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/**.

Acknowledgements

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, 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