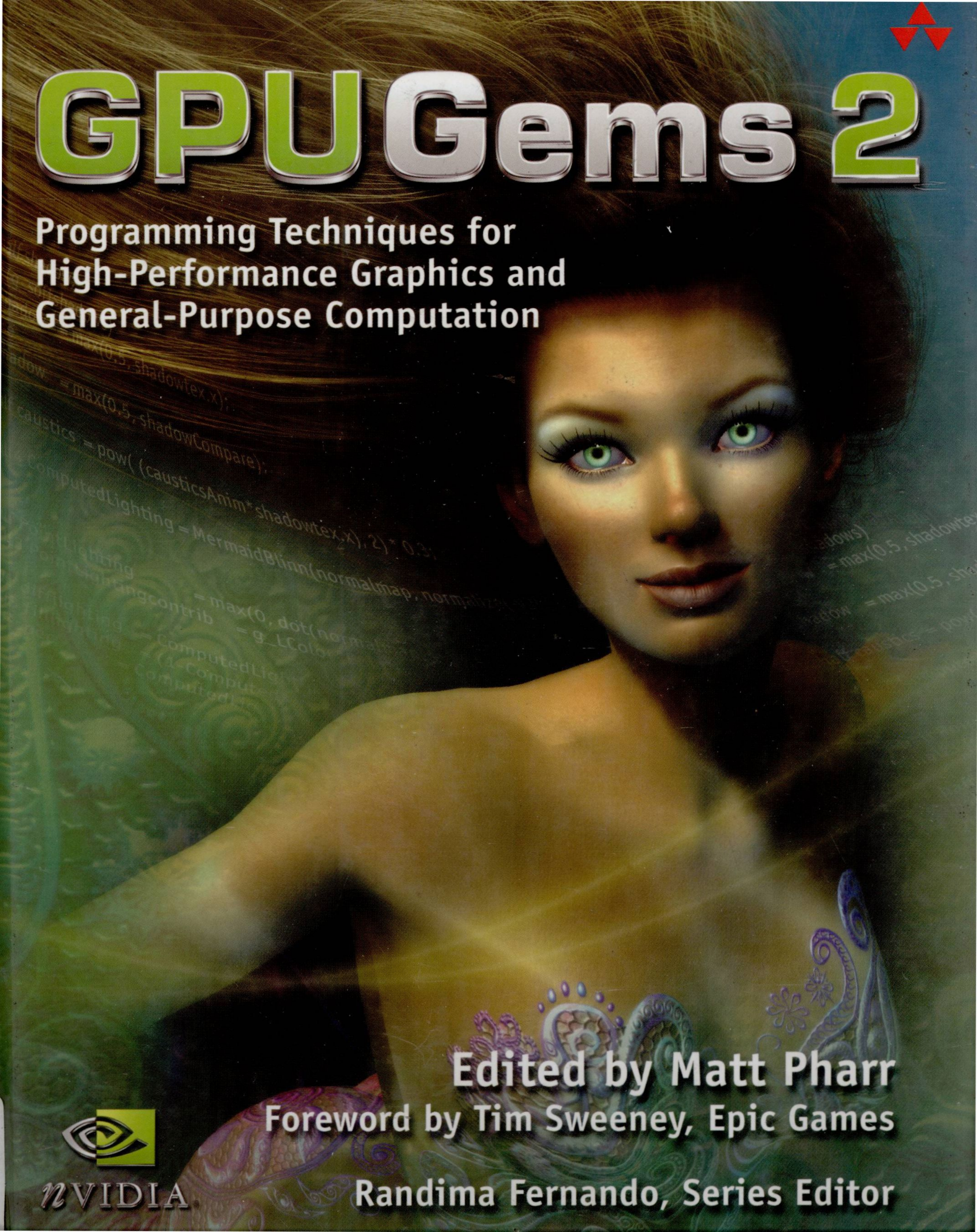


GPU Gems 2

Programming Techniques for High-Performance Graphics and General-Purpose Computation

