My Personal Review:
This book is the second installment of the GPU Gems book series by NVIDIA. Just like the first book, GPU Gems 2 is a collection of articles by various authors from game development companies, academia, and tool developers on advanced techniques for programming graphics processing units (or GPUs for short). It is aimed at intermediate to advanced graphics developers that are familiar with the most common graphics APIs. The reader should also be proficient in C++. As with the first GPU Gems book, GPU Gems 2 is not for beginners. For professional graphics and game developers, however, it is an excellent collection of interesting techniques, tips, and tricks.

The book is divided into six parts, each dealing with a different aspect of GPU programming. Compared to the first book, more emphasis is put on the quickly evolving area of general-purpose computation on GPUs (also called GPGPU). In particular, the last three of the six parts of the book are about GPGPU and its applications. The first three parts, however, are about real-time computer graphics.

The first part of the book contains 8 chapters on photo-realistic rendering that mostly deal with how to efficiently render a large number of objects in a scene, which is a necessity for rendering convincing natural effects, such as grass or trees. Two chapters in this part of the book discuss geometry instancing and segment buffering, and another chapter focuses on using occlusion queries to implement coherent hierarchical occlusion culling. Other interesting topics in this part of the book include adaptive tessellation of surfaces on the GPU, displacement mapping - an extension to the popular parallax mapping used in some current games - that allows to
render realistic bumps on a simple quad, and terrain rendering with geometry clipmaps.

Part two of the book consisting of 11 chapters deals with shading and lighting. This part contains highly interesting chapters on deferred shading in the game S.T.A.L.K.E.R., and computing irradiance environment maps on the GPU in real-time. Furthermore, this part of the book has chapters on rendering atmospheric scattering, implementing bidirectional texture functions on the GPU, dynamic ambient occlusion culling, water rendering, and using shadow mapping with percentage-closer filtering to achieve soft shadows.

The third part of the book consists of 9 chapters on high-quality rendering. Most chapters in this part deal with implementing high-quality filtering in fragment shaders. For example, there is an interesting chapter on filtered line rendering and another chapter on cubic texture filtering. Finally, a GPU-only implementation of improved Perlin Noise is also presented in this part of the book.

The chapters in the fourth part of the book represent an introduction to the fantastic field of GPGPU. The 8 chapters of this part first describe the general streaming architecture of GPUs, and then move on to show how to map conventional CPU data structures and algorithms to the GPU. For example, textures can be regarded as the GPU equivalent to CPU data arrays. There is also a chapter on how to implement flow-control idioms on the GPU and a chapter on optimizing GPU programs.

The 6 chapters of part five of the book are on image-oriented computing and describe a number of GPGPU algorithms for performing global illumination computations, for example by using radiosity, on the GPU. There is also a chapter on doing computer vision on the GPU.

The final chapter in this part of the book explains how to perform conservative rasterization, which is important for some GPGPU algorithms to achieve accurate results.

The final part of the book has 6 chapters that present GPGPU techniques to perform a variety of simulation and numerical algorithms on the GPU. One chapter shows how to map linear algebra operations onto the GPU and develops a GPU framework to solve systems of linear equations. In other chapters the GPU is used for protein structure prediction, options pricing, flow simulation, and medical image reconstruction. These chapters show good examples of how the GPU can be used for non-graphics-related tasks.

The book contains many illustrations and diagrams that visualize the results of certain techniques or explain the presented algorithms in more detail. All images in the book are in color, which is definitely advantageous for a graphics book. In my opinion, the excellent quality and also the
quantity of images and illustrations is one of the strongest points of this book compared to other graphics books.

The book also comes with a CD-ROM with supplemental material, videos, and demo applications to some chapters. Most of the applications include the full source code, which makes it easy to experiment with the techniques presented in the book. Note that most of the applications run on Windows only and many of them require a shader model 3.0 graphics card.

I highly recommend this book to any professional working as graphics or game developer. It is a valuable addition to my library of graphics books and I will come back to a number of articles in the near future. The focus on GPGPU in the second half of the book is a welcome addition and we can expect to see more and more non-graphics-related applications make use of the processing power in todays GPUs.

For More 5 Star Customer Reviews and Lowest Price: