

**Міністерство освіти і науки України**  
**Вінницький національний технічний університет**  
**Вінницька академія неперервної освіти**  
**Національна академія Державної прикордонної служби України**  
**ім. Богдана Хмельницького**  
**Люблінська політехніка (Польща)**  
**Новий університет Лісабону (Португалія)**

**«ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ РЕСУРСИ:  
СТВОРЕННЯ, ВИКОРИСТАННЯ, ДОСТУП»**

**ЗБІРНИК МАТЕРІАЛІВ**

**Міжнародної науково-практичної Інтернет-конференції**

**24-25 жовтня 2016 р.**

**2016**

**УДК 004**  
**ББК 32.97**  
**Е50**

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 9 від 25.02.2016 р.)

**Електронні інформаційні ресурси: створення, використання, доступ: Збірник матеріалів Міжнародної науково-практичної Інтернетконференції.** – Вінниця : ВНТУ, 2016. – 694 с.

ISBN 978-966-641-656-1

Збірник містить матеріали Міжнародної науково-практичної Інтернет-конференції «Електронні інформаційні ресурси: створення, використання, доступ». Матеріали збірника подано у авторській редакції. Автори опублікованих матеріалів несуть повну відповідальність за підбір, точність наведених фактів, цитат, статистичних даних, власних імен та інших відомостей, Матеріали відтворюються зі збереженням змісту, орфографії та синтаксису текстів, наданих авторами.

**УДК 004**  
**ББК 32.97**  
**ISBN 978-966-641-656-1**

© Вінницький національний  
технічний університет, 2016

<b>ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ .....</b>	
Петух А. М., Гончарук В. В.	
<b>СПОСОБИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ КОМП'ЮТЕРНОЇ МЕРЕЖІ.....</b>	<b>391</b>
Петух А. М., Мудрицький В. В.	
<b>АВТОМАТИЗАЦІЯ ТА УПРАВЛІННЯ ЖИВОЮ ЧЕРГОЮ НА ПІДПРИЄМСТВІ.....</b>	<b>395</b>
Подліняєва О. О.	
<b>ВІРТУАЛЬНА ЕКСКУРСІЯ У РОБОТІ ВЧИТЕЛЯ .....</b>	<b>400</b>
Пойда С. А.	
<b>STEM, STEAM, STREAM ЯК ОСНОВА ПОЛІТЕХНІЧНОЇ ОСВІТИ СУЧАСНОГО ШКОЛЯРА .....</b>	<b>414</b>
Прокопенко Є. В., Іванов А. В.	
<b>ВПЛИВ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ НА ФУНКЦІОНУВАННЯ СИСТЕМИ УПРАВЛІННЯ ПІД ЧАС ПРОВЕДЕННЯ СПЕЦІАЛЬНИХ ПРИКОРДОННИХ ОПЕРАЦІЙ .....</b>	<b>418</b>
Рейда О. М., Боднар Р. В.	
<b>МЕТОДИ ВИЯВЛЕННЯ ПОДІБНОСТІ МІЖ ОБ'ЄКТАМИ.....</b>	<b>421</b>
Рейда О. М., Бойко Д. В.	
<b>ВИКОРИСТАННЯ ВЕБ-РЕСУРСІВ ДЛЯ ПОКРАЩЕННЯ ВІЗУАЛЬНОГО СПРИЙНЯТТЯ ІНФОРМАЦІЇ УЧНЯМИ МОЛОДШИХ КЛАСІВ.....</b>	<b>428</b>
Рейда О. М., Доценко Т. В.	
<b>МЕТОДИ ВІДОБРАЖЕННЯ ЗОРОВОЇ ІНФОРМАЦІЇ ДЛЯ ПОКРАЩЕННЯ АСОЦІАТИВНОГО СПРИЙНЯТТЯ ПРИ ВИВЧЕННІ ІНОЗЕМНОЇ МОВИ.....</b>	<b>432</b>
Романюк О. Н., Дудник О. О.	
<b>ЕВОЛЮЦІЯ КОНВЕЄРА РЕНДЕРИНГУ В ВІДЕОКАРТАХ.....</b>	<b>438</b>

*Романюк О.Н,  
д.т.н., проф., Перший проректор Вінницького  
національного технічного університету*

*Дудник О.О,  
аспірант кафедри програмного забезпечення  
Вінницького національного технічного університету*

## **ЕВОЛЮЦІЯ КОНВЕЄРА РЕНДЕРИНГУ В ВІДЕОКАРТАХ**

*В статті розглянуто основні етапи еволюції архітектури конвеєру рендерингу в графічних процесорах.*

**Ключові слова:** відеокарта, архітектура графічних процесорів, конвеєр рендерингу.

*В статье рассмотрены основные этапы эволюции архитектуры конвейера рендеринга в графических процессорах.*

**Ключевые слова:** социальные сети, деятельность библиотек, библиотека университета.

*The article describes the main stages of the evolution of the architecture of the rendering pipeline in a graphics processor.*

**Keywords:** graphics card, architecture of GPUs, rendering pipeline

У загальному випадку тривимірну сцену можна задати як набір окремих груп елементів: групи тривимірних об'єктів, групи джерел освітлення, групи застосовуваних текстурних карт, групи камер.

У більшості випадків для тривимірного об'єкта задають координати вершин трикутників у просторі сцени та локальні координати в просторі текстури [1].

Текстурою (або текстурною картою) називають дво- або тривимірне зображення, що імітує зорове сприйняття людиною властивостей різних поверхонь. На полігон можна накладати кілька текстур, змішуючи їх різними способами, для того, щоб моделювати зоровий образ потрібного матеріалу [2].

У загальному випадку сучасний прискорювач 3D-графіки повинен мати блок геометричних перетворень і розрахунку освітлення (геометричний

процесор), блок механізму установки примітивів, блок обробки текстур і блок обробки буфера кадру. Геометричний процесор обробляє примітиви та будує проєкцію тривимірної сцени. Блок розрахунку освітлення формує дані про параметри освітлення для вершин примітивів. Блок установки примітивів за допомогою даних, отриманих з буфера глибини, визначає видимість точки, що відображається (пікселя). Видимі пікселі обробляються в блоці текстур із застосуванням однієї з технологій фільтрації та накладення текстури. Піксель, оброблений таким чином, знову поміщається в буфер кадру, замінюючи при цьому попереднє для нього значення, або виконується усереднення попереднього значення кольору пікселя із новим за певним правилом [3].

Швидкість та якість обробки тривимірної сцени суттєво залежать від досконалості інструкцій, переданих додатком графічному прискорювачу. Такі інструкції входять до спеціалізованих прикладних програмних бібліотек (Graphics API). Інструкції повинні враховуватися при побудові апаратних засобів графічного адаптера. Конструкція графічного адаптера повинна відповідати можливостям API [3, 4].

Сукупність апаратних і програмних засобів обробки тривимірних сцен утворюють графічний конвеєр, кінцевим результатом роботи якого є кадр зображення, що відображається на екрані монітора. Групу операцій, що відповідає окремій послідовності взаємопов'язаних дій, прийнято називати етапом, або стадією 3D-конвеєра. Послідовність виконання операцій графічного конвеєра є не обов'язковою, а загальноприйнятою в сучасних графічних підсистемах. При конкретній реалізації на програмному та апаратному рівнях можуть з'являтися деякі специфічні дії (процедури), проте смисловий зміст блоків не змінюється [3,4,5]. В загальному вигляді послідовність етапів конвеєра рендерингу відображена на рисунку 1.



**Рисунок 1 - Етапи конвеєра рендерингу**

На першому етапі конвеєру визначається стан об'єктів, які беруть участь в сцені, яку необхідно відобразити. Стан об'єктів і їх взаємне положення формують логіку подальших дій програми. З кожним об'єктом в сцені пов'язана відповідна поточному моменту геометрична модель. Практично всі операції на першому етапі виконує центральний процесор. Результати його роботи пересилаються в графічний чіпсет за допомогою драйвера [5].

Наступним етапом є декомпозиція (поділ на примітиви) геометричних моделей. Зовнішній вигляд об'єкта формується за допомогою набору певних примітивів. Найчастіше в ролі примітиву виступає трикутник як найпростіша плоска фігура, що однозначно розташовується в тривимірному просторі. Всі інші елементи складаються з таких трикутників.

Сучасні графічні процесори виконують додаткові операції, наприклад, розбиття трикутників на складові, тобто поділ вихідних трикутників на більш дрібні. Деякі графічні чіпсети можуть апаратно обробляти геометричні моделі, побудовані на основі параметричних поверхонь (механізм RT-Patches). Частина графічних процесорів має можливість перетворювати плоскі трикутники в тривимірні поверхні шляхом «видавлювання» в третій вимір (механізм N-

Patches).

Далі до вершин трикутників застосовують різні ефекти перетворень і освітленості. Зміст операцій геометричних блоків сучасних графічних чіпсетів можна динамічно змінювати за допомогою вершинних шейдерів (Vertex Shaders) - спеціальних програм, що призначені для виконання процесорами відеокарти. По завершенні операцій трансформації та розрахунку освітленості параметри вершин нормалізуються і приводяться до цілочисельного виду.

