

Ivan Neulander

Los Angeles, CA
(310) 654-0253
ineula@gmail.com

www.neulander.org/work
<https://www.linkedin.com/in/neulander>

Google

Google Play Games Graphics TLM (Staff Software Engineer) 2022-Present

Play Games is Google's gaming platform designed to run Android games on Windows through high-performance virtualization (in beta).

- Setting direction for performance and stability improvements on a virtualized graphics stack.
- Managing and growing a team of 8 Software Engineers.
- Authored 1 patent.

AR Beauty TL, TLM (Staff Software Engineer) 2019-2022

AR Beauty is Google's interactive beauty shopping technology that uses Augmented Reality to help users visualize thousands of products through virtual try-ons and pre-rendered model images.

- Led team of 8 Software Engineers to launch thousands of Lipstick, Eyeshadow, and Foundation virtual try-ons on Google Search and YouTube.
- Scoped, designed, built, released Google's Beauty SDK to partners, unlocking third-party authorship of AR virtual try-ons for cosmetics on iOS, Android, and web.
- Established partnerships with major beauty brands to bring their products to Google.
- Managed team of 3 Software Engineers (2021 onward).
- Authored 2 Patents.

Mobile AR Graphics TL (Senior Software Engineer) 2016-2019

Led a team of 3-5 Software Engineers to build an AR renderer on Android Pixel phones for the built-in camera app.

- Launched AR Stickers/Playground, featuring PBR shading, real-time AR lighting estimation, simulated subsurface scattering, soft shadows, all running at 30 FPS on first-gen Pixel and flagship Android phones.
- Presented our work at FMX and SIGGRAPH LA.
- Authored 5 Patents.

Graphics Engineer and Researcher (Senior Software Engineer) 2013-2016

Various Projects involving Image Processing, Computer Vision, Deep Learning:

- Painerly Rendering: Real-time, coherent painterly animations from photographs.
- Deep Stereo: Deep Learning synthesis of novel views from a sparse set of images. My contribution was incorporating LIDAR data into the machine learning model.
- LIDAR visualization: Tools to help visualize colored point clouds from massive Street View LIDAR collections.
- Authored 4 patents.

Rhythm & Hues

Rendering Team Lead (Principal Software Engineer) 2002-2013

Led a team of 4 Software Engineers in charge of Rhythm's proprietary rendering software. Notable projects, essential to the Oscar-winners *The Golden Compass* and *Life of Pi*, include:

- Importance sampling strategies for global illumination, including support for area lights, multiple importance sampling, and adaptive importance sampling.

- Methods for efficiently rendering photorealistic hair and fur, particularly with ray-traced radiance sampling and interactive image-based lighting.
- Noise reduction techniques for irradiance-cached dipole-based (Jensen-Buhler) multiple scattering approximation.
- A Multithreaded $2\frac{1}{2}$ D motion blur processor, which became the preferred method for motion-blurring heavy production renders; subsequently adapted to synthesize alternate camera views for stereo productions.

Rendering Software Engineer (Software Engineer) 1998-2002

Authored core functionality in company's renderer, including:

- Scanline hair rendering, based on the work from my Master's thesis.
- Displacement mapping, using dynamic, view-adaptive tessellation.
- Specialized texture-space rasterizer for rapidly generating high-resolution textures with geometry-occluded irradiance or displaced shading normals.
- A scripted toolset for multiplatform building, testing, and deployment of Rhythm's proprietary applications and code libraries, with cross-referenced release notes.

Education **Master of Science**, University of Toronto 1995-1997

- Master's Thesis: *Rendering Generalized Cylinders using the A-Buffer*

Bachelor of Science (Honours), University of Toronto 1991-1995

- Specialist Degree in Computer Science
- Major in Mathematics, Minor in Philosophy
- Graduated with High Distinction, GPA 4.09

Technical Skills Languages: C++, C, Python
Technologies: 3D Graphics, Linux