



**Матеріали LV науково-технічної конференції
підрозділів Вінницького національного
технічного університету
(НТКП ВНТУ–2026)**

24-27 березня 2026 року

Збірник доповідей

Міністерство освіти і науки України
Вінницький національний технічний університет

Матеріали LV науково-технічної конференції
підрозділів Вінницького національного
технічного університету (НТКП ВНТУ–2026)

24-27 березня 2026 року

Збірник доповідей

Електронне наукове видання

Вінниця
ВНТУ
2026

Видається за рішенням Вченої ради Вінницького національного технічного університету Міністерства освіти і науки України

Головний редактор: В. В. Біліченко
Відповідальний за випуск: І. Ю. Єпіфанова

Робоча група з підготовки конференції:
Голова робочої групи:
проректор з наукової роботи ВНТУ І. Ю. Єпіфанова

Члени робочої групи:

декани факультетів, директор Інституту Конфуція ВНТУ;
В. М. Шпігунов, начальник РВВ ВНТУ;
Г. М. Багдасар'ян, провідний інженер РВВ ВНТУ;
О. О. Кушнір, провідний інженер РВВ ВНТУ;
С. Г. Могила, інженер 1-ї категорії РВВ ВНТУ.

Матеріали LV науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2026) : збірник доповідей [Електронний ресурс]. – Вінниця : ВНТУ, 2026. – (PDF, 3599 с.) ISBN 978-617-8163-95-2

Збірник містить тексти доповідей LV ювілейної регіональної науково-технічної конференції професорсько-викладацького складу, науковців, аспірантів та студентів Вінницького національного технічного університету з участю працівників підприємств м. Вінниці та Вінницької області з загально-інженерних, технічних, гуманітарних та фундаментальних наук.

НТКП ВНТУ проводиться у вигляді конференцій факультетів та конференції Інституту Конфуція ВНТУ. Кожна конференція має власну тематику, оргкомітет, строки проведення пленарних та секційних засідань, та складається з однієї або кількох секцій.

УДК 001

ISBN 978-617-8163-95-2

© Вінницький національний технічний університет, укладання, оформлення, 2026

СУЧАСНІ МЕТОДИ РЕЛЬЄФНОГО ТЕКСТУРУВАННЯ В GPU-ОРІЄНТОВАНИХ СИСТЕМАХ ВІЗУАЛІЗАЦІЇ

Вінницький національний технічний університет

Анотація

Розглянуто сучасні методи реалізації рельєфного текстуровання на графічному процесорі (GPU), зокрема алгоритми, що базуються на карті висот, такі як *parallax occlusion mapping* (POM), та новітні адаптивні й нейронні підходи. Проаналізовано принципи формування псевдорельєфу без геометричної деталізації, а також інтеграцію цих методів у фізично-обґрунтоване рендеринг-середовище (PBR). Особливу увагу приділено питанням оптимізації продуктивності, адаптивного вибору глибини проходження променя, використанню карт складності, ієрархічних структур даних та GPU-орієнтованих шейдерів. Обґрунтовано доцільність використання рельєфного текстуровання як способу зменшення навантаження на графічний конвеєр при збереженні високої якості візуалізації. Робота має практичне значення для проєктування 3D-додатків у сфері візуалізації, ігрової індустрії, моделювання та віртуальної реальності.

Ключові слова: рельєфне текстуровання, графічний процесор, *parallax occlusion mapping*, карта висот, GPU, мікрогеометрія, візуалізація.

Abstract

Modern methods for implementing relief texturing on a graphics processing unit (GPU) are considered, in particular algorithms based on a height map, such as *parallax occlusion mapping* (POM), and the latest adaptive and neural approaches. The principles of pseudo-relief formation without geometric detailing are analyzed, as well as the integration of these methods into a physically based rendering environment (PBR). Special attention is paid to the issues of performance optimization, adaptive selection of ray depth, the use of complexity maps, hierarchical data structures and GPU-oriented shaders. The feasibility of using relief texturing as a way to reduce the load on the graphics pipeline while maintaining high quality visualization is substantiated. The work has practical significance for the design of 3D applications in the field of visualization, gaming, modeling and virtual reality.

