Anmol Goyal

anmolafk7@gmail.com | anmol-goyal7 | anmol-goyal | anmol-goyal7.github.io

SUMMARY

CS freshman focused on GPU compute, computer graphics, and systems programming. Interested in understanding how computers work at the lowest level — from writing compilers and rasterizers to parallelizing workloads with OpenMP. Comfortable working in C/C++ and building projects from scratch without frameworks. Currently exploring real-time rendering, parallel computing, and low-level code generation.

EDUCATION

SRM Institute of Science and Technology 2025 – 2029
Bachelor of Technology in Computer Science and Engineering Kattankulathur, Chennai

PROJECTS

cuda-imgproc | *C++17, CUDA, OpenMP* 2026

- Header-only image processing library with **three backends**: CPU (serial), OpenMP (parallel), and CUDA (GPU)
- Implements color space conversion, data augmentation, 2D convolution, and histogram equalization
- Built benchmarking infrastructure to compare throughput across all three backends on identical workloads

Ray Tracer | *C++* 2026

- Path tracer built from scratch following Peter Shirley's *Ray Tracing in One Weekend* series
- Implemented ray-sphere intersection, surface normals, antialiasing, and diffuse/metal/dielectric materials
- Automated render pipeline with Makefile and GitHub Actions for continuous output generation

Mini C Compiler | *C++, x86-64 Assembly* 2026

- Compiler for a subset of C targeting x86-64 assembly, built from scratch to understand compiler internals
- Full pipeline: lexer (tokenizer) → parser → abstract syntax tree → code generator → executable
- Supports integer literals, return statements, and arithmetic expressions with operator precedence

Matrix OpenMP Benchmark | *C, OpenMP* 2025

- Benchmarked serial vs. OpenMP-parallelized matrix multiplication; achieved **4.72× speedup**
- Measured scaling behavior across thread counts and matrix dimensions
- Analyzed cache effects and thread overhead to identify optimal parallelization parameters

CPU Software Rasterizer | *C++* 2026

- Software rasterizer built from scratch — line drawing, triangle rasterization, z-buffering, texture mapping
- No graphics API dependencies; renders 3D models to image using only CPU and custom math
- Parses OBJ model files and applies perspective projection, Gouraud shading, and backface culling

SKILLS

Languages: C, C++
Tools: Neovim, Git, Make, OpenMP
Interests: GPU Compute, Computer Graphics, Systems Programming, Parallel Computing