

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Одеська національна академія харчових технологій**  
**Університет Інформатики і прикладних знань, м.Лодзь, Польща**  
**Національний технічний університет України «Київський**  
**політехнічний інститут»**  
**Навчально-науковий інститут комп'ютерних систем і технологій**  
**«Індустрія 4.0» ім. П.М. Платонова**

**XXI Всеукраїнська науково-технічна конференція**  
**молодих вчених, аспірантів та студентів**

**«СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ**  
**ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ»**

*Матеріали конференції*



Одеса

22-23 квітня 2021 р.

Стан, досягнення та перспективи інформаційних систем і технологій / Матеріали XXI Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 22-23 квітня 2021 р. - Одеса, Видавництво ОНАХТ, 2021 р. – 229 с.

Збірник включає матеріали доповідей учасників конференції, які об'єднані за тематичними напрямками конференції.

## **ОРГАНІЗАЦІЙНИЙ КОМІТЕТ**

**Голова** - д.т.н., проф., **Єгоров Б.В.**, ректор ОНАХТ.

### **Співголови:**

**Поварова Н.М.** – к.т.н., доц., проректор з наукової роботи ОНАХТ,  
**Котлик С.В.** – к.т.н., доц., директор ННІКСІТ "Індустрія 4.0" ОНАХТ,  
**Даріуш Долива**, д.математичн.наук, уповноважений декана факультету Інформатики УІтаПЗ, м.Лодзь, Польща,  
**Ковалюк Т.В.** - к.т.н., доц. кафедри АСОІтаУ НТУУ «Київський політехнічний інститут»

### **Члени оргкомітету:**

**Плотніков В. М.** – д.т.н., проф., завідувач кафедри ІТтаКБ ОНАХТ,  
**Артеменко С.В.** – д.т.н., проф., завідувач кафедри КІ ОНАХТ,  
**Хобін В.А.** – д.т.н., проф., завідувач кафедри АТПтаРС ОНАХТ,  
**Тарасенко В.П.** – д.т.н., проф., завідувач кафедри СКС НТУУ «Київський політехнічний інститут»,  
**Невлюдов І.Ш.** – д.т.н., проф., завідувач кафедри КІТАМ ХНУРЕ,  
**Мельник А.О.** – д.т.н., проф., завідувач кафедри ЕОМ НУ “Львівська політехніка”,  
**Жуков І.А.** – д.т.н., проф., завідувач кафедри КСтаМ НАУ.

Матеріали подано українською, російською та англійською мовами.  
Редактор збірника Котлик С.В.

As a result of the analysis, it was concluded that when developing a mobile robot for movement and orientation in confined spaces, the IDA \* search algorithm has advantages, and for the development of combat mobile robot - it is better to use the Lee Algorithm - wave tracing.

#### **List of literature sources**

1. K. K. Tomas, R. L. Rivest, and K. F. Shtajn, "Algoritmi: pobudova i analiz," M.: «Vilyams», pp. 296-391, Aug. 2006.
- 2 Y.G. Martynenko, "Upravlenie dvizheniem mobilnykh kolesnykh robotov," pp. 1590-1594, Sept. 2005.

УДК 004.925

### **АНАЛІЗ ШЕЙДЕРІВ**

РОМАНЮК О. Н., ІВАХА О. А., ДУДНИК О.О.  
Вінницький національний технічний університет

*Комп'ютерна графіка широко використовується в різних галузях діяльності людини, оскільки забезпечує реалістичне відтворення предметів і процесів [1]. При формуванні графічних зображень широко використовуються шейдери.*

Шейдером [1-7] називається програма для візуального визначення поверхні об'єкта. Це може бути опис розсіювання та поглинання світла, накладення текстури.

**Шейдер** – це комп'ютерна програма, яку виконують процесори відеокарти (GPU). Відеокарта обробляє потік даних, які мають точку входу та точку виходу, код команд, які виконують обробку. Під «потокком даних» мається на увазі інформація про координати вершин полігонів і текстурних координат, які передаються в програму шейдера.

Існує три типи рейдерів [2-7]: вершинні (вертексні), геометричні та фрагментні (піксельні). Основна перевага використання шейдерів – їх гнучкість, що спрощує і здешевлює цикл розробки програми, і при тому що підвищує складність і достовірність сцен, що візуалізуються.

**Вертексний (вершинний) шейдер** – це найбільш усталений і найпоширеніший тип серед 3D-шейдерів. Метою таких шейдерів є перетворення 3D-положення кожної вершини у віртуальному просторі до 2D-координати, при якій вона відображається на екрані.

Вершинний процесор може виконувати різні види операцій з графікою: маніпулювання координатами положення кольору або текстури, перетворення вершин і нормалі. Вершинні шейдери забезпечують чіткий контроль над деталями положення, руху, освітлення та кольору.

**Піксельний шейдер**, також відомий як фрагментний, обчислює колір, текстурні координати, глибину та інші параметри кожного фрагмента зображення. Піксельний шейдер використовується на кінцевому етапі графічного конвеєра для формування фрагмента зображення [1].

**Геометричний шейдер** – це шейдер, який здатний обробляти та генерувати нові графічні примітиви, такі як точки, лінії та трикутники. Крім того, геометричний шейдер здатний генерувати примітиви «на льоту», не залучаючи при цьому центрального процесора. Вперше почав використовуватися на відеокартах Nvidia серії 8 [8].

На даний момент існує близько шести різних мов програмування шейдерів, і всі вони так чи інакше транслюють в одні і ті ж машинні коди, які виконуються на GPU [6].

**Шейдерні мови програмування** є C-подібними, але в той же час більш гнучкі в роботі зі змінними. У цілому, мова для шейдерів націлена під конкретне завдання, а саме: вираховування з плаваючою комою, матрицями чи векторами.

Максимальна ефективність і швидкість шейдерної програми залежить від ряду умов :

- програма повинна бути простою та короткою;
- чим менше обчислень, тим швидше буде працювати шейдер;

- чим більш лінійна програма, тим більш оптимізовано буде працювати шейдер.

Типи даних, котрі використовуються при побудові програм шейдерів, діляться на категорії:

- цілочисельні та логічні (bool, int, half);
- з плаваючою комою (float, double);
- вектори (float2, float3, float4, double3, bool2 і т.п., або vector <float, 3>);
- матриці (float2x2, float3x3, float4x4, double2x2, int4x3, або matrix <float, 2, 2> і т. п.).

У шейдерних мовах програмування [5-7] використовуються також арифметичні оператори, але зі своєю специфікою. Наприклад, операція додавання та віднімання по-різному реалізується для float, вектору та матриці. Тим не менш, це не заважає перемножити змінні різних типів: наприклад, вектор можна помножити на число, а матрицю – на вектор.

Шейдерна мова високого рівня HLSL надає розробнику множинну функцій. Ось лише деякі з них, які досить часто застосовують на практиці:

- sin, cos, tan, arctan, arcsin, arccos;
- mul – множить між собою вектори, матриці і числа;
- length (vector) – розраховує довжину вектору;
- distance (a, b) – відстань між двома точками, заданими векторами;
- dot - вираховує dot product;
- lerp - лінійна інтерполяція;
- tex1D, tex2D, tex3D - отримання пікселя з текстурного семплера для одномірної, двомірної та тривимірної текстури [1].

Проведені дослідження показують, що використання шейдерів дозволяє оптимізувати роботу графічних систем.

Підвищення продуктивності та реалістичності можна досягти за рахунок використання нових підходів для формування тривимірних зображень, підвищення функціональних можливостей та продуктивності шейдерних процесорів.

Шляхом апаратної підтримки найбільш поширених інструкцій по реалізації рендеринга можна підвищити продуктивність шейдерних процесорів і зменшити їх кількість. Розширення функціональності шейдерних процесорів дасть можливість реалізувати більш реалістичні зображення, а також зменшить навантаження на центральний процесор і шину обміну даними між системною платою й відеокартою.

#### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] О. Н. Романюк, та А. В. Чорний, *Високопродуктивні методи та засоби зафарбовування тривимірних графічних об'єктів*. Вінниця, Україна: УНІВЕСУМ-Вінниця, 2006.
- [2] О. Н. Романюк, О. О. Дудник, та Н. С. Костюкова. “Реалізація альтернативного конвеєра рендерингу на GPU з використанням обчислювальних шейдерів“, Наукові праці ДонНТУ Серія “Інформатика, кібернетика та обчислювальна техніка”. -№2 (25), с.103-108, 2017.
- [3] О. Н. Романюк “Підвищення ефективності шейдерних моделей графічних відеокарт за рахунок попереднього аналізу спекулярної складової кольору“, Нові технології. - № 2(20). с. 295- 230. 2008.
- [4] О. Н. Романюк “Розподілення обчислювального процесу зафарбовування між вершинними, пікселними і геометричними шейдерами в графічному процесорі”, Праці дев'ятої Міжнародної науково-практичної конференції «Сучасні інформаційні та електронні технології», Одеса, 19–23 травня 2008.с. 23.
- [5] О. Н. Романюк, А. В. Марущак, та В.А Шмалюк. “Шейдери та їх мови програмування”. The 18 th International scientific and practical conference « MODERN SCIENCE, PRACTICE, SOCIETY » (25-26 May 2020). Boston, USA 2020. с. 402-407
- [6] А. В. Боресков. *Разработка и отладка шейдеров*. СПб: БХВ-Петербург, 2006.
- [7] “Шейдер, типи шейдерів, шейдерні мови програмування“. [Електронний ресурс]. Доступно: <https://uk.wikipedia.org/wiki/%D0%A8%D0%B5%D0%B9%D0%B4%D0%B5%D1%80>.