Keywords: render рельєфне текстуровання, графічний процесор, *parallax occlusion mapping*, карта висот, GPU, мікрогеометрія, візуалізація.

Вступ

У сучасному комп'ютерному моделюванні та візуалізації рельєфне текстуровання [1-13] посідає важливе місце в досягненні реалістичності 3D-графіки без надмірного збільшення геометричної складності сцени. Його основна мета - створити враження об'ємної поверхні на плоскій сітці, використовуючи карти висот і нормалей. Особливу актуальність рельєфне текстуровання має в комп'ютерних іграх, VR/AR-середовищах, симуляторах, архітектурній візуалізації, де важливо забезпечити високу якість зображення при обмежених обчислювальних ресурсах. Реалізація рельєфного текстуровання саме на GPU [11] дає змогу ефективно використовувати паралельну обробку пікселів і фрагментів, властиву графічним процесорам. Застосування таких методів, як *parallax occlusion mapping*, дозволяє відтворювати дрібні деталі - тріщини, виступи, гравіювання - без необхідності додавання нових полігонів. Це суттєво знижує навантаження на геометричний конвеєр і дозволяє рендерити сцени в реальному часі навіть на споживчому обладнанні.

Окрім візуальної якості, рельєфне текстуровання впливає на освітлення, затінення та взаємодію світла з матеріалом поверхні. Наприклад, самозатінення у методах на основі карти висот дозволяє точніше відобразити поведінку світла на мікрорівні. Реалізація цих ефектів на GPU забезпечує інтеграцію з фізично-обґрунтованими моделями освітлення, підвищуючи вірогідність сцени та її занурення.

З практичної точки зору, рельєфне текстурювання дозволяє дизайнерам і розробникам створювати високодеталізовані матеріали без значних затрат на обчислення або об'єм пам'яті. Замість геометрії можна зберігати лише дві текстури - карту нормалей і висот, що значно полегшує ресурсоемність проєктів. В умовах мобільної графіки або застосунків зі строгими обмеженнями продуктивності це є критично важливим.

Результати дослідження

Сучасні методи рельєфного текстурювання в основному реалізуються на графічному процесорі (GPU) [11] і використовують фрагментні шейдери. Найбільш поширені техніки - це parallax mapping, steep parallax mapping, parallax occlusion mapping (POM) та relief mapping [1].

Parallax mapping дозволяє змістити UV-координати текстурної вибірки у фрагментному шейдері залежно від кута огляду, зберігаючи при цьому низьке обчислювальне навантаження. Проте така техніка не враховує самозатінення та точний перетин віртуального променя з висотною картою. Для цього використовують покращені варіанти, зокрема steep parallax mapping та parallax occlusion mapping (POM), які виконують дискретний кроковий пошук перетину між псевдопроменем зору та даними карти висот [2].

Реалізація POM на GPU передбачає ітеративне зменшення координат UV у напрямку, оберненому до напрямку погляду, доки обчислена віртуальна глибина не стане меншою або рівною значенню з карти висот. Такий підхід дозволяє точно відобразити локальні зміщення, пори, тріщини та інші мікродеталі поверхні навіть при фіксованій геометрії [3]. Практична реалізація потребує нормалізованого вектору огляду у дотичному просторі та дотримання стабільної кількості ітерацій. У разі поганого кутового положення можуть виникати артефакти, що вимагає запровадження механізмів фейдування або зменшення масштабу рельєфу [4].

Використання POM та аналогічних методів супроводжується проблемами у відтворенні коректного рівня деталізації текстур. Зокрема, звичайне використання функцій вибірки текстур може давати некоректні похідні, що призводить до появи aliasing-ефектів. Для їх уникнення рекомендовано застосовувати методи з явним заданням похідних або з фіксованим рівнем міппаінгу [5]. Це особливо актуально при реалізації PBR (physically based rendering), де точність відтворення мікрогеометрії критична для енергозбереження та візуальної достовірності [7].