Edited by Matt Pharr

Foreword by Tim Sweeney, Epic Games

NVIDIA

Randima Fernando, Series Editor

GPU Gems 2

Programming Techniques for High-Performance Graphics and General-Purpose Computation

Edited by Matt Pharr

Randima Fernando, Series Editor

Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

About the Cover: *The Nalu character was created by the NVIDIA Demo Team to showcase the rendering power of the GeForce 6800 GPU. The demo shows off advanced hair shading and shadowing algorithms, as well as iridescence and bioluminescence. Soft shafts of light from the water surface are blocked by her body, and her skin is lit by the light refracted through the water's surface, with her body and hair casting soft shadows on her as she swims.*

Library of Congress Cataloging-in-Publication Data
GPU gems 2 : programming techniques for high-performance graphics and general-purpose computation / edited by Matt Pharr ; Randima Fernando, series editor.
p. cm.
Includes bibliographical references and index.
ISBN 0-321-33559-7 (hardcover : alk. paper)
1. Computer graphics. 2. Real-time programming. I. Pharr, Matt. II. Fernando, Randima.

T385.G688 2005
006.66—dc22

2004030181

ISBN 0-321-33559-7

Text printed in the United States on recycled paper at Quebecor World Taunton in Taunton, Massachusetts.

Second printing, April 2005

To everyone striving to make
today's best computer graphics
look primitive tomorrow

Foreword

Before the advent of dedicated PC graphics hardware, the industry's first 3D games used CPU-based software rendering. I wrote the first Unreal Engine in that era, inspired by John Carmack's pioneering programming work on *Doom* and *Quake*. Despite slow CPUs and low resolutions, the mid-1990s became a watershed time for graphics and gaming. New visual effects appeared almost monthly, marked by milestones like *Quake*'s light mapping and shadowing and Unreal's colored lighting and volumetric fog. That era faded away as fixed-function 3D accelerators appeared. Deprived of the programmability that drove innovation and differentiation, 3D games grew indistinct.

Today, a new Renaissance in 3D graphics is under way, driven by fully programmable GPUs—*graphics processing units*—that deliver thousands of times the graphics power available just ten years ago. Combining incredible parallel computing power with modern, high-level programming languages, today's GPUs have unleashed a Cambrian Explosion of innovation and creativity. Real-time soft shadowing, accurate lighting models, and realistic material interactions are readily achievable. But the most important gain of programmability is that you can do *anything* with a GPU so long as you can find an algorithm to express your idea. *GPU Gems 2* demonstrates many such ideas-turned-algorithms.

Let us take a moment to review the set of resources available to today's graphics programmer. First, you have access to a GPU that can perform tens of billions of floating-point calculations per second in programmable shading algorithms. It's your workhorse; if you can move your problem into the realm of pixels and vertices, then you can harness the GPU's immense power. Second, you have a CPU, the system's general-purpose computing engine. The CPU sends commands to the graphics processing unit, manages resources, and interacts with the outside world. Finally, you have access to artistic content—texture maps, meshes, and other multimedia data that the GPU can combine, filter, and procedurally modify during rendering.

The Gems in this book employ these resources in novel ways to render realistic scenes, process images, and produce special effects. In doing so, many of the previous era's

graphics rules may be broken. GPUs are fast and flexible enough that you may render a given object many times, decomposing a scene into its components—lighting, shadowing, reflections, post-processing effects, and so on. You can employ the GPU for decidedly non-graphics tasks like collision detection, physics, and numerical computation; and within texture maps you can encode arbitrary data, such as vectors, positions, or lookup tables used by shader programs. And while visual realism is now achievable on GPUs, it is not your only option: nonphotorealistic rendering techniques are available, such as cel shading, exaggerated motion blur and light blooms, and other effects seen frequently in Hollywood productions.

