

Ángel Ortiz

AEROSPACE ENGINEER · GRAPHICS PROGRAMMER

Avenida Icaria 137 3.^{er}.^a, Barcelona, Spain

☎ +34(34)651-856-004 | ✉ angelo12@vt.edu | 🏠 www.aortiz.me | 📱 Angelo1211 | 📧 angelortizelguero

Education

Virginia Polytechnic Institute and State University

B.S. IN AEROSPACE ENGINEERING

Graduated May 2017

Blacksburg, Virginia, U.S.A.

Experience

HRE :: Hybrid (Deferred/Forward) Rendering Engine

Barcelona, Spain

GRAPHICS PROGRAMMING - SOLO PROJECT

Sept. 2018 - Present

- Built a GPU based Clustered tile renderer using OpenGL that enables both deferred and forward based algorithms and includes Image Based Lighting and Physically Based Rendering. The Source code can be found on my GitHub page.
- Performed an overview of modern shading methods by implementing deferred, forward, tiled forward and clustered forward shading in an effort to evaluate and gain familiarity with the taxonomy of modern real-time rendering pipelines.
- Settled on clustered shading and implemented compute shader based light culling for efficient rendering of scenes with many lights.
- Implemented many other staple rendering features such as: Shadow Mapping, MSAA, Linear/HDR Lighting, Ambient Occlusion mapping and exposure based tone-mapping, among others.
- Researched and implemented PBR and IBL and settled on an approach based on Unreal Engine 4 as described in the Siggraph 2013 paper "Real Shading in Unreal Engine 4".

SSGE :: Simple Software Graphics Engine

Barcelona, Spain

GRAPHICS PROGRAMMING - SOLO PROJECT

June - Aug. 2018

- Built a real-time interactive CPU-based 3D Renderer from scratch in C++ capable of modern physically based shading without relying on any external maths or graphics libraries. Source code can be found on my GitHub page.
- Wrote utility libraries for the engine such as: templated vector class, matrix and linear algebra classes and .obj and scene data parsers.
- Used CMake and SDL2 as abstraction layers to implement multiplatform functionality for Windows and Linux.
- Optimized for real-time performance by leveraging OpenMP multithreading, Intel C++ compiler autovectorization and profilers such as Intel's VTune Amplifier, Gprof and KCacheGrind.
- Implemented a programmable shader pipeline using virtual functions and wrote a PBR shader with a cook-torrance specular BRDF and Lambertian Diffuse.

Statoil/C-CORE Iceberg Classifier Challenge

Barcelona, Spain

MACHINE LEARNING - SOLO PROJECT

Oct. - Nov. 2017

- Joined a Kaggle competition with the aim to create an algorithm capable of differentiating icebergs from ships using only radar HH and HV data obtained by a satellite from a 600km altitude orbit. The data set contained over 1600 70x70 pixel images.
- Pre-processed the radar data into a RGB image composite to speed up training and allow for deeper convolutional layers.
- Developed Python scripts to explore different Machine Learning models such as: Random Forest Classifier, Artificial Neural Networks and Convolutional Neural Networks using Sklearn and Keras. Performed their comparison using k-fold cross validation numbers.
- Settled on using a CNN as the main algorithm and performed a grid search on model hyper-parameters to fine tune the performance and reduce error to only 9%.
- Reformatted final code into a Jupyter notebook for better presentation and included with it a summary of the development process. It can be found on my GitHub page.

RASC-AL Commercially Enabled LEO / Mars Habitable Module

Blacksburg, VA

SENIOR DESIGN PROJECT - MARS MODIFICATIONS & FINANCIAL PROJECTIONS LEAD

Aug. 2016 - June 2017

- Won first place overall, first place in undergraduate division and best in theme at the 2017 RASC-AL Design competition final, a university-level, NASA sponsored, full mission architecture engineering design competition.
- Led the Mars modification design with a focus on human spacecraft requirements addressed in the NASA Technical Standards for Human Integration.
- Created a profit projection tool in MATLAB that performs a 15 year revenue projection based on station operational costs and income from experiments to prove the commercial viability of the station.
- Worked closely with a team of 12 students to design a LaTeX report template and led the editing and development of the final report.

Technical Skills

- Graphics Programming with OpenGL
- Compute & GPGPU Programming
- Physically based rendering & IBL
- Clustered & tiled forward shading
- Debugging via Renderdoc, Visual Studio
- Optimization with VTune and SIMD
- Multithreading OpenMP, C++11 threads
- Multiplatform Development
- C, C#, C++
- GLSL
- Mathematica & MATLAB
- Python 3.5