Іншою важливою складовою є самозатінення - ефект затінення областей рельєфу, які є недоступними для джерела світла через виступи самої поверхні. У POM цей ефект досягається шляхом додаткового проходу ray marching у напрямку до джерела світла. Якщо на певному етапі зчитане значення висоти перевищує глибину променя, точка вважається затіненою. Такий підхід є обчислювально дорогим, тому часто застосовують його лише для ближніх LOD-рівнів або на обмежених зонах поверхні [7].

Крім класичних методів POM, сучасні рушії візуалізації впроваджують геометричні displacement-підходи, зокрема на базі тесселяції або mesh shaders. У рушії Unreal Engine 5 активно використовується система Nanite, яка дозволяє представляти детальну геометрію за допомогою мільйонів трикутників, що адаптивно підвантажуються з урахуванням масштабу сцени та кута огляду [10]. Проте навіть у таких умовах рельєфне текстурювання не втрачає актуальності, оскільки воно забезпечує гнучкість для матеріалів без зміни геометрії [11].

З GPU-орієнтованої точки зору, реалізація POM повинна бути оптимізованою з урахуванням обмежень паралельного виконання: уникається надмірна дивергенція вейвлетів, розгалуження в циклах мінімізуються, використовується фіксована кількість ітерацій із подальшим уточненням результату. Крім того, у Vulkan і DirectX 12 реалізовано можливості mesh shading, які дозволяють реалізовувати рельєф на етапі генерації геометрії, а не лише у фрагментному шейдері, що відкриває нові перспективи [12].

Висотно-рельєфне текстурювання також має значний вплив на інтеграцію з системами тіней, глобального освітлення та ambient occlusion. Для забезпечення узгодженості між рельєфом та затіненням використовуються техніки, що враховують мікрогеометрію при генерації глибини в G-буфері, або спеціалізовані методи екранного простору, які враховують рельєф при обчисленні оклюзії [13].

Сучасні засоби рендерингу та графічні API (як-от DirectX 12, Vulkan) підтримують гнучке управління шейдерами, що дозволяє створювати кастомні реалізації POM із підтримкою self-shadowing, мультिकанального освітлення, LOD та екранного фейдування. Завдяки цьому розробники отримують

інструмент для створення глибоких, деталізованих поверхонь без збільшення полігонального навантаження.

