Tim C. Schröder

Email: tim@blitzcode.net | See www.blitzcode.net for further details

Experienced Software Engineer
Systems Programming · HPC · Functional & OOP · Computer Graphics

Solving challenging parallel programming problems, large datasets, realtime constraints, researching and implementing algorithms on everything from a game console to a supercomputer. Writing maintainable and efficient code for shipping software products. Diverse computer graphics background, including games and physically based rendering. Genuinely enjoy learning new things and teaching others.

Skills

Programming Languages

- Strong C/C++ (many different platforms, compilers and environments, C++1x, STL, Boost, parallel extensions like CUDA and OpenMP, projects with >1M LoC, 15+ years)
- Strong Haskell (Cabal/Stack, FFI, Lens, Repa, Async, STM, Conduit, MTL/Transformers, (Atto)parsec, Criterion, Threepenny, Http-Client, Vector, Aeson, EKG, Text, …)
- Rust, C++/CLI, Lua, C#, HTML & CSS, Vim Script, Markdown, XML, JSON, 6502, TIS-100
- GPU shading and compute languages (CUDA, Asm, Cg, CgFX, GLSL, HLSL and FX Files)

Algorithms / Techniques / Data Structures

- Hash tables, HAMTs, functional programming, monads, Monte Carlo tree search / UCT, (Quasi-) Monte Carlo integration, linear algebra, out-of-core processing, lock-free programming, data compression (Burrows-Wheeler transform, Huffman codes), BVH & kD trees, stencil codes, CPU emulation, SIMD code, smart pointers, intrusive containers, dynamic programming, …
- Everything from realtime 3D for games to physically based rendering, techniques such as path tracing, photon mapping, surface parameterization, spherical harmonics, progressive meshes, shadow maps / volumes, spectral rendering, tone mapping, environment map lighting, image compression (DCT based), BRDFs, precomputed radiance transfer, ambient occlusion, distance fields, ray marching, software rasterization, hierarchical sample warping, fractals, …
- Graphics experience on multi CPU / GPU systems, game consoles, CellBE and GPGPU platforms

Linux / UNIX

- Development experience on Ubuntu, Raspbian, BSD, Cray Linux Environment, SGI IRIX/Altix etc.
- Working with standard tools, Makefiles, Vim power user, highly networked environments (ssh, X11, GNU screen, rsync, curl, etc.), pthreads, ptrace, sockets, other POSIX APIs, Cygwin, …
- Clang, gcc and GNU binutils, ELF file format, build optimizations for CellBE SPUs
- Debugging of complex multithreaded applications with TotalView debugger, knowledge of gdb/lldb and its various front ends, also familiar with SGI ProDev WorkShop Debugger (cvd)
- Valgrind (incl. cachegrind / callgrind), Intel VTune Performance Analyzer for Linux, gprof

Macintosh

- OS X power user with development experience, Mach kernel APIs, Homebrew, Apple Shark profiler, Instruments, BSD layer tools like atos, dsymutil, fs_usage etc.
- Developed open source realtime native code profiling tool (rsvp)

Windows

- Deep knowledge of tools and platform (10+ years), Visual Studio power user
- Win32/64 Expert (processes & threads, memory management, DLLs & binary compatible interfaces, networking, .NET environment, DirectX, porting to / from UNIX, debugging etc.)
- User interface development with Windows Forms, MFC and the raw Windows APIs

Software / APIs

- VMWare, VirtualBox, Doxygen, VTune, Quantify, Git, SVN/CVS, Perforce, FogBugz, Bugzilla, Visual Assist, IncrediBuild, BoundsChecker, NVIDIA Visual Profiler, nvidia-smi, FreeType 2, jQuery, …
- 10+ years of experience with Direct3D and OpenGL (including modern 3.x/4.x style), GLFW
- Game console development with the PLAYSTATION 3 SDK & GameOS tools (including PA Suite, ProDG debugger), classic XBox SDK, basic knowledge of the 360 SDK
- mental ray GPU shaders, Phenomena and .mi scene description language
- RESTful APIs using HTTP & JSON (like Twitter and Philips Hue)

Professional Experience

Well-Typed (June 2016 – July 2016, Haskell Consultant, Contract)