Seven years after I wrote Unreal's original software renderer, my company began developing a new game engine, Unreal Engine 3, designed for the capabilities of today's modern GPUs. It has been an incredible experience! Where we once built 300-polygon scenes with static lighting and texture maps, we now combine dynamic per-pixel lighting and shadowing with realistic material effects in million-polygon scenes. We've seen an explosive growth in the power and flexibility available to programmers and artists alike. But while much has changed in graphics development, several truths have remained: that graphics requires a unique combination of engineering, artistry, and invention unmatched in other fields; that innovation moves at an incredible pace as hardware performance increases exponentially; and that graphics programming is *a heck of a lot of fun!*

Here in *GPU Gems 2*, you'll find a wealth of knowledge and insight, plus many just plain neat ideas, which can be readily applied on today's graphics hardware. But the techniques here are only a starting point on your adventure—the real fun and opportunity lie in finding new ways to customize and combine these Gems and to invent new ones.

Tim Sweeney
Founder and Technical Director, Epic Games

Screenshots from Unreal Engine 3 Technology Demo, http://www.unrealtechnology.com

Preface

The first volume of *GPU Gems* was conceived in the spring of 2003, soon after the arrival of the first generation of fully programmable GPUs. The resulting book was released less than a year later and quickly became a best seller, providing a snapshot of the best ideas for making the most of the capabilities of the latest programmable graphics hardware.

GPU programming is a rapidly changing field, and the time is already ripe for a sequel. In the handful of years since programmable graphics processors first became available, they have become faster and more flexible at an incredible pace. Early programmable GPUs supported programmability only at the vertex level, while today complex per-pixel programs are common. A year ago, real-time GPU programs were typically tens of instructions long, while this year's GPUs handle complex programs hundreds of instructions long and still render at interactive rates. Programmable graphics has even transcended the PC and is rapidly spreading to consoles, handheld gaming devices, and mobile phones.

Until recently, performance-conscious developers might have considered writing their GPU programs in assembly language. These days, however, high-level GPU programming languages are ubiquitous. It is extremely rare for developers to bother writing assembly for GPUs anymore, thanks both to improvements in compilers and to the rapidly increasing capabilities of GPUs. (In contrast, it took many more years before game developers switched from writing their games in CPU assembly language to using higher-level languages.)

This sort of rapid change makes a "gems"-style book a natural fit for assembling the state of the art and disseminating it to the developer community. Featuring chapters written by acknowledged experts, *GPU Gems 2* provides broad coverage of the most exciting new ideas in the field.

Innovations in graphics hardware and programming environments have inspired further innovations in how to use programmability. While programmable shading has long been a staple of offline software rendering, the advent of programmability on GPUs has led to the invention of a wide variety of new techniques for programmable shading. Going far beyond procedural pattern generation and texture composition, the state of the art of using shaders on GPUs is rapidly breaking completely new ground, leading to novel techniques for animation, lighting, particle systems, and much more.

Indeed, the flexibility and speed of GPUs have fostered considerable interest in doing computations on GPUs that go beyond computer graphics: general-purpose computation on GPUs, or "GPGPU." This volume of the *GPU Gems* series devotes a significant number of chapters to this new topic, including an overview of GPGPU programming techniques as well as in-depth discussions of a number of representative applications and key algorithms. As GPUs continue to increase in performance more quickly than CPUs, these topics will gain in importance for more and more programmers because GPUs will provide superior results for many computationally intensive applications.

With this background, we sent out a public call for participation in *GPU Gems 2*. The response was overwhelming: more than 150 chapters were proposed in the short time that submissions were open, covering a variety of topics related to GPU programming. We were able to include only about a third of them in this volume; many excellent submissions could not be included purely because of constraints on the physical size of the book. It was difficult for the editors to whittle down the chapters to the 48 included here, and we would like to thank everyone who submitted proposals.

