

Робоча програма дисципліни «Алгоритмічні основи комп'ютерної графіки»

СПЕЦІАЛЬНІСТЬ 123 · КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

БАКАЛАВР

120

Годин

Загальний обсяг дисципліни

4

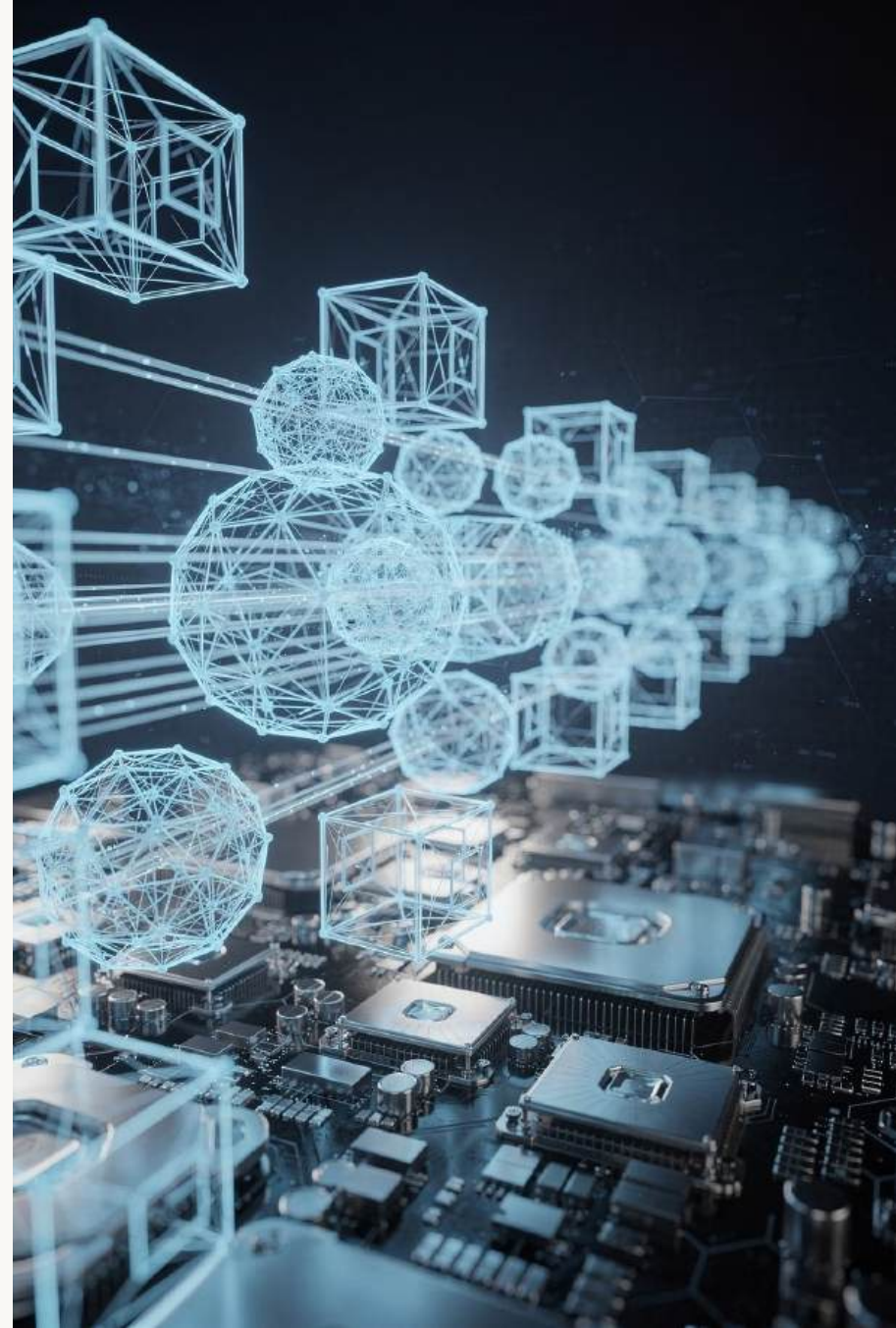
Кредити ЄКТС

Відповідно до стандарту

4

Модулі

Змістовних блоки курсу



Загальна інформація про дисципліну

Ідентифікація

Назва: Алгоритмічні основи комп'ютерної графіки

Спеціальність: 123 Комп'ютерна інженерія

Ступінь: Бакалавр

Обсяг: 120 годин / 4 кредити ЄКТС

Модулів: 4 змістовних блоки

Лекції: 15 год. · **Лабораторні:** 45 год. · **СРС:** 60 год.

Призначення дисципліни

Дисципліна «Алгоритмічні основи комп'ютерної графіки» формує у студентів системне розуміння математичних та алгоритмічних основ побудови та обробки графічних зображень. Курс охоплює растрові та векторні алгоритми, геометричні перетворення, алгоритми видимості, рендерингу та освітлення, а також сучасні технології GPU-програмування.

Дисципліна є профільною для підготовки бакалаврів зі спеціальності «Комп'ютерна інженерія» та забезпечує формування фахових компетентностей у сфері розробки графічних застосунків і систем візуалізації.

Загальні компетентності

Аналітичне мислення

Здатність до аналізу, синтезу та критичного оцінювання алгоритмів комп'ютерної графіки: порівняння ефективності методів, виявлення переваг і недоліків, обґрунтування вибору оптимального підходу для конкретної задачі.

Комунікація

Здатність до усної та письмової технічної комунікації: чітко пояснювати алгоритми, їх складність та реалізацію, оформлювати звіти з лабораторних робіт, аналітичні записки та технічну документацію.

Самоорганізація

Уміння самостійно навчатися, планувати роботу та організовувати виконання завдань у визначені строки, зокрема під час розробки та налагодження програмних реалізацій графічних алгоритмів.

Командна робота

Відповідальність, ініціативність та готовність до спільної роботи над груповими проектами з розробки графічних застосунків, розподілу задач та інтеграції програмних модулів.

Студент повинен вміти: аналізувати алгоритми та оцінювати їх обчислювальну складність; обирати оптимальний метод для заданої графічної задачі; програмувати алгоритми на сучасних платформах; документувати рішення та презентувати результати; працювати в команді при розробці програмних проектів.