- Development of a Haskell wrapper for the Hadoop streaming interface

Barclays Capital (January 2015 – May 2015, Front Office Quantitative Developer, Contract)

- Developer on Monte Carlo GPU code and Haskell payout DSL
- Trained colleagues and assisted with the setup of git repository / workflow for the team
- Provided training and advice for CUDA / GPU development, profiling and architecture

NVIDIA (November 2010 – August 2013, Senior Compute DevTech Engineer)

- Opening up new domains of application for GPUs by researching and developing GPU computing algorithms, driving their adoption with key application developers, and ensuring the best possible performance of GPU computing applications on current and next-generation architectures
- Supported HPC experts and domain scientists during GPU port of a major Numerical Weather Prediction package. Software included stencil metaprogramming library in CUDA C++ and PGI Accelerator instrumented F90 code. Deployment on Tesla GPU machines from Cray and HP
- Customer training through webinars, conference presentations, on-site visits and seminars
- Gave feedback based on customer codes to internal driver, architecture, compiler and tool teams
- Developed CUDA SDK samples (Peer-to-Peer, Unified Virtual Addressing etc.), documentation

NaturalMotion (January 2010 – April 2010, Senior Runtime Engineer)

- Engineer on the PC and console (PS3 / 360) runtimes for morpheme / euphoria game animation middleware (optimization, refactoring, development of new animation nodes, C++ / Lua, ...)
- Direct3D based renderer (characters, shadows, outdoor lighting) for demonstration application

mental images (May 2004 – December 2009, Senior Graphics Software Engineer)

- Autodesk ProMaterial shader support for iray renderer
- Optimization work on mental ray's BSP2 renderer (memory, intersection)
- Early development of a relighting tool, including algorithm / data structure design, implementation of the GUI and file formats (project put on hold for business reasons)
- Researched raytracing algorithms for CellBE processor using simulator / IBM Blade, successful prototype led to contract for a CellBE port of mental ray
- Worked with a large range of CellBE platforms: IBM QS20/22 Blades, Sony BCU-100 and PLAYSTATION 3 (GameOS & Linux), implemented advanced concepts such as cooperative multithreading for SPEs, a pipeline for loading and executing MetaSL SPE shader code, a C++ template library for DMA transfers, trained and supported team members in all things CellBE
- Developed and documented strategy to port raytracing algorithm from proprietary chip to NVIDIA CUDA platform, including implementation of a CPU based prototype
- Maintained and enhanced mental ray's OpenGL / Cg rendering backend while doing a Direct3D / HLSL / FX File port of it
- Developed scalable rendering software on multi-GPU machines from SGI and PANTA Systems
- Worked on next-generation rendering platform for RealityServer 2.0: GLSL / Cg, tile rendering, OpenGL, texturing, DDS support, shadowmaps, anti-aliasing, cubemaps, ...
- Assisted in MetaSL / mental mill development (Cg backend, Direct3D rendering, early evaluation of an VM platform, builds for external employees, CellBE port)
- Early design and implementation of 2D annotation renderer for RealityServer 2.0
- Supported hardware shader writers for 3ds Max and SolidWorks
- Created test suite to ensure quality of mental ray hardware rendering
- Developed in a UNIX-centric environment using Linux / IRIX / Cygwin

Freelancer (March 2004 - May 2004)

- Implemented photon mapping for walkthrough applications as C++ library for KPB
- Created a lightmap compiler and viewer using C++ and OpenGL for S&P Software
- Presented 'Realtime Shadow Techniques' at JOIN04 conference

Crytek (March 2001 - August 2003, Software Engineer)

- Developer on tools, game and engine code for Far Cry / CryENGINE
- Worked on WYSIWYG game world editor (MFC application, procedural tex. generation & object placement, heightfield editing & lighting, game engine integration, ...)
- Did basic rendering R&D and a proof of concept port of CryENGINE on the Xbox
- Wrote game logic code in C++ / Lua (had to work in multiplayer scenarios)
- Implemented Lua script code debugger integrated in game & editor (including GUI)
- Programmed lightmap system that combines classic lightmaps with bumpmapping

Portfolio

Please visit my website www.blitzcode.net and my GitHub account github.com/blitzcode.