На наступному етапі відбувається так звана установка примітивів (Triangle Setup). До цього часу графічний процесор обробляв вершини окремо. Тепер необхідно «зібрати» вершини в трикутники і перетворити результати в координати і колір кожного пікселя, а також відсікти невидимі області.

У ході формування полігонів визначається видимість об'єктів з позиції спостерігача. Полігони, що знаходяться ближче до камери, можуть загородити більш віддалені полігони. Для зберігання інформації про ступінь віддаленості об'єкта від площини проектування використовують спеціальний буфер глибини (Z-буфер). Сучасні графічні процесори застосовують різні механізми відсікання невидимих полігонів на ранніх етапах 3D-конвеєра для, щоб уникнути зайвих операцій. Дані буфера глибини обробляються спеціалізованими блоками графічного процесора. В результаті, на виході блоку геометричних перетворень отримують проекцію тривимірної сцени на площину візуалізації. Координати і вихідний колір видимих пікселів передаються в текстурний конвеєр (Texture Pipeline). Текстурний конвеєр виконує функції вибірки, фільтрації та накладання текстур.

Графічний чіпсет може мати кілька паралельних текстурних конвеєрів. У кожному з них відбувається накладання текстур різного типу, в тому числі і тих, які самі не відображаються (наприклад, карти висот), а служать для модифікації інших текстур. На цьому етапі в графічних чіпсетах можливе виконання піксельних шейдерів (Pixel Shaders) - спеціальних програм, що реалізують заданий метод зафарбовування поверхонь, порядок змішування текстур, отриманих на виході з кожного конвеєра і т.д.. Тут також враховується

належність пікселя до певного трикутника. В цілому зазначені операції складають суть процесу візуалізації (рендерингу).

На заключному етапі роботи 3D-конвеєра до отриманого в текстурному блоці плоского зображення застосовують операції усунення дефектів (наприклад, часто використовують білінійні, трилінійні, анізотропні і інші фільтри). Операції кінцевої обробки можуть застосовувати певні ефекти (наприклад, туман) до цілком сформованого зображення [4, 5].

Перераховані етапи в конкретних графічних чіпсетах можуть бути переставлені, розділені, об'єднані, виконуватися в кілька проходів, проте їх фізичний зміст залишається незмінним. Технологічно деякі елементи етапів або етапи цілком можуть бути виконані різними способами. Метод реалізації залежить від особливостей програми та відеокарти. Залежно від типу відеоприскорювача частина етапів прораховується програмно, а частина - апаратно.

Розглянемо основні етапи розвитку графічних прискорювачів та їх покоління.

Апаратне прискорення 3D-графіки на персональних комп'ютерах почалося з появи апаратних блоків завантаження і фільтрації текстур. Перший вдалий 3D-акселератор для масового ринку, *Voodoo Graphics* (1996 рік), був двочіповим рішенням. Один чіп, *TexelFX*, являв собою один простий текстурний блок, що завантажувач чотири текселі і виконував білінійну інтерполяцію між ними за один такт. Інший чіп, *PixelFX*, був простим блоком растеризації (ROP), що виводить один піксель за такт. У *Voodoo 2* був доданий другий текстурний блок, що дозволило застосовувати кілька більш складних ефектів, накладаючи до двох текстур на піксель за такт або виконувати трилінійну фільтрацію. Для інших операцій конвеєру використовувався центральний процесор.





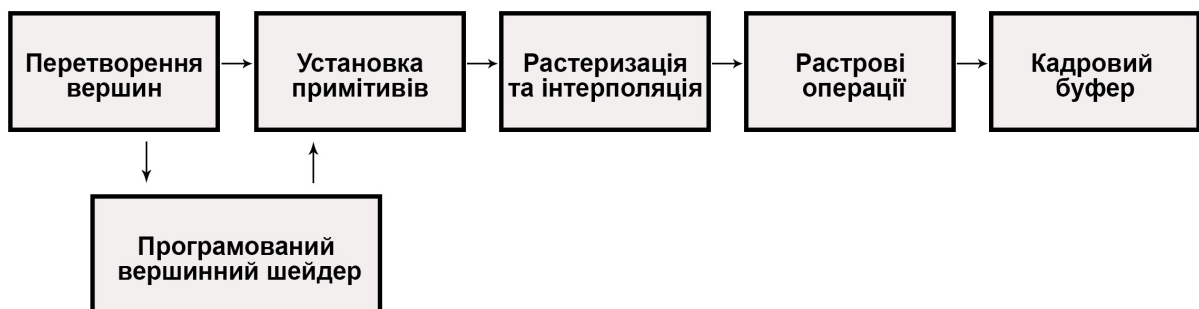
**Рисунок 2 - 3D конвеєр відеокарт першого покоління**

Для відеокарт другого покоління (GeForce/Radeon 7500 1998 рік) основною інновацією стала можливість виконання T&L обчислень на GPU. Крім того, важливим нововведенням стала поява мультитекстурування, що дало можливість накладання в реальному часі карт висот, карт освітлення та інших.



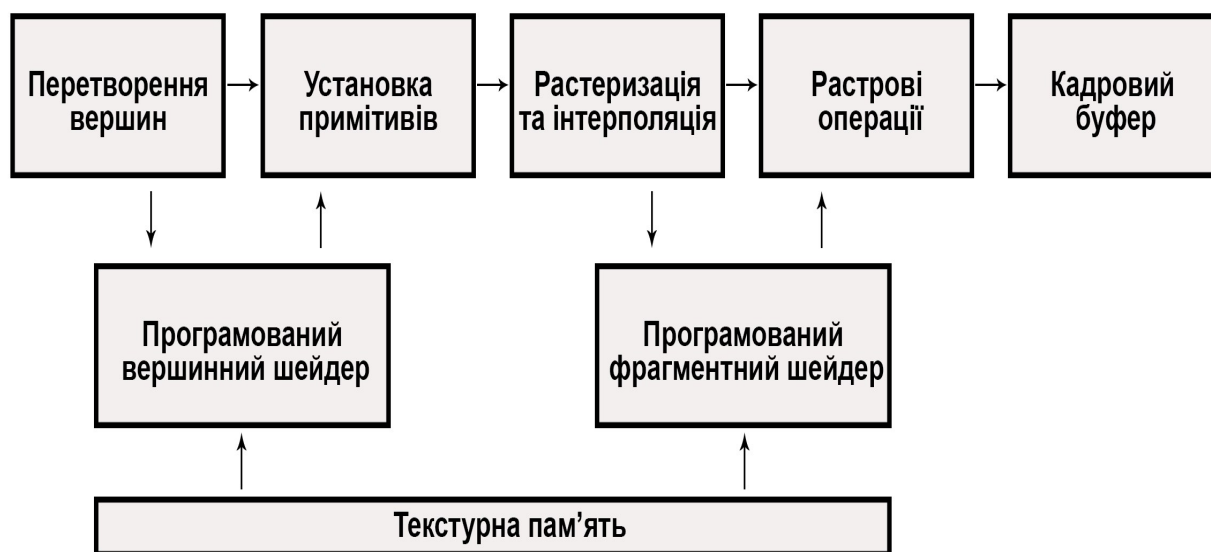
**Рисунок 3 - 3D конвеєр відеокарт другого покоління**

Відеокарти третього покоління (GeForce3/Radeon 8500 2001 рік) характеризуються появою перших програмованих шейдерних блоків для виконання T&L обчислень. Крім того нові текстурні блоки давали можливість об'ємного текстуровання та мультисемплінгу.



**Рисунок 4 - 3D конвеєр відеокарт третього покоління**

Четверте покоління (Radeon 9700/GeForce FX 2002 рік) відеокарт стало першим повністю програмованим поколінням після появи програмованих фрагментних процесорів для виконання операцій зафарбовування та текстурювання. З 2004 року у відеокартах GeForce6/X800 з'явилася можливість доступу до текстурної пам'яті з вершинних шейдерів, що дало можливість використовувати їх у процесі текстурювання.



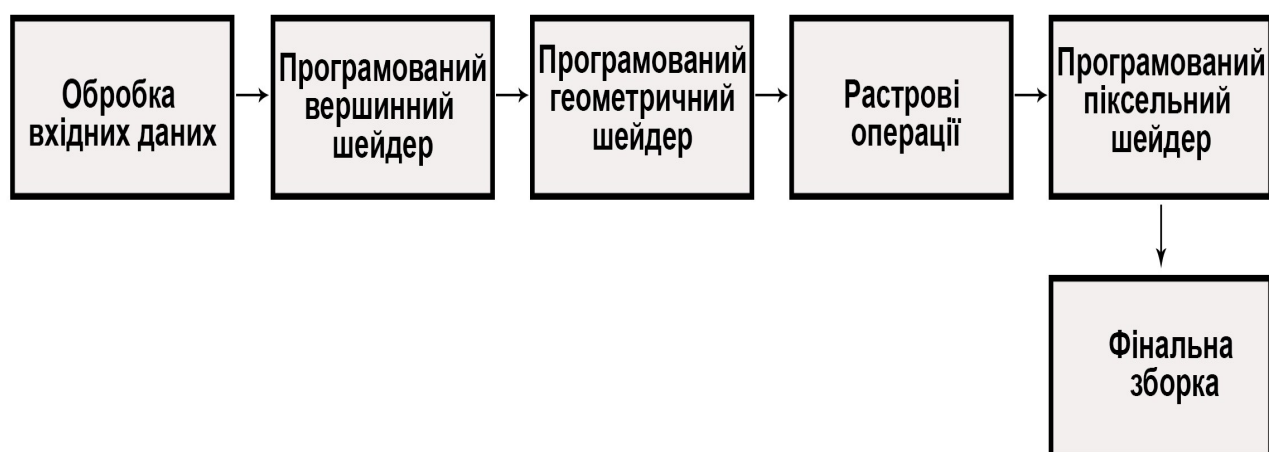
**Рисунок 5 - 3D конвеєр відеокарт четвертого покоління**

П'яте (2006 рік), покоління відеокарт побудовані на базі уніфікованої шейдерної архітектури. Це означає, що усі обчислювальні блоки графічного процесора «уніфіковані», тобто здатні виконувати будь-який тип шейдерної програми.

Такі графічні процесори складені з масиву обчислювальних блоків і блоків динамічного планування завантаження, для розподілу виконання шейдерних програм між усіма обчислювальними блоками.

Уніфікована шейдерна архітектура дозволяє більш гнучко використовувати ресурси графічного процесора. Наприклад, в умовах із симуляцією важкої геометрії уніфікована шейдерна архітектура може задіяти всі блоки графічного процесора для обчислення вершин і геометричних шейдерів.

І навпаки, коли геометрія не є складаною, а симулює складні піксельні ефекти, такі як parallax occlusion mapping, система часток і т. д., всі обчислювальні блоки можуть бути спрямовані на виконання піксельних шейдерів. Уніфіковану шейдерну архітектуру підтримують графічні процесори починаючи з: GeForce 8, Radeon R600, S3 Graphics Chrome 400, Intel GMA X3000.



**Рисунок 6 - 3D конвеєр відеокарт п'ятого покоління**

### **Висновки**

Швидкість та якість обробки тривимірної сцени суттєво залежать від досконалості як програмного, так і апаратного забезпечення. Сукупність апаратних і програмних засобів обробки тривимірних сцен утворюють графічний конвеєр.

Етапи графічного конвеєру в конкретних реалізаціях можуть бути переставлені, розділені, об'єднані, виконуватися в кілька проходів, проте їх зміст залишається незмінним. В процесі еволюції графічні чіпсети перейшли від суто апаратної реалізації обчислювальних блоків до уніфікованої шейдерної архітектури. В сучасних графічних прискорювачах обчислювальні блоки проектують так, щоб будь-який з них міг виконати будь-який тип програмованого шейдера.

### Список використаної літератури

1. Романюк О.Н. Високопродуктивні методи та засоби зафарбовування тривимірних графічних об'єктів. Монографія. // О.Н. Романюк - Вінниця: УНІВЕСУМ-Вінниця, 2006
2. Вяткин С. И. Анизотропная фильтрации текстуры в реальном времени [Текст] / С. И. Вяткин, А. Н. Романюк, А. А. Дудник // Измерительная и вычислительная техника в технологических процессах. — 2015. — № 4. — С. 217-221.
3. Романюк О.Н. Аналіз архітектур графічних відеокарт / О. Н. Романюк, Д. Т. Обідник, О. В. Поліщук, П. О. Величко // П'ята міжнародна науково-технічна конференція "Моделювання та комп'ютерна графіка", Донецьк, 24-27 вересня 2013 р. – Донецьк : ДонНТУ, 2013. — С. 132–138.
4. Romanyuk S.A. Function-based gpu architecture // S.A. Romanyuk, S.V. Pavlov, O.O. Dudnyk - Вимірювальна та обчислювальна техніка в технологічних процесах No 1' 2015
5. Романюк О. Н. Аналіз архітектур відеокарт компанії NVIDIA / О. Н. Романюк, О. В. Даньковська, С. І. Вяткін // Збірник матеріалів Міжнародної науково-практичної Інтернет-конференції "Електронні ресурси: створення, використання, доступ", Вінниця, грудень 2014 р. — 2014. — С. 3-15.