Фахові компетентності

Математичні основи графіки

Розуміння математичного апарату комп'ютерної графіки: векторна та матрична алгебра, системи координат, афінні та проєктивні перетворення, параметричні криві та поверхні. Вміння застосовувати математичний інструментарій при реалізації алгоритмів.

Растрові алгоритми

Здатність реалізовувати та аналізувати растрові алгоритми: побудова ліній (Брезенхем, Ву), кіл, еліпсів; алгоритми заповнення областей; методи відсічення; антиаліасинг. Оцінка ефективності алгоритмів за часом та пам'яттю.

Тривимірна графіка та рендеринг

Знання методів 3D-моделювання, проектування та рендерингу: побудова та перетворення 3D-об'єктів, алгоритми видалення невидимих ліній і поверхонь, моделі освітлення (Ламберт, Фонг), методи тонування, трасування променів.

Сучасні технології графіки

Орієнтування в сучасних графічних API (OpenGL, Vulkan) та архітектурі GPU; розробка шейдерних програм; розуміння конвеєра рендерингу; застосування сучасних методів реального часу для ігрових та промислових застосунків.

Результати навчання

01

Пояснення базових понять

Пояснювати сутність ключових понять: растр, піксель, вектор, колірна модель, геометричне перетворення, проекція, рендеринг, шейдер, конвеєр графічного процесора, антиаліасинг, текстуровання.

03

Розробка графічних застосунків

Розробляти графічні застосунки із застосуванням сучасних API (OpenGL/Vulkan); програмувати вершинні та фрагментні шейдери; налагоджувати та профілювати графічний конвеєр.

02

Реалізація алгоритмів

Реалізовувати класичні та сучасні алгоритми комп'ютерної графіки мовою програмування (C++/Python/GLSL); аналізувати обчислювальну складність; порівнювати альтернативні підходи та обирати оптимальний.

04

Робота з фаховою літературою

Орієнтуватися у фаховій науковій та технічній літературі з комп'ютерної графіки; використовувати стандарти, специфікації та дослідницькі матеріали для розробки та вдосконалення графічних систем.

Математичні основи та примітиви комп'ютерної графіки

1

Тема 1.1

Вступ до комп'ютерної графіки. Колірні моделі та системи координат.

2

Тема 1.2

Математичний апарат: вектори, матриці, афінні перетворення на площині.

3

Тема 1.3

Растрові алгоритми побудови примітивів: лінії, кола, еліпси.

4

Тема 1.4

Алгоритми відсічення та заповнення полігонів. Антиаліасинг.

Вступ до комп'ютерної графіки. Колірні моделі

Лекційний матеріал (2 год.)

Лекція 1: Предмет і задачі курсу. Галузі застосування комп'ютерної графіки. Класифікація: растрова, векторна, фрактальна, 3D-графіка. Пристрої введення та виведення графічної інформації. Архітектура графічної підсистеми комп'ютера.

Колірні моделі: фізична природа кольору. Модель RGB: представлення кольору, колірний куб, глибина кольору. Модель CMY/CMYK для поліграфії. Моделі HSV та HSL: відтінок, насиченість, яскравість. Перетворення між колірними моделями. Гамма-корекція.

Ключові питання лекції

- Класифікація та галузі застосування комп'ютерної графіки
- Пристрої введення та виведення: характеристики та принципи роботи
- Колірна модель RGB та її програмне представлення
- Порівняння колірних моделей RGB, CMY, HSV, HSL
- Поняття глибини кольору та гамма-корекції

Лабораторна робота (3 год.)

Завдання 1. Налаштування середовища розробки (C++/OpenGL або Python/Pillow). Реалізація програми для відображення колірного куба RGB та демонстрації колірних градієнтів у різних моделях.

Завдання 2. Реалізація функцій перетворення між колірними моделями RGB ↔ HSV ↔ HSL. Побудова колірного колеса та демонстрація гамма-корекції.

Завдання 3. Тестування та оформлення звіту.

Самостійна робота (8 год.)

- Опрацювання: Пічугін М. Ф., Канкін І. О. «Комп'ютерна графіка» — розділ «Основи та колірні моделі» (2023)
- Підготовка реферату: «Сучасні дисплейні технології та колірний простір sRGB/HDR»
- Складання порівняльної таблиці колірних моделей та їх застосування



Очікуваний результат: студент вміє характеризувати архітектуру графічних систем, реалізовувати перетворення між колірними моделями та застосовувати гамма-корекцію.

Математичний апарат: вектори, матриці, афінні перетворення

1

Векторна алгебра

Вектори, скалярний і векторний добутки. Нормалізація. Застосування у графіці.

2

Матрична алгебра

Матриці перетворень. Множення матриць. Однорідні координати (homogeneous).

3

Афінні перетворення

Перенесення, масштабування, обертання, відображення, зсув на площині 2D.

4

Складені перетворення

Комбінування матриць перетворень. Порядок множення. Зворотне перетворення.

Лекційний матеріал (1 год.)

Векторна алгебра у задачах графіки. Матриці як засіб компактного запису перетворень. Однорідні координати: розширений простір, переваги представлення. Матриці 2D-перетворень: перенесення T , масштабування S , обертання R , відображення, зсув. Складені перетворення та порядок множення матриць. Обернені перетворення.

Лабораторна робота (3 год.) та СРС (8 год.)

Лабораторна: Реалізація бібліотеки 2D-перетворень у однорідних координатах. Демонстрація послідовних та складених перетворень графічних об'єктів з інтерактивним керуванням параметрами.

СРС: Опрацювання: Маценко В. Г. «Комп'ютерна графіка: навч. посіб.» (вид. 2-ге, оновл., 2023) — розділ «Математичні основи»; підготовка конспекту «Застосування однорідних координат у графічних конвеєрах».

Растрові алгоритми побудови примітивів

Лекційний матеріал (1 год.)

Растрова розгортка (scan conversion). Алгоритм ЦДА (цифрового диференціального аналізатора) для побудови ліній: принцип, реалізація, недоліки. Алгоритм Брезенхема для ліній: ціла арифметика, висока продуктивність. Алгоритм Ву для ліній з антиаліасингом. Алгоритм Брезенхема для кіл та еліпсів: принцип симетрії, критерій вибору пікселя. Порівняльний аналіз алгоритмів за складністю та якістю.

