GPU Zen Frequently Asked Questions

Update April 26, 2018.

What is this book about?

A group of authors publish short explanations of their graphics programming tricks and tips. Articles are grouped into sections, and each section has a dedicated editor. This is similar in concept to the prior *ShaderX* and *GPU Pro* book series.

What is the target audience of the book?

The book is intended for intermediate to advanced graphics programmers.

What is required for an article proposal?

A proposal needs to show that you actually implemented the technique. With that in mind, screenshots are the most important along with a short description of the algorithm. An sample program that demonstrates the technique is also helpful. Techniques that have shipped or will ship in a commercial game are especially desirable for the book.

If my proposal is accepted, does that mean my article will be published?

In order for an article to published, the author must:

1. Meet a baseline level of quality, including proper language and proof that the technique works.

2. Meet all review and publication deadlines.

How many pages are expected for each contribution?

The page count depends on the subject: more complex topics will need more page, and simpler topics will need fewer. External references should mitigate the need for extensive review of existing work. 15-20 printed pages is typical.

What writing standards are expected?

A lot of readers won't be native English speakers, so it is important to write using English that is easy to translate. There are therefore two challenges: you have to explain your topic for the target audience of intermediate and advanced programmers as well as for the non-English-speaking readers in that target audience. Diagrams, screenshots and code snippets will help with the language barrier. Videos (where applicable) should also be developed, after the text, images, and sample code.

A suggestions for a rough article is as follows:

1. What is the topic of this paper or this section of this paper?
2. Why is the topic important? Where is the topic used or applicable?
3. Presentation of any theory behind the topic, derivations, etc.
4. Algorithmic statement of the technique.
5. Presentation of the technique.

6. Analysis, including strengths, weaknesses, and future work.
7. Concluding remarks.
8. References.

(1) and (2) are important for setting the context. (4) makes understanding (5) much easier.

Spell checking prior to submission is expected, in order to reduce avoidable work during the review process.

What image formats are supported?

Lossless image formats such as PNG and TIFF are preferred. For diagrams, it is helpful to have vector art, e.g. in SVG format. Images should be provided separate from the main layout file.

What are the rules for citation?

We recommend looking at previous editions of *ShaderX*, *GPU Pro*, and *GPU Zen* for examples of citations. Some examples from the first chapter *GPU Zen* are included below. Please also consult the writing guidelines from the publisher.

BUSCHMANN, S., TRAPP, M., AND DÖLLNER, J. 2015. