UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

FAKULTAS ILMU KOMPUTER

CSCE604029 • Computer Graphics
Semester Gasal 2025/2026
Fakultas Ilmu Komputer, Universitas Indonesia

Final Project: Classical Meets AI-based Graphics

Deadline: 02 January 2026 at 23.59 WIB

Relation to Sub-CPL (Course Learning Outcomes)

- Menguasai konsep-konsep dasar grafika komputer secara teoritis seperti model kamera, properti warna dan cahaya, rendering, iluminasi, dan geometri.
- Dapat mendesain solusi grafika komputer untuk pembentukan citra digital menggunakan teknik pemodelan dasar, terutama menggunakan metode berbasis rasterization dan ray tracing.

Project Information

- **Project release date:** 11 December 2025
- **Project type:** Group project (3–4 members)
- **Format:** Implementation + Technical Report + Video Presentation
- **Submission platform:** SCeLE

1 Introduction

This final project combines classical computer graphics theory and algorithms with modern machine learning-based approaches. You are required to implement a computer graphics system that has a foundation in classical rendering and extends it through AI-based graphics algorithms. This project is completed in groups of 3–4 people.

Implementation can take the form of 3D or 2D rendering, graphics-based image processing, 3D generative models, neural rendering, or AI-assisted 3D reconstruction. Each group is expected to combine classical computer graphics components such as rasterization, ray tracing, shading, illumination, geometry, lighting, or animation with AI-based graphics techniques discussed in class such as diffusion-based models, neural radiance fields, gaussian splatting, denoising, generative textures, geometry estimation, or other related techniques.

Each group is also expected to evaluate visual quality and performance according to technical report best practices in the computer graphics field.

2 Project Scope

The final output of this project must involve the following components:

2.1 A. Classical Computer Graphics Techniques

You must implement at least **one** elements from the following list:

- Rasterization pipeline
- Ray tracing
- Volume rendering
- Classical lighting models (e.g., Phong, Blinn-Phong)
- Explicit or implicit geometry representation
- Geometry processing (mesh manipulation, subdivision, decimation)
- Texture mapping
- Classical animation techniques (e.g., keyframing, skeleton-based models)

2.2 B. AI-based Computer Graphics Extension

You must implement at least **one** element from the following list:

- Neural rendering (NeRF, Gaussian Splatting, etc.)
- Diffusion-based models for graphics generation
- Generative texture synthesis
- 3D reconstruction from 2D images via deep learning
- Geometry extraction using learned depth or normal prediction
- Deep Signed Distance Functions
- Learned motion synthesis
- Neural relighting or material estimation

3 Output Requirements

The output of this project must produce a computer graphics system with complexity similar to the following examples:

1. **3D Virtual Environment:** An interactive 3D scene with classical rendering enhanced by neural techniques
2. **Interactive 3D Editor:** A tool for creating or manipulating 3D content using classical and AI methods
3. **Image-to-Geometry Converter:** A system that reconstructs 3D geometry from 2D images

4. **Neural Relighting Engine:** A system that can relight scenes or objects using learned representations
5. **Capture and Animation System:** A pipeline for capturing real-world objects/scenes and animating them

Your system should demonstrate clear integration between classical and AI-based techniques, showing how they complement each other to achieve results superior to using either approach alone.

4 Suggested Project Ideas

Each group is given freedom to develop their project idea. Here are some recommended example ideas:

4.1 A. Re-creating Real Environments in 3D with Animation

Example pipeline:

- Capture photos of a scene to generate geometry using photogrammetry or neural reconstruction
- Apply shading and relighting using classical lighting models
- Animate characters or camera paths within the reconstructed environment

4.2 B. Classically Relight Your Selfies with 3D Gaussians

Example pipeline:

- Perform face segmentation from input images
- Model the face in 3D using 3D Morphable Models (3DMM)
- Apply relighting using classical illumination models
- Render the face using AI-based approaches like 3D Gaussian Splatting

4.3 C. Simple Virtual Try-On System

Example pipeline:

- Capture 3D model of a person or object from images or video
- Generate new textures for the captured model using diffusion models
- Display video containing the same 3D person or object with different costumes or textures

5 Deliverables

5.1 A. Implementation Package

Submit a folder containing:

- **Source code:** Well-documented, organized code with clear structure
- **Assets:** All necessary 3D models, textures, trained models, datasets

- **Demo video:** Video demonstrating your system's capabilities
- **Executables:** Compiled binaries or easy-to-run scripts
- **README:** Installation instructions, dependencies, and usage guide (Optional, add this if program is complex)

5.2 B. Technical Report

Write a technical report following **ACM SIGGRAPH** or **ACM Transactions on Graphics** format (6–8 pages) with the following structure:

Report Format Templates

Use the official ACM templates for your report:

- **ACM SIGGRAPH:** `https://www.siggraph.org/preparing-your-content/author-instructions/`
- **ACM TOG:** `https://dl.acm.org/journal/tog/author-guidelines`

Download the LaTeX or Word template and follow the formatting guidelines carefully.

Required Sections

Your report must include all of the following sections. The specific content within each section should be tailored to your project, but all sections must be present:

1. **Title:** Descriptive title of your project
2. **Authors:** List all group members with NPM
3. **Abstract:** 150–250 word summary of the project
4. **Introduction:** Introduce your project, motivate the problem, and describe your approach
5. **Literature Review:** Survey relevant prior work in classical and AI-based graphics
6. **System Design:** Describe your system architecture and technical approach
7. **Implementation:** Detail your implementation of both classical and AI components
8. **Evaluation:** Present and analyze your experimental results
9. **Discussion:** Discuss findings, limitations, and future work
10. **References:** Properly cited academic papers and resources

5.3 C. Video Presentation

Record a 10–15 minute video presentation explaining the methods used in your project. The presentation should be clear, well-paced, and well-produced. All group members should participate in the presentation.

Suggested content structure:

- Overview of the problem and your approach
- Explanation of classical graphics methods employed

- Explanation of AI-based techniques used
- System integration and pipeline
- Demonstration of results
- Key findings and conclusions

6 Assessment Criteria

Your project will be evaluated based on the following criteria:

Component	**Weight**
Classical method implementation	30%
AI-based implementation	30%
Quality of output (visual results, performance)	20%
Technical report (clarity, completeness, rigor)	20%

7 Important Guidelines

7.1 Use of External Resources

You are **permitted** to use external trained models or existing pipelines, provided that:

- You add original components or modifications
- You clearly document what is borrowed vs. what is your own contribution
- The integration itself demonstrates technical understanding
- Your report explicitly states what was used from external sources

Examples of acceptable use:

- Using a pre-trained NeRF model but adding custom lighting controls
- Using Stable Diffusion for texture generation within your custom pipeline
- Adapting an existing 3D reconstruction method to a new application

Unacceptable use:

- Running an existing system without modifications and claiming it as your work
- Minimal wrapper around existing code without substantial additions

7.2 Technical Recommendations

- **Start early:** Complex systems require significant development time
- **Iterative development:** Build a simple baseline first, then add features
- **Version control:** Use Git for collaboration and backup
- **Regular testing:** Test components individually before integration
- **Documentation:** Document as you code, not at the end
- **Checkpoints:** Save intermediate results and models

8 Submission Requirements

8.1 File Organization

Submit a single ZIP file named: `<GROUP_NAME>-FinalProject.zip`

8.2 Submission Platform

- **Platform:** SCeLE
- **Alternative:** If file size exceeds limit, submit a link to Google Drive or similar cloud storage (ensure permissions are set correctly)

9 Academic Integrity

- All code must be properly attributed if borrowed
- Plagiarism will result in severe penalties
- Collaboration between groups is not allowed
- External help must be acknowledged in the report

Good luck with your final project!
May your renders be beautiful and your gradients converge!