Ключові питання

- Принцип дискретизації геометричних об'єктів на растрі
- Алгоритм ЦДА: переваги та недоліки
- Алгоритм Брезенхема: чому тільки ціла арифметика
- Алгоритм Ву та субпіксельна точність
- Алгоритм Брезенхема для кіл та використання симетрії

Лабораторна робота (4 год.)

Завдання 1. Реалізація та порівняння алгоритмів ЦДА та Брезенхема для відрізків. Вимірювання часу виконання для різних довжин відрізків (бенчмарк).

Завдання 2. Реалізація алгоритму Брезенхема для кіл та еліпсів з використанням восьмисторонньої симетрії. Демонстрація растеризації для різних радіусів.

Завдання 3. Реалізація алгоритму Ву для ліній з антиаліасингом. Візуальне порівняння якості.

Самостійна робота (8 год.)

- Опрацювання: Тотосько О. В., Микитишин А. Г., Стухляк П. Д. «Комп'ютерна графіка: навч. посіб.» — розділ «Растрові алгоритми» (ТНТУ, 2022)
- Підготовка аналітичного звіту «Порівняльний аналіз растрових алгоритмів побудови ліній»
- Вивчення матеріалів щодо сучасних GPU-реалізацій растеризації примітивів



Очікуваний результат: студент вміє реалізовувати та аналізувати растрові алгоритми побудови геометричних примітивів, оцінювати їх ефективність.

Алгоритми відсічення та заповнення полігонів

Лекційний матеріал (1 год.)

Алгоритми відсічення відрізків: Коена–Сазерленда (коди регіонів), Ліанга–Барські (параметричне відсічення).
Алгоритми відсічення полігонів: Сазерленда–Годмана.
Алгоритми заповнення полігонів: сканування рядків (scan-line fill), алгоритм заливки (flood fill, seed fill). Парність перетинів.
Антиаліасинг полігонів: надвибірка (supersampling, MSAA).

Ключові питання

- Алгоритм Коена–Сазерленда: коди регіонів та логіка відсічення
- Алгоритм Ліанга–Барські: параметричне представлення відрізка
- Алгоритм заповнення скануванням рядків: таблиця ребер
- Flood fill: рекурсивний та ітеративний варіанти
- MSAA та сучасні методи антиаліасингу (TAA, DLAA)

Лабораторна робота (4 год.)

Завдання 1. Реалізація алгоритму відсічення відрізків Коена–Сазерленда та Ліанга–Барського з інтерактивним налаштуванням вікна відсічення.

Завдання 2. Реалізація алгоритму заповнення полігону скануванням рядків. Демонстрація для випуклих та невивуклих полігонів.

Завдання 3. Реалізація flood fill (стекова версія). Порівняння з scan-line fill за продуктивністю.

Самостійна робота (8 год.)

- Опрацювання: Тотосько О. В., Микитишин А. Г., Стухляк П. Д. «Комп'ютерна графіка» — розділ «Відсічення та заповнення» (ТНТУ, 2022)
- Підготовка звіту «Сучасні методи антиаліасингу у реальному часі (MSAA, TAA, DLAA)»
- Дослідження алгоритму Сазерленда–Годмана для відсічення полігонів

Тривимірна графіка: перетворення, проекції та ВИДИМІСТЬ

Тема 2.1

Тривимірні геометричні перетворення. Матриці 3D-перетворень. Однорідні координати у просторі.

Тема 2.2

Проекції: паралельна (ортографічна, аксонометрична) та перспективна. Матриці проекцій.

Тема 2.3

Алгоритми видалення невидимих ліній і поверхонь. Z-буфер, BSP-дерева, алгоритм художника.

Тема 2.4

Параметричні криві та поверхні: Безьє, B-сплайни, NURBS. Тесселяція.

Тривимірні геометричні перетворення

Лекційний матеріал (1 год.)

Розширення 2D-перетворень на тривимірний простір. Однорідні координати у 3D: вектор (x, y, z, w) . Матриці 3D-перетворень: перенесення, масштабування, обертання навколо осей X , Y , Z . Формули Родрига для обертання навколо довільної осі. Кути Ейлера та проблема гімбал-лок. Кватерніони: визначення, множення, представлення обертання. Складені 3D-перетворення. Система координат камери: вектори *right*, *up*, *forward*.

Ключові питання

- Матриці обертання у 3D та їх властивості
- Формула Родрига для обертання навколо довільної осі
- Проблема гімбал-лок та шляхи її усунення
- Кватерніони: математичне визначення та перевага над кутами Ейлера
- Побудова матриці виду (*view matrix*) для камери

Лабораторна робота (4 год.)

Завдання 1. Реалізація класу матриць 4×4 із підтримкою всіх 3D-перетворень. Демонстрація складених перетворень на тривимірній сцені.

Завдання 2. Реалізація кватерніонних обертань. Порівняння інтерполяції LERP та SLERP між двома орієнтаціями. Демонстрація проблеми гімбал-лок.

Завдання 3. Реалізація класу Camera з матрицею виду, підтримкою орбітального керування.

Самостійна робота (8 год.)

- Опрацювання: Пічугін М. Ф., Канкін І. О., Воротніков В. В. «Комп'ютерна графіка» — розділ «3D-перетворення та кватерніони» (2023)
- Підготовка реферату: «Кватерніони у 3D-графіці та ігрових рушіях»
- Дослідження реалізації 3D-перетворень у бібліотеці GLM (OpenGL Mathematics)



Очікуваний результат: студент вміє застосовувати матриці та кватерніони для 3D-перетворень, реалізовувати систему координат камери.

Проекції: паралельна та перспективна



Лекційний матеріал (1 год.)

Класифікація проекцій. Паралельна: ортографічна матриця. Аксонометрична: ізометрична, диметрична. Перспективна проекція: точки сходу (1, 2, 3), матриця перспективи (FOV, aspect, near, far). NDC-простір. Конвеєр перетворень MVP. Відсічення за frustum. Перетворення viewport.

Лабораторна робота (4 год.) та СРС (8 год.)

Лабораторна: Реалізація та порівняння ортографічної та перспективної проекцій. Реалізація повного конвеєра MVP з інтерактивним керуванням кутом огляду (FOV). Візуалізація frustum culling.

СРС: Опрацювання: Маценко В. Г. «Комп'ютерна графіка» (2-ге вид., 2023) — розділ «Проекції та конвеєр перетворень»; підготовка порівняльної таблиці видів проекцій та їх застосувань у CAD, іграх, VR.