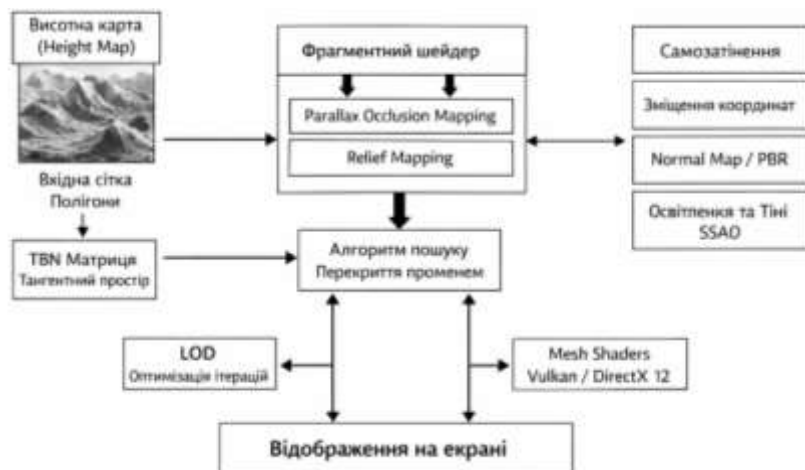


Рис. 1. Реалізація рельєфного текстуровання

Схема (рис. 1) зображає повну логіку реалізації рельєфного текстуровання у графічному процесорі, починаючи з вихідних даних у вигляді карти висот і закінчуючи відображенням результату на екрані. У лівій частині схеми знаходиться блок «Висотна карта», що позначає текстурне зображення у відтінках сірого, де яскравість відповідає локальній висоті поверхні. Від цієї карти йде стрілка до фрагментного шейдера, у якому відбувається основна обробка. Нижче розташовано блок «Вхідна сітка (Mesh)», що передає топологію об'єкта та використовується для побудови TBN-матриці -базису в дотичному просторі. Побудована TBN-матриця подається в блок «Перетворення напрямку погляду» для формування напрямку огляду у дотичному просторі, необхідного для подальшого проходження променем. Центральне місце на схемі займає блок «Фрагментний шейдер», що обробляє фрагменти з урахуванням текстур. У цьому блоці виконуються методи рельєфного текстуровання, зокрема «Parallax Mapping» та «Parallax Occlusion Mapping». У процесі використовується карта висот і нормалей, а також напрямок погляду, щоб обчислити зсув UV-координат і симулювати рельєф. Із фрагментного шейдера також відгалужуються окремі блоки, які відповідають за розрахунок самозатінення, зміщення текстурних координат та отримання нормалей для PBR-освітлення. Блок самозатінення показує проходження додаткового променя до джерела світла, що перевіряє, чи певна точка на поверхні потрапляє в тінь від елементів рельєфу. Окремий блок відповідає за інтеграцію з системою глобального освітлення, зокрема SSAO -оклюзії у просторі екрана. Праворуч від центральної частини схеми розташовані блоки «LOD» (рівні деталізації) та «Mesh Shaders / Vulkan / DirectX», що позначають сучасні графічні API та методи оптимізації обчислень на GPU. Вони отримують дані з фрагментного шейдера та TBN-простору, забезпечуючи адаптацію обчислень під конкретний рівень деталізації. У нижній частині схеми відображено блок «Екран», до якого сходяться всі потоки даних і обчислень, демонструючи, що результат рельєфного текстуровання формується на основі поєднання геометрії, текстур, освітлення та алгоритмів на GPU. Стрілки між блоками ілюструють послідовність обчислень: спочатку формується TBN-простір, потім фрагментний шейдер виконує ітеративний пошук за допомогою height map, далі UV зміщується, вибираються нові текстурні значення, обчислюється нормаль, додається самозатінення і фінально отриманий колір виводиться на екран. У верхній частині зазначено, що реалізація виконується у шейдерному коді на мові HLSL або GLSL, залежно від API (DirectX чи Vulkan).

Розглянемо нові методи, які з'явилися або набули поширення протягом останніх кількох років.

Один з таких методів -адаптивний Parallax Occlusion Mapping. Його суть полягає в тому, що кількість кроків при трасуванні променя через карту висот не є фіксованою, а залежить від кута огляду камери або локальної складності карти. Таким чином, на ділянках із високим нахилом або великою деталізацією виконується більше ітерацій для точного обчислення рельєфу, тоді як на спокійних або майже перпендикулярних до погляду ділянках витрати зменшуються. Це дозволяє досягти кращого балансу між якістю й продуктивністю, що особливо важливо для ігор і VR-додатків.

Іншим новим підходом є використання карт складності. У таких схемах ще до виконання POM обчислюється рівень складності висотної карти для кожного пікселя або блоку. Це можна зробити за допомогою фільтрів, як-от оператор Собеля або інших градієнтних методів, які виявляють зміну висоти. Отримана карта потім використовується для визначення кількості ітерацій під час ray-marching, або навіть для вибору того, чи варто застосовувати POM взагалі. Цей метод особливо ефективний на гетерогенних поверхнях, де деякі області потребують детального рельєфу, а інші - ні.

Ще один важливий напрям - впровадження ієрархічних структур даних, що дозволяють прискорити виконання обчислень у POM. Сюди належать Rectangular MIP та інші спеціалізовані індексаційні структури, які допомагають уникати надмірного обчислення в гладких зонах. Завдяки цим структурам фрагментний шейдер може швидше виключати непотрібні ділянки й зосереджуватись лише на тих, що містять складний рельєф.