The accepted chapters went through a rigorous review process in which the book's editors, the authors of other chapters in the same part of the book, and in some cases additional reviewers from NVIDIA carefully read them and suggested improvements or changes. In almost every case, this step noticeably improved the final chapter, due to the high-quality feedback provided by the reviewers. We thank all of the reviewers for the time and effort they put into this important part of the production process.

Intended Audience

We expect readers to be familiar with the fundamentals of computer graphics and GPU programming, including graphics APIs such as Direct3D and OpenGL, as well as GPU languages such as HLSL, GLSL, and Cg. Readers interested in GPGPU programming may find it helpful to have some basic familiarity with parallel programming concepts.

Developers of games, visualization applications, and other interactive applications, as well as researchers in computer graphics, will find *GPU Gems 2* an invaluable daily resource. In particular, those developing for next-generation consoles will find a wealth of timely and applicable content.

Trying the Examples

GPU Gems 2 comes with a CD-ROM that includes code samples, movies, and other demonstrations of the techniques described in the book. This CD is a valuable supplement to the ideas explained in the book. In many cases, the working examples provided by the authors will provide additional enlightenment. You can find sample chapters, updated CD content, supplementary materials, and more at the book's Web site, **http://developer.nvidia.com/GPUGems2/**.

Acknowledgments

An enormous amount of work by many different people went into this book. First, the contributors wrote a great collection of chapters on a tight schedule. Their efforts have made this collection as valuable, timely, and thought provoking as it is.

The section editors—Kevin Bjorke, Cem Cebenoyan, Simon Green, Mark Harris, Craig Kolb, and Matthias Wloka—put in many hours of hard work on this project, working with authors to polish their chapters and their results until they shone, consulting with them about best practices for GPU programming, and gently reminding them of deadlines. Without their focus and dedication, we'd still be working through the queue of submissions. Chris Seitz also kindly took care of many legal, logistical, and business issues related to the book's production.

Many others at NVIDIA also contributed to *GPU Gems 2*. We thank Spender Yuen once again for his patience while doing a wonderful job on the book's diagrams, as well as on the cover. Helen Ho also helped with the illustrations as their number grew to more than 150. We are grateful to Caroline Lie and her team for their continual support of our projects. Similarly, Teresa Saffaie and Catherine Kilkenny have always been ready and willing to provide help with copyediting as our projects develop. Jim Black coordinated communication with a number of developers and contributors, including Tim Sweeney, to whom we are grateful for writing a wonderfully focused and astute Foreword.

At Addison-Wesley Professional, Peter Gordon, Julie Nahil, and Kim Boedigheimer oversaw this project and helped to expedite the production pipeline so we could release this book in as timely a manner as possible. Christopher Keane's copyediting skills and Jules Keane's assistance improved the content immeasurably, and Curt Johnson helped to market the book when it was finally complete.

The support of several members of NVIDIA's management team was instrumental to this project's success. Mark Daly and Dan Vivoli saw the value of putting together a second volume in the *GPU Gems* series and supported this book throughout. Nick Triantos allowed Matt the time to work on this project and gave feedback on a number of the GPGPU chapters. Jonah Alben and Tony Tamasi provided insightful perspectives and valuable feedback about the chapter on the GeForce 6 Series architecture. We give sincere thanks to Jen-Hsun Huang for commissioning this project and fostering the innovative, challenging, and forward-thinking environment that makes NVIDIA such an exhilarating place to work.

Finally, we thank all of our colleagues at NVIDIA for continuing to push the envelope of computer graphics day by day; their efforts make projects like this possible.

Matt Pharr
NVIDIA Corporation

Randima (Randy) Fernando
NVIDIA Corporation

Contributors

Tomas Akenine-Möller, Lund University

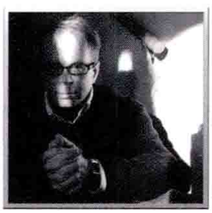
Tomas Akenine-Möller is an associate professor in the department of computer science at Lund University in Sweden. His main interests lie in real-time rendering, graphics on mobile devices, and shadows.

Arul Asirvatham, Microsoft Research

Arul Asirvatham is a Ph.D. student in the School of Computing, University of Utah. He received a B.Tech. in computer science and engineering in 2002 from the Indian Institute of Information Technology in India. His primary research interest is digital geometry processing; he has been working on mesh parameterization techniques. He is also interested in real-time computer graphics. Currently he is focusing on rendering huge terrain data sets interactively.

Jiří Bittner, Vienna University of Technology

Jiří Bittner is currently affiliated with the Institute of Computer Graphics and Algorithms of the Vienna University of Technology. He received his Ph.D. in 2003 from the department of computer science and engineering of the Czech Technical University in Prague. His research interests include visibility computations, efficient real-time rendering techniques, global illumination, and computational geometry.

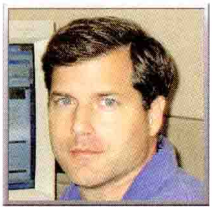


Kevin Bjorke, NVIDIA Corporation

Kevin Bjorke is a member of the Developer Technology group at NVIDIA. He was a section editor and authored several chapters for *GPU Gems*. He has an extensive and award-winning production background in live-action and computer-animated films, television, advertising, theme park rides, and, of course, games. Kevin has been a regular speaker at events such as Game Developers Conference (GDC) and ACM SIGGRAPH since the mid-1980s. His current work at NVIDIA involves exploring and harnessing the power of programmable shading for high-quality real-world applications.

Ian Buck, Stanford University

Ian Buck is completing his Ph.D. in computer science at the Stanford Computer Graphics Lab, researching general-purpose computing models for GPUs. He received a B.S.E. in computer science from Princeton University in 1999 and received fellowships from the Stanford School of Engineering and NVIDIA. His research focuses on programming language design for graphics hardware as well as general-computing applications that map to graphics hardware architectures.



Michael Bunnell, NVIDIA Corporation

Michael Bunnell graduated from Southern Methodist University with degrees in computer science and electrical engineering. He wrote the Megamax C compiler for the Macintosh, Atari ST, and Apple IIGS before cofounding what is now LynuxWorks. After working on real-time operating systems for nine years, he moved to Silicon Graphics, focusing on image-processing, video, and graphics software. Next, he worked at Gigapixel, then at 3dfx, and now at NVIDIA, where, interestingly enough, he is working on compilers again—this time, shader compilers.

Iain Cantlay, Climax Entertainment

Iain Cantlay is currently a senior engineer at Climax, where he was responsible for the graphical aspects of the *Leviathan* MMO engine and *Warhammer Online*. His current projects include *MotoGP 3* (to be published for Xbox and PC by THQ in 2005). Iain is passionate about exploiting the best visuals from the latest technology, but natural phenomena interest him most: terrain, skies, clouds, vegetation, and water.

Francesco Carucci, Lionhead Studios

Francesco Carucci graduated from the Politecnico di Torino in Italy with a degree in software engineering. When he was eight, rather than make pizza (like every good Italian), he decided to make video games, and he tried to animate a running character in BASIC on an Intellivision. He is now writing code to animate running characters at Lionhead, working on the latest rendering technology for *Black & White 2*. He contributed to various Italian technical 3D sites and to *ShaderX2*. His main interests include lighting and shadowing algorithms, 3D software construction, and the latest 3D hardware architectures. And when he needs help, he writes shaders for food.

Cem Cebenoyan, NVIDIA Corporation

Cem Cebenoyan is a software engineer working in the Developer Technology group at NVIDIA. He was an author and section editor for *GPU Gems*. He spends his days researching graphics techniques and helping game developers get the most out of graphics hardware. He has spoken at past Game Developer Conferences on character animation, graphics performance, and nonphotorealistic rendering. Before joining NVIDIA, he was a student and research assistant in the Graphics, Visualization, and Usability Lab at the Georgia Institute of Technology.