Алгоритми видалення невидимих поверхонь

Лекційний матеріал (1 год.)

Проблема видимості у 3D. Алгоритм художника (painter's algorithm): сортування полігонів по глибині, обмеження. Z-буфер (depth buffer): принцип роботи, реалізація у GPU, проблема z-fighting. Відсічення за нормаллю (back-face culling): dot product нормалі та вектора погляду. BSP-дерева (Binary Space Partitioning): побудова, обхід для відсортованого рендерингу. Метод порталної видимості (portal culling). НОМ (Hierarchical Occlusion Maps). Сучасні підходи: precomputed visibility.

Ключові питання

- Алгоритм художника: коли він не працює
- Z-буфер: переваги та проблема точності (z-fighting)
- Back-face culling: математичне обґрунтування
- BSP-дерева: побудова та застосування у Quake-рушіях
- Сучасні GPU-методи відсічення невидимої геометрії

Лабораторна робота (4 год.)

Завдання 1. Реалізація z-буфера для software-рендерингу. Демонстрація проблеми z-fighting та методів її усунення (polygon offset, reversed-z).

Завдання 2. Реалізація back-face culling. Вимірювання приросту продуктивності на сценах із різною кількістю полігонів.

Завдання 3. Побудова та обхід BSP-дерева для простої 2D-сцени. Перенесення концепції на 3D.

Самостійна робота (8 год.)

- Опрацювання: Тотосько О. В., Микитишин А. Г., Стухляк П. Д. «Комп'ютерна графіка» — розділ «Видимість і відсічення» (ТНТУ, 2022)
- Підготовка реферату: «Алгоритми видимості у сучасних ігрових рушіях (Unreal Engine, Unity)»
- Аналіз реалізації occlusion culling у відкритих ігрових проектах

Параметричні криві та поверхні

Лекційний матеріал (1 год.)

Параметричне представлення кривих. Криві Безьє: визначення, формула Бернштейна, алгоритм де Кастельжо, властивості (опукла оболонка, варіаційне зменшення). Кубічні криві Безьє та матрична форма. В-сплайни: переваги над Безьє, вузловий вектор, базисні функції. NURBS (Non-Uniform Rational B-Splines): раціональне представлення, кола та коніки. Поверхні Безьє та В-сплайнові поверхні. Тесселяція: адаптивна, рівномірна. Geometry shader та tessellation shader у сучасному OpenGL.

Лабораторна робота (4 год.) та СРС (8 год.)

Лабораторна: Реалізація інтерактивного редактора кривих Безьє із алгоритмом де Кастельжо. Демонстрація кривих різних степенів. Реалізація В-сплайнів та порівняння з Безьє. Тесселяція поверхні Безьє.

СРС: Опрацювання: Маценко В. Г. «Комп'ютерна графіка» (2023) — розділ «Параметричні криві та поверхні»; підготовка аналітичного звіту «NURBS у CAD-системах: теорія та практика застосування»; дослідження реалізації tessellation shaders у OpenGL 4.x.

- ❑ **Очікуваний результат:** студент вміє реалізовувати параметричні криві та поверхні, застосовувати алгоритм де Кастельжо, порівнювати Безьє та В-сплайни.

Моделі освітлення, тонування та текстуровання

Тема 3.1

Фізичні основи освітлення. Локальні моделі: Ламберта, Фонга, Блінна–Фонга. Глобальне освітлення.

Тема 3.2

Алгоритми тонування поверхонь: плоске, Гуро, Фонг-тонування. Нормалі та їх інтерполяція.

Тема 3.3

Текстуровання: UV-розгортка, фільтрація текстур, міпмепінг, процедурні текстури.

Тема 3.4

Карти нормалей, тіней та відображень. PBR (Physically Based Rendering): металевість, шорсткість.

Фізичні основи освітлення. Локальні та глобальні моделі

Лекційний матеріал (1 год.)

Фізика взаємодії світла з поверхнею: дифузне та дзеркальне відбиття, заломлення, поглинання. BRDF (Bidirectional Reflectance Distribution Function): визначення, властивості. Модель Ламберта: ідеальне дифузне відбиття, закон косинуса. Модель Фонга: амбієнтна, дифузна та дзеркальна компоненти, показник дзеркальності. Модель Блінна–Фонга: напівдиференціальний вектор, ефективність. Джерела світла: точкове, спрямоване, прожектор (spot light). Затухання інтенсивності зі відстанню. Глобальне освітлення: передумови, відмінність від локального.

Лабораторна робота (4 год.)

Завдання 1. Реалізація моделі освітлення Ламберта у software-рендерингу. Демонстрація впливу кута падіння світла на яскравість поверхні.

Завдання 2. Реалізація повної моделі Фонга та Блінна–Фонга. Порівняння візуального результату та часу обчислення. Налаштування параметрів матеріалу.

Завдання 3. Реалізація різних типів джерел світла (точкове, спрямоване, прожектор) з урахуванням затухання.

Самостійна робота (8 год.)

- Опрацювання: Гуржій А. М. та ін. «Основи інформаційних технологій» — розділ «Комп'ютерна графіка та моделі освітлення» (Літера ЛТД, 2023)
- Підготовка реферату: «BRDF та фізично коректне освітлення у сучасних рендерерах»
- Аналіз моделей освітлення у Blender Cycles та Unreal Engine Lumen



Очікуваний результат: студент вміє реалізовувати локальні моделі освітлення, налаштовувати параметри матеріалів та джерел світла.

Алгоритми тонування поверхонь

Лекційний матеріал (1 год.)

Плоске тонування (flat shading): одна нормаль на полігон, швидкість, артефакти. Тонування Гуро: інтерполяція кольорів у вершинах, бікварна апроксимація, проблема з дзеркальними відбликами. Тонування Фонга: інтерполяція нормалей у кожному пікселі, реалістичність, обчислювальна вартість. Порівняльний аналіз методів тонування. Вершинний та фрагментний шейдери у контексті методів тонування. Нормалі: обчислення на основі геометрії, усереднення нормалей у вершинах.

Лабораторна робота (3 год.) та СРС (8 год.)

Лабораторна: Реалізація трьох методів тонування (flat, Gouraud, Phong) в OpenGL через шейдери GLSL. Порівняльна демонстрація на сфері та складній поверхні. Вимірювання FPS для кожного методу.