Новим революційним напрямом є застосування нейронних методів. У рамках цього підходу замість традиційних height map і normal map використовуються так звані нейронні матеріали. Вони можуть генерувати дані для рельєфу за допомогою попередньо навчених моделей машинного навчання. Наприклад, методи на зразок NeuMIP дозволяють створювати складні візуальні ефекти рельєфу без явного використання карт висот, замінюючи їх нейронними офсетами (neural offsets). Такі підходи демонструють вражаючу візуальну якість, зменшуючи обсяг даних і тиск на пам'ять GPU.

Крім того, у нових API, таких як Vulkan та DirectX 12, з'явилися mesh shaders та task shaders, які дозволяють перенести частину обробки рельєфу на геометричний рівень. Завдяки цьому стало можливим частково геометрично представляти рельєф, комбінуючи його з POM або displacement mapping. Наприклад, замість трасування у фрагментному шейдері деякі елементи поверхні можуть бути створені або уточнені за допомогою mesh shader, що дозволяє досягти правильних силуетів та уникнути псевдоперетинів, притаманних POM.

Також спостерігається тенденція до інтеграції рельєфного текстуровання з сучасними методами глобального освітлення, або навіть трасування променів у реальному часі. Це дозволяє враховувати мікрогеометрію рельєфу в глобальній моделі затінення, що значно підвищує реалістичність сцени. Наприклад, рельєф може не лише деформувати поверхню, а й створювати тіні та оклюзії в залежності від конфігурації поверхні та положення джерела світла.

На завершення, новітні методи рельєфного текстуровання на GPU [11] поєднують у собі гнучкість класичних підходів і потужність сучасного апаратного та програмного забезпечення. Завдяки адаптивному POM, картам складності, ієрархічним структурам, нейронним матеріалам та інтеграції з mesh shaders і глобальним освітленням, розробники отримують широкий спектр інструментів для створення високореалістичних візуальних ефектів без критичного навантаження на систему. Ці методи активно розвиваються й поступово інтегруються в рушії наступного покоління, зокрема Unreal Engine 5, Unity HDRP та інші.

Таким чином, реалізація рельєфного текстуровання на GPU - це поєднання теоретичних принципів побудови мікрогеометрії, алгоритмічної оптимізації для масового паралелізму та інтеграції з сучасними рендеринг-пайплайнами. Досвід українських дослідників, зокрема ВНТУ, підтверджує актуальність дослідження цього напрямку як з точки зору освітніх програм, так і з позицій інженерного впровадження в ігрову та візуалізаційну індустрію [15].

Висновки

Рельєфне текстуровання є важливим компонентом сучасної комп'ютерної графіки, що дозволяє значно підвищити візуальну якість зображень без потреби у деталізованій геометрії. Реалізація цього підходу на GPU забезпечує ефективне використання паралельної архітектури графічного процесора, що критично важливо для реального часу. Серед новітніх методів рельєфного текстуровання особливу увагу заслуговують адаптивні схеми Parallax Occlusion Mapping, використання карт складності, ієрархічних структур даних та нейронних моделей. Ці техніки дозволяють динамічно зменшити обчислювальне навантаження, адаптуючи рендеринг до змісту сцени та параметрів камери.

