Andrew**Selle**

aselle@andyselle.com
http://andyselle.com

Objective | Engineering position that combines machine learning, computer graphics, simulation, and numerics.

EXPERIENCE

Senior Staff Software Engineer, Google 2018-Present
Created next generation of machine learning infrastructure.

- **Designed and prototyped next-generation machine learning tooling.** Analyzed trends and user feedback to conclude that research-to-production friction was the chief user problem. Advocated, gained support, and prototyped a more general runtime and authoring system for machine learning models. Included data structures, abstraction, encapsulation, and interoperability. Empowered tighter integration of JIT and AOT compilers for transformations, optimizations, and code generation to ensure generality and performance on arbitrary machine learning programs.
- **Unified TensorFlow Mobile and Server Runtimes.** As lead of Mobile TFRT, designed and prototyped path for TensorFlow Lite utilizing asynchronous features in the new TensorFlow Runtime. Led effort to create a kernel compatibility interface to allow TensorFlow Lite and the new TensorFlow Runtime to be fully compatible with existing TensorFlow software.
- **Tech lead of TensorFlow Lite Core**. Led collaboration MLIR team that created a new TensorFlow Lite compiler (first production use of MLIR). Implemented a microcontroller version of TensorFlow Lite proving the viability on smaller devices. Led TF Lite metadata design that enabled easy-to-use Android Studio integration. Created improved delegate interface to support calling TF operations, using mobile TPU accelerators, GPUs, and ahead of time delegation. Led specification of sparse data structures, improved functions, and memory management. Collaborated cross-functionally with many production users and groups both internally and externally.
- **Built AI-based dance technology.** Tech lead of a team that built a new AI-based interactive dance technology which utilized on-device inference of pose-detection and dynamic time warping. Demonstrated a subset of the technology as a demo application at Google IO 2019 and features in the developer keynote. Generalized technology to build a mobile interactive video system.

Staff Software Engineer, Google Brain 2016-2018
Architected, developed, and productionized machine learning software for server and edge devices.

- **Architected, Built, and Productionized TensorFlow Lite.** Specified, architected, developed, and productionized a runtime and API for mobile/edge machine learning. It is used on billions of devices including phones, smart speakers, cars, smart sensors, and more. Created the TensorFlow Lite C++ API, Python API, interpreter, kernel interface, and delegate interface. Designed and implemented a flatbuffer-based serialization format that provides strong version compatibility. Supported cross-functional partner teams at Google including Android (Neural Network API) including Speech, OCR, Nest Home, YouTube, Photos, GBoard, Coral. Convinced developers of existing ML frameworks to migrate to TensorFlow Lite.
- **Drove Open Source TensorFlow Engagement.** Presented new technology and represented TensorFlow at external conferences. Assembled team of engineers to triage and engage on Github issues. Designed processes, dashboards, and tooling to ensure prompt response to customer needs. Built a Chrome Extension to improve triage efficiency.
- **Researched automatic neural architecture search.** Developed evolutionary model for neural architecture search on image models using Lamarckian evolution. Published positive CIFAR10/CIFAR100 results. Used research difficulties to propose and drive improvements to TensorFlow's model mutation capabilities.
- **Made TensorFlow API Friendly.** Served as an API leader by owning and reviewing API changes. Conformed the TensorFlow API toward NumPy and Python standards. Removed inconsistencies in naming and created organized namespaces. Adhered to proper division behavior, NaN handling, and implemented NumPy-style slicing, created namespaces to organize the API. Identified the need, socialized, implemented and productionized a tool that automatically upgrades users' TensorFlow programs.

Principal Software Engineer, Walt Disney Animation Studios 2013-2016
Devised, planned, execute and lead research and software projects for production needs.

- **Architected and led development of production water system**. Consisted of a new artist friendly deep water wave system, boat buoyancy simulator for layout, a novel fluid simulator based on our dissipation reducing APIC research, as well as implicit surface compositing and rendering system. Partially used for *Zootopia* and fully being deployed on *Moana*.
- **Architecting and designing experimental content-addressable next-generation data pipeline.** This approach seeks to allow data-pipelines to be distributed and replicated for better efficiency in mapping to large clusters. This contrasts with traditional film pipelines that rely on NFS global file systems.
- **Architected and implemented major subsystems on Disney's in-house renderer Hyperion.** Developed instancing, motion blur, depth of field, volumes, geometry pipeline, fine-grain statistical profiling, web-based statistics, and physically-based lenses. First deployed on Academy Award Winning *Big Hero 6* and *Feast*. Continues to be the exclusive renderer at Walt Disney Animation Studios.
- **Technology leader at Disney Animation.** Evangelized improving skill set of myself and others. Advised the CTO on new technologies as part of the tech trust group. Member of patent panel. Led study groups and taught classes in C++11, CUDA, git, gitlab, etc.

Senior Software Engineer, Walt Disney Animation Studios 2008-2013
Devise, plan, execute and lead research and software projects for production needs.

- **Led research, development and deployment of snow simulation system for *Frozen***. Developed a material point method (MPM) snow simulator with an elasto-plastic constitutive model that allowed better realism than any previous method. Published in SIGGRAPH. Led to Disney CEO Innovator Award.
- **Built water simulation tools**. Consisted of PhysBAM simulation, a custom shallow water simulator and curl-noise based waterfall simulation for *Tangled*. Developed a flexible node-based field-authoring tool for artistic control which also became used in cloth and hair simulators. Built as Maya and Houdini plugins.
- **Built rigid body simulation and fracture tools.** Voronoi fracture, cutting, PTex texture remapping, bullet-based rigid body simulation, and Houdini integration. First version used in *Bolt* and updated and improved for *Wreck-It-Ralph*.
- **Designed, developed and led project to create flexible particle authoring and instancing tool.** Achieved scaling to hundreds of millions of particles where previous system only achieve tens-of-thousands. Improved, integrated and open sourced the SeExpr programming language for flexibility including creating a more efficient LLVM JIT backend. Designed and developed the open source Partio particle library for file IO and processing. Previewing using OpenGL (GLSL, VBOs, display lists). Used in *Tangled, Wreck-It-Ralph, Frozen, Big Hero 6*, and all future films. Became a flexible platform for a multitude of other effects in films including volumetric beams (*Tangled*), aurora borealis (*Frozen*), crowds (*Tangled, Big Hero 6, Zootopia*), and massive city layout and rendering (*Big Hero 6, Zootopia*).

Research and Development Engineer, Industrial Light + Magic 2006-2008
Provided PhysBAM and physical simulation expertise, fulfilled production needs.

- Technical directed Physical simulation-oriented shots.
- Consulted with full-time developers on various aspects of physical simulation needs.
- Responsible for software transfer between Stanford and ILM. Maintained PhysBAM software branch at ILM. Updated ILM dynamics software and Zeno interfaces when PhysBAM was re-architected.
- Developed distributed memory parallel (MPI) version of cloth and dynamics system.

Research Assistant, Stanford University 2003-2008
Researched, developed and published new simulation technology.

- Major developer on the PhysBAM physics based simulation library. Participated in many architectural refactors, developed many simulators.
- Extended the PhysBAM renderer to be physically based, improved volumetrics and added photon mapping.
- Developed expertise in level set implicit surfaces including rendering and meshing.
- Teaching assistant for numerical analysis classes.
- Researched new simulation technologies and published several SIGGRAPH papers.
- System administrated lab computer systems. LDAP, Wiki, Linux, VLANs, NFS, Samba, Torque PBS, Sun Grid Engine, etc. Wrote a TCP/IP based instant messaging system capable of sharing multimedia.

Research Consultant, Microprocessor Technology Labs, Intel Corporation 2004-2007

- Parallelized simulation kernels and entire simulators in PhysBAM using task stealing work-queues using shared-memory parallel systems.
- Provided physics simulation kernels to act as test-workloads for behavioral architectural simulations of potential many-core (100+) processors. Later others in the group targeted similar kernels to Larrabee.

EDUCATION

Ph.D., M.S. Computer Science
Stanford University (2008)
Advisor: Ronald Fedkiw Thesis: "Hybrid Techniques for High-fidelity Physical Simulation of Solids and Fluids"

B.S. Computer Science and Mathematics with Honors
University of Wisconsin Madison (2003)

OPEN SOURCE

- **TensorFlow**: A library for machine learning.
- **Partio**: A library for loading and manipulating particle systems.
- **SeExpr**: A simple and easy to use expression language with an interactive graphical user interface.

PROFESSIONAL ACTIVITIES

- **Academy of Motion Picture Arts & Sciences**: Digital Imaging Technology Subcommittee (DITS) of the Scientific and Technical Awards Committee : 2015
- **SIGGRAPH Technical Papers Committee**: 2016, 2015, 2013
- **SIGGRAPH Asia Technical Papers Committee**: 2013, 2011
- **Symposium on Computer Animation Papers Committee**: 2015, 2013, 2010, 2009
- **Professional Organizations**: Visual Effects Society, Association for Computing Machinery.

FILM CREDITS

Moana (2016); *Zootopia* (2016); *Big* Hero 6 (2014); *Frozen* (2013); *Wreck-It-Ralph* (2012); *Winnie the Pooh* (2011); *Tangled* (2010); *Princess and the Frog* (2009); *Pirates of the Caribbean 3* (2007); *Evan Almighty* (2007); *Poseidon* (2006).

PATENTS

- Physical simulation: Material point method for simulation of granular materials (20150187116); Efficient elasticity for character skinning (9135738); Augmented material point method for simulating phase changes and varied materials (20150186565); Computer graphic system and method for simulating hair (8803887);
- Rendering: Integration cone tracing (9123162); Distributed element rendering (9123154); Ray cone hierarchy renderer (9058690); Streaming light propagation (9053582); Streaming hierarchy traversal renderer (8957896); Ray-mediated illumination control (20140327675); Normalized diffusion profile for subsurface scattering rendering (20140267274); Residual Ratio Tracking for Estimating Attenuation in Heterogeneous Volumes (US20160061729A1) Ordering rays in rendered graphics for coherent shading (20140253576);
- UX: Collaboration plane (8645845)

PAPERS

Large-Scale Evolution of Image Classifiers. Real E., Moore S., Selle A., Saxena S., Suematsu Y.L., Tan J., Le Q., Kurakin A. ICML 2017 (34th International Conference on Machine Learning).

Fluxed Animated Boundary Method, Stomakhin, A., Selle, A. ACM Transactions on Graphics (SIGGRAPH), 2017.

The Affine Particle-In-Cell Method, Jiang, C. Schroeder, C., Selle, A., Teran, J., Stomakhin, A., ACM Transactions on Graphics (SIGGRAPH), 2015.

Residual Ratio Tracking for Estimating Attenuation in Participating Media. Novak, J., Selle, A., Jarosz, W. ACM Transactions on Graphics (SIGGRAPH ASIA Proceedings), 2014.

Augmented MPM for phase-change and varied materials, Stomakhin, A., Schroeder, C., Jiang, C., Chai, L., Selle, A., ACM Transactions on Graphics (SIGGRAPH), 2014.

A material point method for snow simulation, Stomakhin, A., Schroeder, C., Chai, L., Teran, J., Selle, A., ACM Transactions on Graphics (SIGGRAPH), 2013.

Sorted Deferred Shading for Production Path Tracing, Eisenacher, C., Nichols, G., Selle, A., Burley, B., Eurographics Symposium on Rendering 2013, Best Paper.

A Programmable System for Artistic Volumetric Lighting, Nowrouzezahrai, D., Johnson, J., Selle, A., Lacewell, D., Kaschalk, K., Jarosz, W., ACM Transactions on Graphics (SIGGRAPH), 2011.

Efficient elasticity for character skinning with contact and collisions , McAdams, A., Zhu, Y., Selle, A., Empey, M., Tamstorf, R., Teran, J., Sifakis, E., ACM Transactions on Graphics (SIGGRAPH), 2011.

Gaussian Quadrature for Photon Beams in "Tangled", Johnson, J., Jarosz., W., Lacewell, D., Selle, A„ SIGGRAPH Talks 2011.

Tangled Choreographing Destruction: Art Directing a Dam Break, Kaschalk, M., Boggs, B., Selle, A., Chai, L., SIGGRAPH Talks 2011.

Art-directing Disney's Tangled Procedural Trees, Shek, A., Lacewell, D., Selle, A., Teece, D., Thompson, T., ACM SIGGRAPH Talks 2010.

Geometric Fracture Modeling in BOLT, Hellrung, J., Selle, A., Shek, A., Sifakis, E., Teran, J., ACM SIGGRAPH Talks 2009.

Synthetic Turbulence using Artificial Boundary Layers, Pfaff, T., Thürey, N., Selle, A., and Gross, M., ACM SIGGRAPH Asia 2009 Papers; ACM Press, 2009.

Detail Preserving Continuum Simulation of Straight Hair, McAdams, A., Selle, A., Ward, K., Sifakis, E., Teran J., ACM Transactions on Graphics SIGGRAPH 2009, ACM TOG 28, 3 (2009).

Hybrid Techniques for High-fidelity Physical Simulation of Solids and Fluids, Selle, A., Stanford University Doctoral Dissertation (2008).

A Mass Spring Model for Hair Simulation, Selle, A., Lentine, M., G., Fedkiw, R., ACM Transactions on Graphics SIGGRAPH 2008, ACM TOG 27, 64.1-64.11 (2008).

Robust High-Resolution Cloth Using Parallelism, History-Based Collisions and Accurate Friction, Selle, A., Su, J., Irving, G., Fedkiw, R., IEEE Transactions on Visualization and Graphics (TVCG) 15(2) 339-350.

An Unconditionally Stable MacCormack Method, Selle, A., Fedkiw, R., Kim, B.-M, Liu, Y., Rossignac, J., J. Sci. Comput. 35, 350-371 (2008).

Physical Simulation for Animation and Visual Effects: Parallelization and Characterization for Chip Multiprocessors, Hughes, C.J., Grzeszczuk, R., Sifakis, E., Kim, D., Kumar, S., Selle, A., Chhugani, J., Holliman, M., Chen, Y.K., In the Proceedings of IEEE/ACM International Symposium on Computer Architecture (ISCA), San Diego, California, June 2007.

Simulating Speech with a Physics-Based Facial Muscle Model, Sifakis, E., Selle, A., Robinson-Mosher, Fedkiw, R., Symposium on Computer Animation, ACM/Eurographics.

Multiple Interacting Liquids, Losasso, F., Shinar, T., Selle, A., Fedkiw, R., SIGGRAPH 2006, ACM TOG 25.

A Vortex Particle Method for Smoke, Water and Explosions, Selle, A., Rasmussen, N., Fedkiw, R., SIGGRAPH 2005, ACM TOG 24, pg 910-914.

Coupling Water and Smoke to Thin Deformable and Rigid Shells, Guendelman, E., Selle, A., Losasso, F. and Fedkiw, R., SIGGRAPH 2005, ACM TOG 24.

Cartoon Rendering of Smoke Animations, Selle, A., Mohr, A., Chenney, S., Proceedings of NPAR 2004 (Non-Photorealistic Animation and Rendering), 2004, pp 57-60.

PERSONAL PROJECTS

Electronics software/hardware: Acoustic waveguide simulator using boundary element method, Class D audio amplifiers, Audio DSP, USB HID microcontrollers, FPGA custom computer, Verilog, 4k color VGA display using discrete R-2R DAC. Designed custom PCBs. From scratch HTML5/JavaScript SPICE-like circuit simulator (Newton-Raphson on nodal equations). http://blog.andyselle.com

Manufacturing technology: CNC machining (took classes in conventional machining, CNC machining, CAD/CAM). Customized 3D printers. Built DIY CNC router.

SKILLS

Expertise: machine learning infrastructure (mobile runtimes, interpreters, operation sets, automatic differentiation, hardware acceleration), physical simulation (fluids, solids, rigid bodies), numerical algorithms, graphics, ray-tracing rendering (path tracing, volume rendering), performance optimization, parallel programming (distributed and shared memory).

Programming Languages: Fluent: C++, C, Python. Experienced: Java, JavaScript, Objective-C, GLSL, CUDA, VEX, RSL, SQL, PHP, Perl, Swift, Verilog.

Markup/Interchange: HTML, XML, JSON, LaTeX, PostScript.

Assembly: x86_64 (SSE/AVX intrinsics), MIPS, 68000, RISCV, Z80, 6502.

Tools: bazel, cmake, bash, scons, SWIG, sed, awk, perf, valgrind, gdb, lldb, VTune, PyBind.

Version control: Fluent: Git, Experienced: Mercurial, CVS, Perforce, RCS, SVN.

Machine learning: TensorFlow, TensorFlow Lite, JAX, MLIR.

Scientific computing: Eigen, LAPACK, Mathematica, Matlab, SciPy, NumPy, SymPy.

Libraries: Flatbuffers, Qt, LLVM, Boost, Sockets, Maya API, Houdini API, MPI, Pthreads, OpenMP, OpenGL, Protobuf, Win32, FITk, TBB, jsoncpp, libunwind, googletest, tcmalloc, OpenVDB, OpenEXR, Field3D, JNI.

Programming Environments: Linux/UNIX, Mac OS X, Windows, Android, iOS.