СРС: Опрацювання: Тотосько О. В., Микитишин А. Г., Стухляк П. Д. «Комп'ютерна графіка» — розділ «Тонування поверхонь» (ТНТУ, 2022); підготовка аналітичного звіту «Вплив методу тонування на якість і продуктивність: від flat до Phong»; дослідження toon shading як нефотореалістичного методу.

- ❑ **Очікуваний результат:** студент вміє реалізовувати методи тонування через GLSL-шейдери та порівнювати їх за якістю та продуктивністю.

Текстурування та сучасні техніки матеріалів

Тема 3.3 — Текстурування (Лекція, 1 год.)

Принцип текстурування: відображення 2D-текстури на 3D-поверхню. UV-координати та UV-розгортка. Фільтрація текстур: nearest-neighbor, білінійна, трилінійна. Міпмепінг: побудова міп-ланцюжка, анізотропна фільтрація. Процедурні текстури: шум Перліна, шум Ворона. Cubemap та environment mapping. Текстурний атлас та sprite sheets.

Лабораторна (3 год.): Реалізація текстурування у OpenGL: завантаження зображення, UV-розгортка для куба та сфери. Порівняння методів фільтрації (nearest, bilinear, trilinear, anisotropic). Генерація процедурної текстури за шумом Перліна.

СРС (8 год.): Опрацювання: Маценко В. Г. «Комп'ютерна графіка» (2023) — розділ «Текстурування та фільтрація»; підготовка звіту «Процедурні текстури та шум Перліна у генерації ігрових світів».

Тема 3.4 — PBR та карти матеріалів (Лекція, 1 год.)

Карта нормалей (normal map): tangent-space vs. object-space, алгоритм tangent-space обчислення. Карта тіней (shadow map): алгоритм двопрохідного рендерингу, артефакти, PCF-фільтрація. PBR (Physically Based Rendering): металевість (metallic), шорсткість (roughness), карта оклюзії (AO). Рівняння рендерингу Кука–Торранса. Diffuse та specular BRDF. IBL (Image-Based Lighting): HDR-карти навколишнього середовища.

Лабораторна (4 год.): Реалізація PBR-матеріалу у GLSL: завантаження albedo, normal, metallic, roughness, AO карт. Реалізація shadow mapping з PCF. Порівняння Phong та PBR на однаковій сцені.

СРС (8 год.): Опрацювання: Пічугін М. Ф., Канкін І. О. «Комп'ютерна графіка» (2023) — розділ «PBR та сучасні матеріали»; підготовка аналітичного звіту «PBR у реальному часі: від теорії до реалізації у Unity та Unreal Engine».

Алгоритми рендерингу та сучасні технології GPU

Тема 4.1

Трасування променів (Ray Tracing): алгоритм, пересічення з об'єктами, відбиття, заломлення, тіні.

Тема 4.2

Глобальне освітлення: Path Tracing, метод Монте-Карло, фотонне зіставлення, радіозитет.

Тема 4.3

Архітектура GPU та графічний конвеєр OpenGL/Vulkan.
Шейдерне програмування GLSL/HLSL.

Тема 4.4

Сучасні методи та тенденції: DLSS/FSR, трасування на GPU (DXR), обчислювальні шейдери.

Трасування променів (Ray Tracing)

Лекційний матеріал (1 год.)

Ідея зворотного трасування променів. Алгоритм Whitted (1980): первинні промені, тіньові, відбиті, заломлені. Математика пересічення променя з примітивами: сфера (аналітичне), трикутник (алгоритм Мюллера–Трумбора), AABB. Обчислення нормалі у точці перетину. Рекурсивне трасування: рівні рекурсії, обрізання (Russian roulette). Прискорювальні структури: BVH (Bounding Volume Hierarchy), kd-дерева. Складність алгоритму. Переваги над растеризацією.

Ключові питання

- Принцип зворотного трасування: чому промені йдуть від камери
- Алгоритм Мюллера–Трумбора: пересічення з трикутником
- BVH: побудова дерева та обхід при пересіченні
- Рекурсивне трасування: відбиття та заломлення
- Порівняння ray tracing та растеризації за якістю та швидкістю

Лабораторна робота (4 год.)

Завдання 1. Реалізація базового ray caster (без відбиття): первинні промені, перетин зі сферами, дифузне освітлення. Рендеринг простої сцени у PPM-файл.

Завдання 2. Розширення до ray tracer: відбиті та тіньові промені. Рекурсія до 4 рівнів. Матеріали: дзеркальний, скляний, матовий.

Завдання 3. Реалізація BVH для прискорення. Вимірювання speedup на сцені з 1000+ об'єктів.

Самостійна робота (8 год.)

- Опрацювання: Тотосько О. В., Микитишин А. Г., Стухляк П. Д. «Комп'ютерна графіка» — розділ «Трасування променів» (ТНТУ, 2022)
- Підготовка реферату: «Апаратне прискорення трасування променів: NVIDIA RTX та DirectX Raytracing (DXR)»
- Аналіз відкритого Ray Tracing проекту (Peter Shirley «Ray Tracing in One Weekend»)



Очікуваний результат: студент вміє реалізовувати алгоритм трасування променів, будувати BVH та оцінювати вплив прискорювальних структур на продуктивність.

Глобальне освітлення: Path Tracing та радіозитет

Лекційний матеріал (1 год.)

Рівняння рендерингу Кейджіа. Метод Монте-Карло для чисельного інтегрування: дисперсія, стратифіковане семплювання, важлива вибірка (importance sampling). Path Tracing: ідеальна реалізація рівняння рендерингу, конвергенція. Bidirectional Path Tracing. Метод фотонного зіставлення (photon mapping): фаза трасування фотонів, фаза рендерингу. Радіозитет (radiosity): дифузне глобальне освітлення, матриця форм-факторів. Кеш іррадіансу. Сучасні реалізації: ReSTIR, DDGI.

Лабораторна робота (4 год.) та СРС (8 год.)

Лабораторна: Реалізація Path Tracer на основі попереднього ray tracer. Реалізація косинусного семплювання півсфери. Демонстрація конвергенції зі збільшенням кількості семплів (1, 4, 16, 64, 256 spp). Порівняння direct illumination та full global illumination.

СРС: Опрацювання: Маценко В. Г. «Комп'ютерна графіка» (2023) — розділ «Глобальне освітлення»; підготовка аналітичного звіту «Path Tracing у виробничому рендерингу: Cycles, Arnold, Octane — порівняльний аналіз»; дослідження реалізації ReSTIR для рендерингу в реальному часі.

❏ **Очікуваний результат:** студент вміє реалізовувати Path Tracing, застосовувати метод Монте-Карло та оцінювати якість глобального освітлення.

Архітектура GPU та шейдерне програмування



Архітектура GPU

Потокові мультипроцесори (SM), SIMD-виконання, варп (warp), окупантність (occupancy).



Графічний конвеєр

Вершинний → Тесселяція →
Геометричний → Растеризація →
Фрагментний → Framebuffer.



GLSL-шейдери

Вершинний та фрагментний шейдери. Uniform-змінні. Атрибути. Varying-інтерполяція.

Лекційний матеріал (1 год.)

Архітектура сучасного GPU: потокові мультипроцесори, варпи, SIMD. Порівняння CPU та GPU: паралелізм, пам'ять. Графічний конвеєр OpenGL 4.x: стадії, програмовані та фіксовані. Мова GLSL: типи даних, вбудовані функції, qualifier'и. Вершинний шейдер: `gl_Position`. Фрагментний шейдер: `out`-колір. VAO/VBO/EBO. Vulkan: чому він складніший і потужніший за OpenGL. Compute шейдери: GPGPU-обчислення.

Лабораторна робота (5 год.) та СРС (8 год.)

Лабораторна: Розробка повноцінного OpenGL-застосунку: завантаження геометрії (OBJ), PBR-шейдери, skybox, тіні (shadow map), bloom-ефект через framebuffer. Профілювання за допомогою RenderDoc.

СРС: Опрацювання: Пічугін М. Ф., Канкін І. О. «Комп'ютерна графіка» (2023) — розділ «GPU та шейдерне програмування»; підготовка аналітичного звіту «Порівняння OpenGL та Vulkan: архітектурні відмінності та вибір API для проекту».

Сучасні методи: DLSS, FSR, AI-рендеринг та тенденції

Лекційний матеріал (1 год.)

Проблема продуктивності рендерингу у реальному часі. Методи апскейлінгу: DLSS (Deep Learning Super Sampling, NVIDIA), FSR (FidelityFX Super Resolution, AMD), XeSS (Intel). Порівняння якості та обчислювальних витрат. Апаратна підтримка Ray Tracing (DXR, Vulkan RT Extensions). Гібридний рендеринг: растеризація + ray tracing. Compute-шейдери для нетрафічних задач. Mesh shaders. AI-деноїзинг. Тенденції: нейронні мережі у графіці, NeRF (Neural Radiance Fields), Gaussian Splatting.

Ключові питання

- DLSS vs FSR: принципи та якість апскейлінгу
- Гібридний рендеринг: де доцільно застосовувати ray tracing
- Compute shaders: GPGPU поза графічним конвеєром
- NeRF: нейронні поля яскравості та їх застосування
- Майбутнє рендерингу: AI та нейронна графіка

Лабораторна робота (5 год.)

Завдання 1. Реалізація compute-шейдера для GPU-паралельного обчислення: генерація фракталу Мандельброта, порівняння з CPU-версією за часом.

Завдання 2. Реалізація простого деноїзера (bilateral filter) для зашумленого Path Tracer-виводу. Оцінка метриками PSNR та SSIM.

Завдання 3. Підготовка та захист фінального проекту: розробка інтерактивної 3D-сцени з PBR, тінями, постпроцесингом (bloom, HDR tonemapping, FXAA).

Самостійна робота (8 год.)

- Опрацювання: Гуржій А. М. та ін. «Основи інформаційних технологій» — розділ «Сучасні технології візуалізації» (Літера ЛТД, 2023)
- Підготовка аналітичної записки «NeRF та Gaussian Splatting: нейронні методи у 3D-реконструкції»
- Аналіз технічних специфікацій DLSS 3 та FSR 3 та порівняння їх ефективності



Очікуваний результат: студент вміє застосовувати compute-шейдери, орієнтується у сучасних тенденціях AI-рендерингу та може аналізувати новітні технології галузі.

Розподіл навчального часу — 120 годин

Модуль	Тема	Лекції (год.)	Лабор. (год.)	СРС (год.)
Модуль 1	Тема 1.1 — Колірні моделі та системи координат	2	3	8
	Тема 1.2 — Матриці та афінні перетворення 2D	1	3	8
	Тема 1.3 — Растрові алгоритми примітивів	1	4	8
	Тема 1.4 — Відсічення та заповнення полігонів	1	4	8
Модуль 2	Тема 2.1 — Тривимірні перетворення та кватерніони	1	4	8
	Тема 2.2 — Проекції та конвеєр MVP	1	4	8
	Тема 2.3 — Алгоритми видалення невидимих поверхонь	1	4	8
	Тема 2.4 — Параметричні криві та поверхні	1	4	8
Модуль 3	Тема 3.1 — Моделі освітлення (Ламберт, Фонг, PBR)	1	4	8
	Тема 3.2 — Алгоритми тонування (flat, Gouraud, Phong)	1	3	8
	Тема 3.3 — Текстурування та фільтрація	1	3	4
	Тема 3.4 — Карти матеріалів та PBR	1	4	4
Модуль 4	Тема 4.1 — Трасування променів (Ray Tracing)	1	4	8
	Тема 4.2 — Глобальне освітлення та Path Tracing	1	4	8
	Тема 4.3 — Архітектура GPU та GLSL-шейдери	1	5	8
	Тема 4.4 — Сучасні методи та тенденції (DLSS, AI)	1	5	8
Разом	16 тем у 4 модулях	15	45	60



Загальний обсяг: 120 годин = 4 кредити ЄКТС. Розподіл є орієнтовним і може коригуватися відповідно до специфіки навчального процесу.

Структура навчального часу



Коментар до розподілу

Загальний обсяг — **120 годин (4 кредити ЄКТС)** — розподілено між трьома формами роботи для забезпечення балансу між теорією та практичними навичками програмування.

Лекції — 15 год. (12,5%)

Системне подання ключових алгоритмів, математичного апарату та теоретичних основ комп'ютерної графіки з наочними ілюстраціями та аналізом складності.