Застосування рельєфного текстуровання також має вагоме значення для фізично-обґрунтованого рендерингу, впливаючи на затінення, розсіювання світла та мікрофасетну модель поверхні. Технології нового покоління - такі як mesh shaders, GPU-driven geometry, а також інтеграція з глобальним освітленням - підсилюють потенціал рельєфу в системах візуалізації. У підсумку, рельєфне текстуровання на GPU не лише зберігає свою актуальність, а й еволюціонує у відповідь на нові апаратні можливості, стаючи невід'ємною частиною графічного конвеєра як у науково-технічних, так і в ігрових застосуваннях.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Романюк О. Н., Майданюк В. П., Новосельцев О. О. Класифікація методів рельєфного текстуровання для високореалістичного рендерингу. Наукові праці Вінницького національного технічного університету. 2025. Вип. 3. DOI: 10.31649/2307-5376-2025-3-101-111.
2. Романюк О. Н., Захарчук М. Д., Новосельцев О. О., Котлик С. В. Використання рельєфного текстуровання в комп'ютерних іграх. Матеріали конференції «Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації - 2025». Одеса, 2025. С. 454-456.
3. Романюк О. Н., Дудник О. О., Романюк О. В., Озерчук Д. А. Метод перспективно-коректного текстуровання. Інформаційні технології та комп'ютерна інженерія. 2021. Том 50, № 1. DOI: 10.31649/1999-9941-2021-50-1-55-63.
4. Романюк О. Н., Романюк О. В., Чехмestрук Р. Ю. Комп'ютерна графіка: електронний навчальний посібник. Вінниця: ВНТУ. 2023.
5. Karis B. Nanite: UE5's New Virtualized Geometry System. Advances in Real-Time Rendering in Games (SIGGRAPH Courses). 2021.
6. Wihlidal G. A Deep Dive into Nanite Virtualized Geometry. Advances in Real-Time Rendering in Games (SIGGRAPH Courses). 2021.
7. Pharr M., Jakob W., Humphreys G. Physically Based Rendering: From Theory to Implementation. 4th ed. Cambridge, MA: The MIT Press. 2023. ISBN 9780262048026.
8. Романюк О. Н., Новосельцев О.О., Теренчук А.Т., Котлик С.В. Динамічне рельєфне текстуровання. Інформаційні технології і автоматизація – 2025 / Матеріали XVIII міжнародної науково-практичної конференції. Одеса, 30-31 жовтня 2025 р. - Одеса, Видавництво ОНТУ, 2025 р. -С. 1165-1167.
9. Романюк О. Н., Захарчук М. Д., Новосельцев О. О., Котлик С. В. Використання рельєфного текстуровання в комп'ютерних іграх. Матеріали конференції «Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації - 2025». Одеса, 2025. С. 454-456.
10. Романюк О.Н., Майданюк В.П., Романюк О.В., Новосельцев О.О., Мельник О.О. Використання паралаксного текстуровання для підвищення реалістичності формування тривимірних зображень .Збірник наукових праць IV міжнародної науково-практичної конференції (Запоріжжя Мелітополь, 20 травня 2025 р., МДПУ імені Богдана Хмельницького). Випуск 14. Запоріжжя: Видавництво МДПУ ім. Б. Хмельницького, 2025. 543-550.
11. Романюк О.Н., Новосельцев О.О. Реалізації рельєфного текстуровання в GPU Перспективи розвитку сучасної науки та освіти: матеріали XIV Міжнародної науково-практичної конференції м. Львів, 19-20 січня 2025 року. – Львів : Львівський науковий форум, 2025. – с.52-55
12. Романюк О.Н., Романюк О.В.Бази даних для задач текстуровання тривимірних зображень. Теоретичні та практичні аспекти розвитку науки та освіти: матеріали XV Міжнародної науково-практичної конференції м. Львів, 15-16 липня 2025 року. – Львів : Львівський науковий форум, 2025. – 148 с.
13. Романюк О. Н., Новосельцев О. О., Станіславенко С. Г., Майданюк В. П., Романюк С. О. Аналіз нових методів текстуровання. Збірник матеріалів Міжнародної науково-практичної Інтернет конференції "Електронні інформаційні ресурси: створення, використання, доступ та управління", 20-21 листопада 2024 р. Суми/Вінниця : НІКО / КЗВО «Вінницька академія безперервної освіти», 2024. С. 171-174

Романюк Олександр Никифорович— д-р техн. наук, професор, завідувач кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця

Майданюк Володимир Павлович — канд. техн. наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця

Новосельцев Олександр Олександрович — аспірант кафедри програмного забезпечення інженерії, Вінницький національний технічний університет, м. Вінниця

Romanyuk Oleksandr Nikiforovich— Dr. of Technical Sciences, Professor, Head of the Department of Software, Vinnytsia National Technical University, Vinnytsia

Maydanyuk Volodymyr Pavlovych — Candidate of Technical Sciences, Associate Professor of the Department of Software, Vinnytsia National Technical University, Vinnytsia

Novoseltsev Oleksandr Oleksandrovich — Postgraduate Student of the Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia