рис. 5).

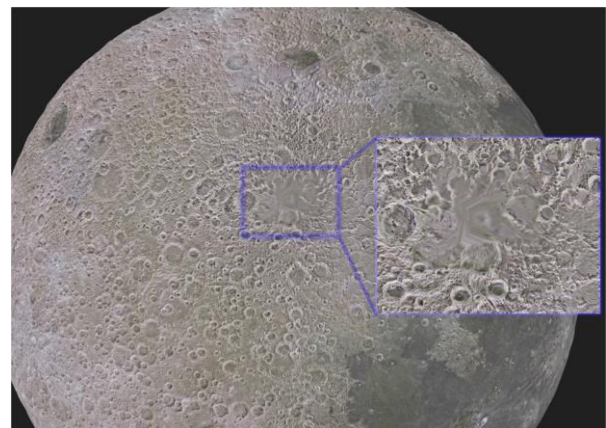


Рис. 5. Візуальні артефакти у полярних областях моделі Місяця, які обумовлені інтерполяцією текстурних координат

Відомі декілька методів як прибрати ці артефакти, найбільш простіший варіант - на полюсах моделі зробити спеціальну геометрію з іншим типом проєціювання, але такий варіант потребує окремої текстури та геометрії на моделі для полярних областей, та вирішення проблем з плавним переходом основної моделі у допоміжну, що непрактично.

Інший підхід для того щоб прибрати такі артефакти - обчислення текстурних координат у фрагментній мікропрограмі, безпосередньо в процесі візуалізації. Але це, в свою чергу, породжує інший тип артефактів, пов'язаний з тим як графічний процесор розраховує MIP рівень текстури для кожного фрагменту зображення - вертикальний прошарок завдовжки в один піксель (рис. 6). А полярні області, хоч і стають більш гладкими порівняно з попереднім (рис. 5), але все ще містять відчутні артефакти проєціювання (рис. 7). Таким чином потрібно розробити метод, який би покращив якість накладання текстур на модель сфери, який би не мав перелічених завад.

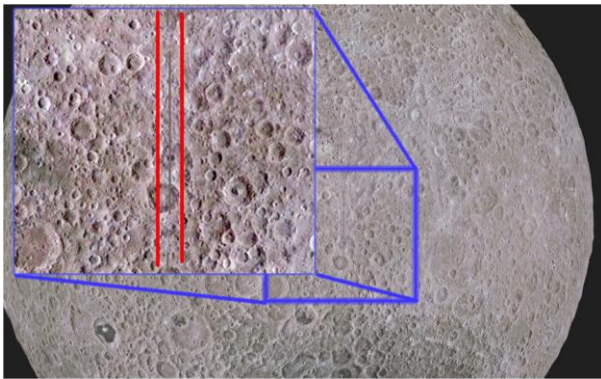


Рис. 6. Вертикальний прошарок пов'язаний з алгоритмом обрання MIP рівня текстури



Рис. 7. Артефакти проєціювання при розрахунку текстурних координат для кожного фрагменту

Проаналізуємо причини появи зазначених артефактів. Обидва типи артефактів виникають тому, що графічні процесори обчислюють MIP рівень текстури за так званими частковими похідними екранного про-

стору - величини зміни значення текстурної координати (u, v) від довільного пікселя до пікселя поруч із ним, вище чи нижче. Графічні процесори зазвичай обчислюють це значення для кожного набору пікселів 2×2 , тому MIP рівень визначається тим, наскільки (u, v) змінюється з кожним із цих наборів. І коли обчислюється горизонтальна складова текстурних координат u для моделі сфери згідно з (3), на межі значення u значно змінюється між двома пікселями. Через це графічний процесор вважає, що вся текстура відображається між цими двома пікселями, і при розрахунку MIP рівня використовується найменший можливий рівень. Це в свою чергу веде до візуального артефакту. Координата v такого стрибка не робить ні при обчисленні згідно з (3), ні (4), тому в горизонтальній площині артефакту не виникає. Для того щоб прибрати цей артефакт дослідниками [9, 10] пропонується, або виділити ті пікселі для яких виникає артефакт і примусово змінити їх текстурні координати, або примусово розраховувати MIP рівень використовуючи отримані координати з (3), (4).

Всі перелічені методи мають декілька недоліків, серед яких:

- 1) Додаткові складні обчислення, що зменшують швидкість процесу візуалізації.
- 2) Залежність від обладнання та можливостей GPU.
- 3) Необхідність використання спеціальних інструкцій запиту до текстур із зазначенням часткових похідних.

В статті пропонується дещо інший підхід до вирішення цієї задачі. Так як артефакти, пов'язані з інтерполяцією текстурних координат, виникають фактично лише в полярних областях. При цьому, у стандартному запиті даних з текстури (коли MIP рівні розраховуються автоматично за текстурними координатами що інтерполюються) артефактів наведених на рис.6 не виникає. Доцільно розраховувати текстурні координати у фрагментній мікропрограмі, безпосередньо в процесі візуалізації лише для полярних областей (наприклад, більших за 95%, та менших за 5% за вертикальною складовою текстурних координат). Це також теоретично, збільшить швидкість системи візуалізації для всієї решти поверхні сфери. Тобто для всіх інших (не полярних областей) запит даних з текстури буде відбуватися за попередньо розрахованими текстурними координатами які зберігаються в геометричному описі моделі, з автоматичним розрахунком MIP рівнів. А для полярних областей потрібно скорегувати MIP рівень у бік збільшення в залежності від відстані до найближчого полюса. Для цього потрібно обчислити відносні відстані до полюсів (v_S, v_N) від довільно обраного кола - полярних областей, для яких ведеться розрахунок текстурних координат на рівні фрагментів (рис. 8):

$$v_N = \frac{[v-(1-b)]}{b},$$

$$v_S = 1 - \frac{v}{b}, \quad (5)$$

де b - відстань від полюсу до кола умовного початку детального розрахунку, v - вертикальна складова текстурних координат.

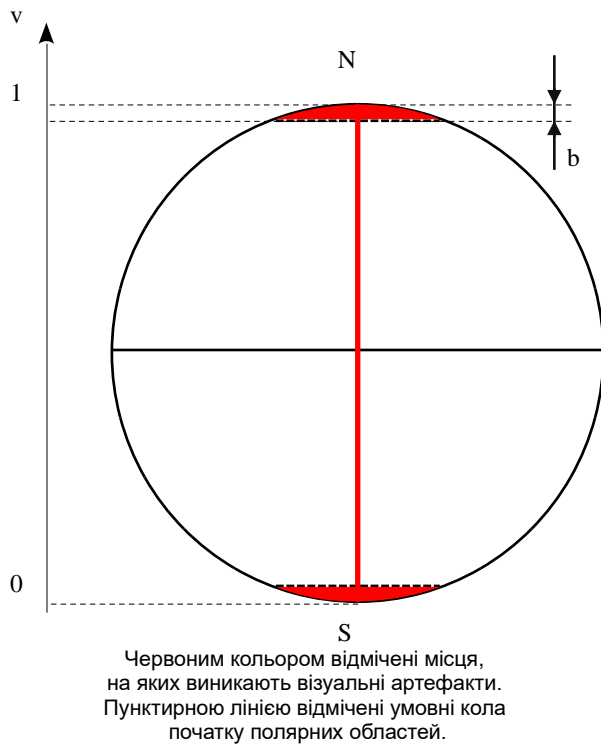


Рис. 8. Межі областей на моделі сфери (вид збоку)

Далі необхідно скорегувати поточний МІР рівень l за допомогою обраної відстані $\hat{v} = (v_N | v_S)$, яка обирається як відстань до найближчого плюсу:

$$\hat{l} = \frac{\max(0, l - 1)}{a\hat{v}}, \quad (6)$$

де \hat{l} - скоригований МІР рівень, a - масштабуючий коефіцієнт (від 2 до 8 в залежності від загальної кількості МІР рівнів у текстурі яка накладається).

Після розрахунку \hat{l} за виразом (6), він безпосередньо використовується у запитах даних з текстур.

Практичне дослідження запропонованого методу

Випробування проводилися на ПК (CPU - Intel Core i5 11400H 2.7GHz, GPU - nVidia 3050 3Gb, RAM - 64Gb) під керуванням ОС Windows 11, із застосуванням графічної бібліотеки DirectX11.

Модель сфери була побудована з 64 розбиттів і містить 14596 вершин які поєднані у 27648 трикутників. Мікропрограми для GPU були написані на мові HLSL (Pixel Shader 5.0). Для рендерингу використовувалася текстура розміром 8192x4096, 32 біт на піксель.

На рис. 9 наведені результати рендерингу полярної області сфери згідно запропонованому алгоритму розрахунку МІР рівня.

Як можна побачити, всі зазначені вище артефакти повністю зникли.

Результати роботи системи візуалізації із застосуванням запропонованого методу для різних тіл сонячної системи наведені на рис.10.

Після компіляції мікропрограми, з'ясовано, що реалізація алгоритму потребує 59 додаткових інструкцій, які виконуються лише для 10% поверхні

сфери, за допомогою оператора динамічного розгалуження. Для профілювання мікропрограми застосовувався спеціалізований додаток PIX on Windows. З мікропрограми були вилучені всі операції пов'язані з розрахунком освітлення.

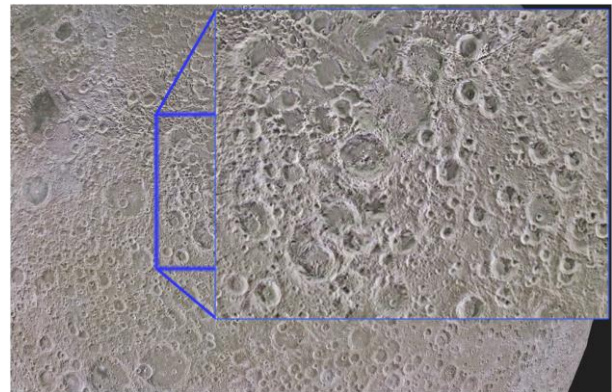


Рис. 9. Рендеринг полярних областей моделі за допомогою запропонованого алгоритму

Для оцінки швидкодії вимірявся час рендерингу моделі сфери, яка на екрані була вписана у квадрат розміром 1024x1024 пікселя, таким чином мікропрограмою оброблялося приблизно 824 тисячі фрагментів.

Проводилося три фази тестування:

1) стандартний рендеринг з інтерпольованими текстурними координатами (мікропрограма довжиною 6 інструкцій);

2) рендеринг за допомогою запропонованого методу (мікропрограма довжиною 59 інструкцій) вид збоку з мінімальною кількістю фрагментів для яких ведеться розрахунок текстурних координат;

3) рендеринг за допомогою запропонованого методу (мікропрограма довжиною 59 інструкцій) вид згори на полюс з максимальною кількістю фрагментів для яких ведеться розрахунок текстурних координат.

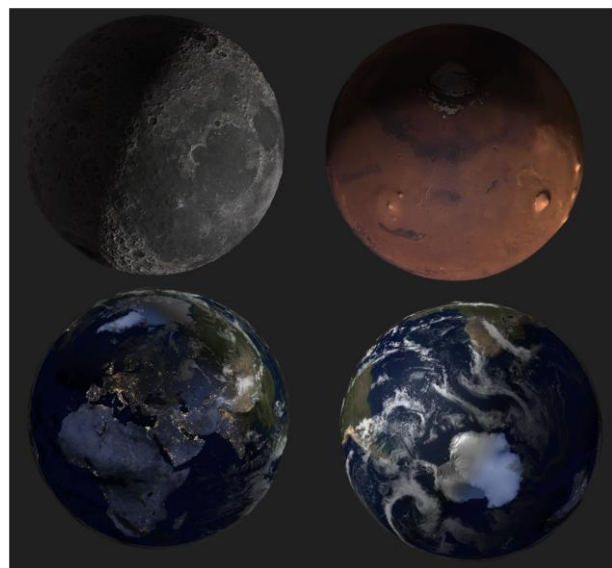


Рис. 10. Результати роботи запропонованого методу у складі системи візуалізації

Кожен тест проводився 10 разів, результат усереднювався.

Результати тестування наведені у табл. 1 (А – № тесту; В – час рендерингу сфери, мкс; С – загальний час рендерингу кадру, мкс).

Таблиця 1 – Час рендерингу сфери

А	Тип мікропрограми	В	С
1	Стандартний запит з текстур (6 інструкцій)	188	443
2	Запропонований метод (59 інструкцій) – вид збоку	188	443
3	Запропонований метод (59 інструкцій) – вид згори на полюс	205	465

Висновки

Наведений алгоритм дозволяє видалити артефакти які виникають при розрахунку текстурних координат як безпосередньо у процесі візуалізації, так і на етапі побудови моделі сфери, що дозволяє

значно покращити відображення полярних областей сфери.

