

РЕАЛІЗАЦІЯ РЕЛЬЄФНОГО ТЕКСТУРУВАННЯ НА ГРАФІЧНИХ ПРОЦЕСОРАХ

Романюк О.Н., Майданюк В.П., Новосельцев О.О.

Вінницький національний технічний університет, м. Вінниця

Реалізація рельєфного текстуровання [1] на графічних процесорах (GPU) базується на використанні спеціальних текстур, що визначають деталі поверхні без зміни геометрії самої моделі. Головна ідея полягає в тому, що замість створення додаткових полігонів для відображення рельєфу, використовується нормал-мапа або карта висот, яка в шейдері коригує нормалі поверхні, впливаючи на освітлення і створюючи ілюзію нерівностей. Технічно цей процес здійснюється на етапі піксельного або фрагментного шейдингу в графічному конвеєрі GPU. Піксельний шейдер зчитує значення з нормал-мапи, перераховує нормаль поверхні з урахуванням отриманих даних та використовує отриману нормаль для обчислення освітлення кожного пікселя. Це забезпечує візуальне сприйняття складних поверхонь, таких як камінь, дерево чи метал, з високим рівнем деталізації, без необхідності складного геометричного моделювання. GPU сучасних архітектур, таких як Nvidia CUDA чи AMD RDNA, виконують ці операції паралельно, дозволяючи швидко і ефективно відтворювати складні сцени в реальному часі. Завдяки цьому, рельєфне текстуровання широко застосовується у комп'ютерних іграх, віртуальній реальності, візуалізації продуктів і високоякісній 3D-графіці.

При реалізації рельєфного текстуровання на GPU використовуються кілька основних методів. Найбільш поширеним є *normal mapping*, де кольори текстури зберігають інформацію про вектори нормалей поверхні у вигляді RGB-значень. Ці значення в шейдері переводяться в нормалізовані вектори, що використовуються для точного розрахунку відбиття світла. Інший важливий метод – *parallax mapping*, який створює більш переконливу ілюзію глибини за рахунок зміщення текстурних координат залежно від кута огляду і карти висот. *Parallax Occlusion Mapping* є удосконаленим варіантом цього методу, що використовує ітеративний процес пошуку перетину променя з поверхнею на основі карти висот. Це дає змогу досягти візуально точніших і складніших ефектів, таких як глибокі тріщини чи нерівності, що залежать від кута спостереження. Ці методики ефективно виконуються саме на GPU завдяки високій паралельності та оптимізованим потоковим обчисленням сучасних графічних процесорів. При програмній реалізації використовується GLSL, HLSL або інші шейдерні мови, де вся логіка текстуровання та освітлення описується безпосередньо у шейдерних програмах.

Література

1. Романюк О. Н. Комп'ютерна графіка : електронний навчальний посібник / О. Н. Романюк, О. В. Романюк, Р. Ю. Чехмestрук. – Вінниця : ВНТУ, 2023. – 147 с.