Eric Chan, Massachusetts Institute of Technology

Eric Chan is a Ph.D. student in the Computer Science and Artificial Intelligence Laboratory at M.I.T. He fiddles with graphics architectures, shading languages, and real-time rendering algorithms. He has recently developed efficient methods for rendering hard and soft shadows. Before attending graduate school, Eric was a research staff member in the Stanford Computer Graphics Laboratory. As part of the Real-Time Programmable Shading team, he wrote compiler back ends for the NV30 and R300 fragment architectures and developed a pass-decomposition algorithm for virtualizing hardware resources. Eric enjoys photography and spends an unreasonable amount of his free time behind the camera.

Greg Coombe, The University of North Carolina at Chapel Hill

Greg Coombe is a graduate student at the University of North Carolina at Chapel Hill. He received a B.S. in mathematics and a B.S. in computer science from the University of Utah in 2000. Greg's research interests include global illumination, graphics hardware, nonphotorealistic rendering, virtual environments, and 3D modeling. During the course of his graduate studies, he has worked briefly at Intel, NVIDIA, and Vicious Cycle Software. Greg was the recipient of the NVIDIA Graduate Fellowship in 2003 and 2004.

Jürgen Döllner, University of Potsdam, Hasso-Plattner-Institute

Jürgen Döllner, a professor at the Hasso-Plattner-Institute of the University of Potsdam, directs the computer graphics and visualization division. He has studied mathematics and computer science and received a Ph.D. in computer science. He researches and teaches in real-time computer graphics and spatial visualization.

William Donnelly, NVIDIA Corporation and University of Waterloo

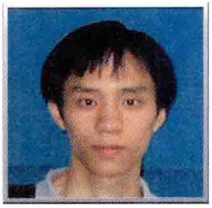
William Donnelly is a fourth-year undergraduate in computer science and mathematics at the University of Waterloo in Ontario. He interned with Okino Computer Graphics, where he worked on global illumination and volumetric rendering; and with NVIDIA's Demo Team, where he worked on the "Last Chance Gas" and "Nalu" demos. He has been destined for greatness in computer graphics since mastering the art of the Bezier spline at age ten.

Frédo Durand, Massachusetts Institute of Technology

Frédo Durand received a Ph.D. from Grenoble University in France in 1999, where he worked on both theoretical and practical aspects of 3D visibility. From 1999 until 2002, he was a postdoc in the M.I.T. Computer Graphics Group, where he is now an assistant professor. His research interests span most aspects of picture generation and creation, including realistic graphics, real-time rendering, nonphotorealistic rendering, and computational photography. He received a Eurographics Young Researcher Award in 2004. *(Digital drawing courtesy of Victor Ostromoukhov)*

Eric Enderton, NVIDIA Corporation

Eric Enderton is a senior engineer at NVIDIA, where he is working on the Gelato film renderer. After studying computer science at the University of California, Berkeley, Eric spent a decade developing rendering and animation software at Industrial Light & Magic, and he later consulted at other studios. His film credits include *Terminator 2*; *Jurassic Park*; and *Star Wars, Episode I: The Phantom Menace*.

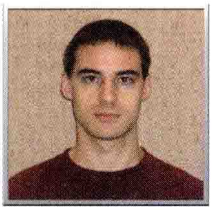


Zhe Fan, Stony Brook University

Zhe Fan is a Ph.D. candidate in the computer science department at Stony Brook University. He received a B.S. in computer science from the University of Science and Technology of China in 1998 and an M.S. in computer science from the Chinese Academy of Sciences in 2001. His current research interests include GPU clusters for general-purpose computation, parallel graphics and visualization, and modeling of amorphous phenomena.

Randima Fernando, NVIDIA Corporation

Randima (Randy) Fernando has loved computer graphics since age eight. Working in NVIDIA's Developer Technology group, he helps teach developers how to take advantage of the latest GPU technology. Randy has a B.S. in computer science and an M.S. in computer graphics, both from Cornell University. He has published research in SIGGRAPH and is coauthor, with Mark Kilgard, of *The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics*. He edited *GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics* and is the *GPU Gems* series editor.

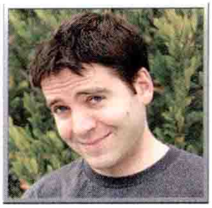


Nathaniel Fout, University of California, Davis

Nathaniel Fout received a B.S. in chemical engineering and an M.S. in computer science from the University of Tennessee in 2002 and 2003, respectively. He is a Ph.D. student in computer science at the University of California, Davis, where he is a member of the Institute for Data Analysis and Visualization. His research interests include volumetric compression for rendering, multivariate and comparative visualization, and tensor visualization.

James Fung, University of Toronto

James Fung is completing his Ph.D. in engineering. He received a B.A.Sc. in engineering science and an M.S. in electrical engineering from the University of Toronto. His research interests include wearable computing, mediated reality, and exploring new types of musical instrument interfaces based on EEG brain-wave signal processing. His most recent work has been the development of the GPU-based computer vision and mediated reality library called OpenVIDIA.



Simon Green, NVIDIA Corporation

Simon Green is a senior software engineer in the Developer Technology group at NVIDIA. He started graphics programming on the Sinclair ZX-81, which had 1 kB of RAM and a screen resolution of 64×48 pixels. He received a B.S. in computer science from the University of Reading, in the United Kingdom, in 1994. Since 1999 Simon has found a stable home at NVIDIA, where he develops new rendering techniques and helps application developers take maximum advantage of GPU hardware. He is a frequent presenter at GDC, has written for *Amiga Shopper* and *Wired* magazines, and was a section editor for *GPU Gems*. His research interests include cellular automata, general-purpose computation on GPUs, and analog synthesizers.

Toshiya Hachisuka, University of Tokyo

Toshiya Hachisuka is an undergraduate in the Department of Systems Innovation at the University of Tokyo. He also works as a programmer for MagicPictures, integrating cutting-edge research results into current computer graphics software. He has studied computer graphics since age ten. His current research interests are physically based rendering, physically based modeling, real-time rendering techniques, and general-purpose computation on GPUs.

Markus Hadwiger, VRVis Research Center

Markus Hadwiger received his Ph.D. in computer science from the Vienna University of Technology in 2004, where he concentrated on high-quality real-time volume rendering and texture filtering with graphics hardware, in cooperation with the VRVis Research Center. He has been a researcher at VRVis since 2000, working in the Basic Research on Visualization group and the Medical Visualization group (since 2004). From 1996 to 2001, he was also the lead programmer of the cross-platform 3D space-shooter game *Parsec*, which is now an open source project.

Mark Harris, NVIDIA Corporation

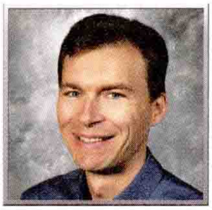
Mark Harris received a B.S. from the University of Notre Dame in 1998 and a Ph.D. in computer science from the University of North Carolina at Chapel Hill (UNC) in 2003. At UNC, Mark's research covered a wide variety of computer graphics topics, including real-time cloud simulation and rendering, general-purpose computation on GPUs, global illumination, nonphotorealistic rendering, and virtual environments. Mark is now a member of NVIDIA's Developer Technology team based in the United Kingdom.

Jon Hasselgren, Lund University

Jon Hasselgren received an M.Sc. from Lund University. He now pursues graduate studies in the computer science department, where he researches graphics for mobile phones.

Oliver Hoeller, Piranha Bytes

Oliver Hoeller is a senior software engineer at Piranha Bytes, which developed the RPGs *Gothic I* and *Gothic II*. Previously he was director of development at H2Labs/Codecult, where he was responsible for development and architecture design of the Codecreatures game system. He was an active member of the German demo scene in the 1980s and early 1990s. After exploring different areas—developing music software, creating a security program, and working as a Web services consultant—Oliver returned to his roots and now guarantees a high level of visual quality for Piranha Bytes' forthcoming *Gothic III*.