1) Для реалізації алгоритму розрахунку MIP рівня потрібно 59 інструкцій (при компіляції мікропрограми з мови HLSL).

2) Час рендерингу сфери при застосуванні запропонованого методу, при максимальній кількості фрагментів для яких розраховуються текстурні координати, збільшується не більше ніж на 10% (загальний час рендерингу кадру збільшується не більше ніж на 5%).

3) Фактично алгоритм розрахунку MIP рівня виконується лише для 10% фрагментів сфери, а для решти, за допомогою оператора динамічного розгалуження, обирається стандартна мікропрограма запити з текстури з автоматичним розрахунком MIP рівня за текстурними координатами що інтерполюються.

4) Запропонований метод дозволяє видалити артефакти пов'язані з проєціюванням текстури без модифікації геометричної моделі сфери та додавання якихось додаткових елементів до неї.

СПИСОК ЛІТЕРАТУРИ

1. Snyder, J. P. (1997). *Flattening the Earth. Two Thousand Years of Map Projections*. Chicago and London: University of Chicago Press. 384 p. ISBN: 9780226767475
2. Orsolya Gáspár. The optimization process leading to the tessellation of the first geodesic dome structure, the first Planetarium of Jena. *International Journal of Space Structures* 2022, Vol. 37(1) 49–64. DOI: 10.1177/09560599211064110
3. Catmull, E.; Clark, J. (1978). Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*. 10 (6): 350. DOI:10.1016/0010-4485(78)90110-0.
4. Edward S. Popko, Christopher J. Kitrick. *Divided Spheres: Geodesics and the Orderly Subdivision of the Sphere*. CRC Press, 2021. 484 pages. ISBN 1000412431, 9781000412437
5. Henrik Hargitai, Jue Wang, Philip J. Stooke et al. Choosing a Map Projection. Chapter 7. *Map Projections in Planetary Cartography*. April 2017. pp.177-202. DOI: 10.1007/978-3-319-51835-0_7
6. Goldberg, David M.; Gott III, J. Richard (2007). Flexion and Skewness in Map Projections of the Earth. *Cartographica*. 42 (4): 297–318. arXiv:astro-ph/0608501. DOI:10.3138/carto.42.4.297.
7. Adrian Dziembowski, Dawid Mieloch, Olgierd Stankiewicz et al. Virtual View Synthesis for 3DoF+ Video. 2019 Picture Coding Symposium (PCS). November 2019. DOI: 10.1109/PCS48520.2019.8954502
8. Olga Lukashova-Sanz, Siegfried Wahl. Saliency-Aware Subtle Augmentation Improves Human Visual Search Performance in VR. February 2021. *Brain Sciences* 11(3):283. DOI: 10.3390/brainsci11030283
9. Marco Tarini. Cylindrical and Toroidal Parameterizations Without Vertex Seams. *Journal of Graphics Tools*. 2012 Vol.16(3). pp. 144-150. DOI:10.1080/2151237X.2012.654054
10. Ben Golus. Distinctive Derivative Differences. Pesky Problems with Procedural UVs. [Електронний ресурс] URL: <https://bgolus.medium.com/distinctive-derivative-differences-cce38d36797b>.

Received (Надійшла) 25.09.2023

Accepted for publication (Прийнята до друку) 22.11.2023

Seamless textured sphere synthesis method for visualization in geo-information and mapping systems

A. Zuev

Abstract. The method of creating a seamless textured sphere model and removing artifacts associated with the issues of calculating texture coordinates during cylindrical projection is considered. The **purpose** of the article is to develop a method for synthesis and visualizing spherical textured objects that would not have the disadvantages of existing methods, which are associated with the appearance of visually noticeable visual artifacts related with the textures applied to the sphere surface. A method is given that allows you to correct texture coordinates both at the geometric level and in the process of visualization, in which the sphere is conventionally divided into parts according to the distance from the poles. For each part, the most optimal texturing algorithm is used, which ensures the performance of the method. Calculations using the proposed algorithms can be fully implemented at the cost of GPU resources. Based on the **results** of the study, was determined that the proposed method allows removing artifacts that occur when calculating texture coordinates for cylindrical projection both directly in the visualization process and at the stage of synthesis of the sphere model, which allows to significantly improve the quality of the appearance of the sphere surface.

Keywords: tessellation, equal-area subdivision, GPU, MIP levels, realistic visual scenes, texturing.