Лабораторні — 45 год. (37,5%)

Програмна реалізація алгоритмів, розробка графічних застосунків, налагодження шейдерів, порівняльне тестування. Значна частка підкреслює практичну, інженерну спрямованість курсу.

Самостійна робота — 60 год. (50%)

Поглиблене вивчення теорії, підготовка рефератів та аналітичних звітів, робота з фаховою літературою. СРС становить **50% обсягу дисципліни**.

Самостійна робота студентів — Детальний опис



Опрацювання літератури

Поглиблене вивчення теоретичного матеріалу за навчальними посібниками з комп'ютерної графіки та алгоритмів (Маценко, Тотосько, Пічугін, Гуржій та ін.), стандартами OpenGL/Vulkan, науковими статтями у фахових виданнях. Студент конспектує ключові алгоритми, аналізує їх складність та порівнює підходи.



Робота з інформаційними ресурсами

Опрацювання офіційних ресурсів: документація Khronos Group ([khronos.org](https://www.khronos.org)), OpenGL Reference (docs.gl), NVIDIA Developer (developer.nvidia.com), Shadertoy (shadertoy.com). Вивчення відкритих репозиторіїв графічних алгоритмів.



Індивідуальні завдання

Виконання аналітичних завдань (порівняння алгоритмів за складністю, аналіз якості рендерингу, оцінка продуктивності GPU); творчих завдань — реферати, аналітичні записки, міні-дослідження з актуальних питань графічних технологій.



Підготовка до занять

Підготовка до лабораторних робіт (опрацювання алгоритмів, написання базового коду); підготовка до тестувань (ключові терміни, формули, властивості алгоритмів); підготовка до захисту лабораторних та модульного контролю.

Форми контролю знань

1

Поточний контроль

Захист лабораторних робіт:
демонстрація працюючої програми,
відповіді на запитання з теорії, аналіз
коду. Оцінюються коректність
реалізації алгоритму, якість коду,
розуміння математичних основ.
Проводиться після кожної лабораторної
роботи; студент повинен захистити
роботу протягом двох тижнів після
виконання.

2

Проміжний контроль

Модульна контрольна робота:
теоретичні питання (алгоритми,
математика, порівняльний аналіз) та
практичне завдання (написання
фрагментів коду або GLSL-шейдера).
Охоплює всі теми відповідного модуля.
Проводиться наприкінці кожного
змістовного модуля у письмовій або
комп'ютерній формі.

3

Підсумковий контроль

Залік відповідно до навчального плану:
захист фінального проекту (3D-сцена з
PBR, тінями, постпроцесингом) та
теоретичне опитування. Підсумкова
оцінка враховує поточний контроль
(лабораторні), проміжний (модульні) та
підсумковий за шкалою: **100-бальна
система** з переведенням у національну
шкалу та ЄКТС (A, B, C, D, E, FX, F).

Методи викладання

Викладання дисципліни «Алгоритмічні основи комп'ютерної графіки» поєднує класичні академічні та інтерактивні методи, що забезпечує формування не лише міцної теоретичної бази, а й практичних навичок програмування, необхідних для майбутньої професійної діяльності у сфері комп'ютерної інженерії та розробки програмних систем.



Лекції

Системне подання алгоритмів та математичного апарату з використанням анімованих схем, псевдокоду, візуалізацій та аналізу складності; студенти ведуть конспекти.



Лабораторні заняття

Програмна реалізація алгоритмів (C++/Python/GLSL), налагодження, профілювання та порівняльне тестування; захист із демонстрацією працюючого застосунку.



Порівняльний аналіз

Аналіз альтернативних алгоритмів за складністю, якістю та ресурсами; студенти обґрунтовують вибір оптимального підходу для конкретної задачі.



Технічні дискусії

Обговорення переваг та недоліків сучасних графічних технологій (растеризація vs. ray tracing, OpenGL vs. Vulkan); розвиток технічного мислення та аргументації.



Самостійне дослідження

Підготовка аналітичних записок і рефератів за сучасними технологіями графіки; розвиток навичок роботи з технічною документацією, стандартами та науковими публікаціями.



Проектна робота

Розробка фінального інтерактивного 3D-проекту: планування, реалізація, налагодження, презентація та захист перед групою із демонстрацією результатів.



Усі методи спрямовані на формування компетентностей, передбачених освітньо-професійною програмою спеціальності 123 «Комп'ютерна інженерія».

Рекомендована література — Підручники та посібники (2022–2024 рр.)

1. Маценко В. Г.

Комп'ютерна графіка: Навчальний посібник. — 2-ге вид., переробл. та допов. — Чернівці : Рута, 2023. — 360 с. *Рекомендується для всіх модулів курсу.*

2. Тотосько О. В., Микитишин А. Г., Стухляк П. Д.

Комп'ютерна графіка: навчальний посібник: у 2-х кн. — 2-ге вид., оновл. — Тернопіль : ТНТУ ім. Івана Пулюя, 2022. — 304 с. *Рекомендується для модулів 1–3.*

3. Пічугін М. Ф., Канкін І. О., Воротніков В. В.

Комп'ютерна графіка: навчальний посібник. — Київ : Видавничий дім «Кондор», 2023. — 320 с. *Рекомендується для модулів 2–4.*

4. Гуржій А. М., Возненко Л. І., Поворознюк Н. І., Самсонов В. В.

Основи інформаційних технологій: навчальний посібник. — Київ : Літера ЛТД, 2023. — 384 с. *Рекомендується для тем 1.1, 3.1, 4.4.*

5. Василюк А. С., Мельникова Н. І.

Комп'ютерна графіка: навчальний посібник для студентів ЗВО. — Київ : Академвидав, 2022. — 296 с. *Рекомендується для модулів 1–2 (растрові алгоритми, перетворення).*

6. Веселовська Г. В., Ходакова В. Є.

Алгоритмічна та математична основа комп'ютерної графіки: навч. посіб. — 2-ге вид. — Київ : Кондор, 2023. — 420 с. *Рекомендується для модулів 1–2 (математичний апарат).*

7. Гончаренко І. М., Єршова О. О.

Шейдерне програмування та графічні API: навч. посіб. — Харків : ХНУРЕ, 2023. — 248 с. *Рекомендується для модуля 4 (OpenGL, GLSL, Vulkan).*

8. Кравченко О. В., Лисенко Т. І.