Hugues Hoppe, Microsoft Research

Hugues Hoppe is a senior researcher in the Computer Graphics Group at Microsoft Research. His primary interests lie in the acquisition, representation, and rendering of geometric models. He received the 2004 ACM SIGGRAPH Achievement Award for his pioneering work on surface reconstruction, progressive meshes, geometry texturing, and geometry images. His publications include twenty papers at ACM SIGGRAPH, and he is associate editor of *ACM Transactions on Graphics*. He received a B.S. in electrical engineering in 1989 and a Ph.D. in computer science in 1994 from the University of Washington.

Daniel Horn, Stanford University

Daniel Horn is a Ph.D. candidate at the Stanford Computer Graphics Lab; he received his B.S. from the University of California, Berkeley. While Daniel focuses on programming graphics hardware and real-time graphics, theory and compilers have always interested him deeply, and he tries to incorporate knowledge from those fields into his graphics research. In his spare time, Daniel enjoys hacking with his brother, Patrick, on their open source space sim, *Vega Strike*. He also enjoys roaming with friends in the Bay Area's many natural parks, from Palo Alto's Foothills Park to Berkeley's Tilden Park.

Samuel Hornus, GRAVIR/IMAG–INRIA

Samuel Hornus is a Ph.D. candidate at INRIA in Grenoble, France. He is a former student of the Ecole Normale Supérieure de Cachan. His research focuses on 3D visibility problems, as well as other aspects of computer graphics, such as texture authoring, interactive walkthroughs, real-time shadows, realistic rendering, implicit surfaces, and image-based modeling.

Arie Kaufman, Stony Brook University

Arie Kaufman is the director of the Center for Visual Computing, a distinguished professor and chair of the Computer Science Department, and distinguished professor of radiology at Stony Brook University. He received a B.S. in mathematics and physics from the Hebrew University of Jerusalem in 1969; an M.S. in computer science from the Weizmann Institute of Science, Rehovot, Israel, in 1973; and a Ph.D. in computer science from the Ben-Gurion University, Israel, in 1977. Kaufman has conducted research and consulted for more than thirty years, with numerous publications in volume visualization; graphics architectures, algorithms, and languages; virtual reality; user interfaces; and multimedia.

Jan Kautz, Massachusetts Institute of Technology

Jan Kautz is a postdoctoral researcher at M.I.T. He is particularly interested in realistic shading and lighting, hardware-accelerated rendering, textures and reflection properties, and interactive computer graphics. He received his Ph.D. in computer science from the Max-Planck-Institut für Informatik in Germany; a diploma in computer science from the University of Erlangen in Germany; and an M.Math. from the University of Waterloo in Ontario.

Emmett Kilgariff, NVIDIA Corporation

Emmett Kilgariff is a director of architecture in the GPU group at NVIDIA, where he has contributed to the design of many GeForce chips, including the GeForce 6 and GeForce 7 Series. He has more than twenty years of experience designing graphics hardware, at Sun Microsystems, Silicon Graphics, 3dfx, and many small companies whose memories have faded over time.

Gary King, NVIDIA Corporation

Unscrupulous. Unconventional. Uncouth. Unkempt. All are accurate adjectives for the worst thing to happen to the graphics industry since Execute Buffers. A master of GPU arcana, lore, and the occult, he spends his days at NVIDIA crafting increasingly ingeniously nefarious rendering techniques, imbuing next-generation architectures with unholy energies, worshipping the Dark Lord, and kicking puppies.

Peter Kipfer, Technische Universität München

Peter Kipfer is a postdoctoral researcher in the Computer Graphics and Visualization Group at the Technische Universität München. He received his Ph.D. from the University of Erlangen-Nürnberg in 2003 for his work on parallel and distributed visualization and rendering within the KONWIHR supercomputing project. His current research focuses on general-purpose computing and geometry processing on the GPU.