Рендеринг та алгоритми візуалізації 3D-сцен: навч. посіб. — Київ : Наукова думка, 2024. — 312 с. *Рекомендується для модулів 3–4 (освітлення, рендеринг).*

Додаткова література та наукові статті

9. Стухляк П. Д., Микитишин А. Г.

Алгоритми растрової графіки та обробки зображень: монографія. — Тернопіль : ТНТУ, 2023. — 184 с. *Рекомендується для тем 1.3–1.4 (растрові алгоритми та антиаліасинг).*

11. Морзе Н. В., Барна О. В.

Комп'ютерна графіка та візуалізація в освіті та науці. — Київ : Оріон, 2022. — 196 с. *Рекомендується для розуміння галузі застосування та методик навчання.*

10. Бондаренко О. О., Шестопапов Є. А.

GPU-обчислення та паралельна графіка: навч. посіб. — Харків : Ранок, 2022. — 210 с. *Рекомендується для тем 4.3–4.4 (GPU-архітектура, compute shaders).*

12. Ривкінд Й. Я., Лисенко Т. І., Чернікова Л. А.

Алгоритмічні основи обробки графічних даних: навч. посіб. для ЗВО. — Київ : Генеза, 2024. — 268 с. *Рекомендується для модулів 1–2 (математика та растрові алгоритми).*

Нормативно-правова база

→ Конституція України

Прийнята 28 червня 1996 р. Гарантує свободу наукової та технічної творчості, основа правового регулювання освітньої діяльності.
zakon.rada.gov.ua

→ Закон «Про освіту» від 05.09.2017 № 2145–VIII

Визначає засади освітньої діяльності в Україні. zakon.rada.gov.ua

→ Закон «Про вищу освіту» від 01.07.2014 № 1556–VII

Регулює відносини у сфері вищої освіти, визначає стандарти підготовки бакалаврів зі спеціальності 123. zakon.rada.gov.ua

→ Стандарт вищої освіти за спеціальністю 123 «Комп'ютерна інженерія» (бакалавр), МОН України, 2023 р.

Визначає компетентності та результати навчання для підготовки бакалаврів. mon.gov.ua

→ Положення про організацію освітнього процесу у ЗВО

Наказ МОН України № 161 від 02.06.1993 зі змінами. Регулює форми навчання, контролю та розподіл навчального часу.
zakon.rada.gov.ua

Офіційні ресурси та фахові видання

Khronos Group

khronos.org — офіційні специфікації OpenGL, Vulkan, WebGL, GLSL. Первинне джерело для розробників графічних застосунків.

OpenGL Reference Pages

docs.gl — інтерактивна документація OpenGL за версіями. Зручний пошук функцій та їх параметрів.

NVIDIA Developer

developer.nvidia.com — технічні статті, SDK, профайлери (Nsight), матеріали з ray tracing, DLSS та GPU-обчислень.

Shadertoy

shadertoy.com — платформа для написання та обміну GLSL-шейдерами у браузері. Практичні приклади алгоритмів графіки.

Національна бібліотека України ім. В. І. Вернадського

nbuv.gov.ua — електронні ресурси, наукові статті, монографії з комп'ютерних наук та інформаційних технологій.

Журнал «Проблеми програмування»

pp.isoftware.kiev.ua — провідне фахове видання НАН України з питань програмування, алгоритмів та комп'ютерної графіки.

LearnOpenGL

learnopengl.com — структурований англomовний навчальний ресурс з OpenGL: від базового до PBR та ray tracing.

МОН України

mon.gov.ua — освітні стандарти, навчальні програми, методичні матеріали для спеціальності 123 «Комп'ютерна інженерія».

Зв'язок дисципліни зі спеціальністю 123 «Комп'ютерна інженерія»

Чому «Алгоритмічна графіка» для інженерів?

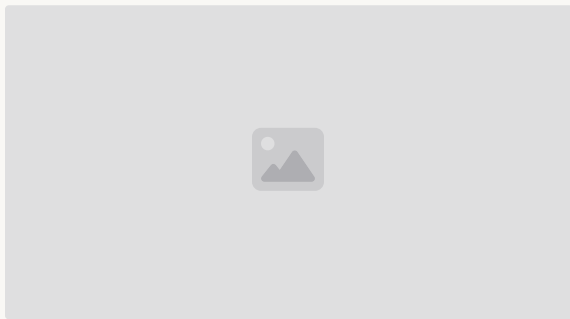
Дисципліна «Алгоритмічні основи комп'ютерної графіки» є профільною для підготовки бакалаврів зі спеціальністю 123 «Комп'ютерна інженерія». Розуміння графічних алгоритмів та GPU-архітектури є обов'язковою компетентністю для розробників систем реального часу, ігрових рушіїв, CAD-систем, медичної візуалізації та промислових симуляторів.

Сучасний фахівець у сфері комп'ютерних технологій дедалі частіше стикається з задачами GPU-обчислень, паралельного програмування та візуалізації даних — усе це безпосередньо ґрунтується на знаннях, що надає ця дисципліна.

Практичне значення для майбутньої кар'єри

- Розробка ігор та інтерактивних симуляторів (Unreal Engine, Unity)
- Створення CAD/CAM-систем та програм для інженерного проектування
- Медична та наукова візуалізація (MPT, 3D-реконструкція, CFD)
- Розробка AR/VR-застосунків та метавсесвітів
- GPU-програмування для задач машинного навчання та нейронних мереж
- Комп'ютерний зір та обробка зображень у промислових системах

Ключові компетентності курсу — Підсумок



Курс охоплює повний стек алгоритмічної комп'ютерної графіки — від математичних основ та растрових алгоритмів до сучасних методів GPU-рендерингу та нейронної графіки — і формує у студентів цілісну інженерну компетентність у галузі графічних систем.

Бажаємо успіхів у вивченні алгоритмічної графіки!

120

Годин

Загальний обсяг дисципліни

4

Кредити ЄКТС

Відповідно до стандарту

16

Тем

У 4 змістовних модулях

8

Джерел

Рекомендованої літератури 2022–2024 рр.

«Комп'ютерна графіка — це мистецтво перетворення математики на красу, що можна побачити.» — Джим Блінн, піонер комп'ютерної графіки

Спеціальність 123 · Комп'ютерна інженерія · Бакалавр · 4 кредити ЄКТС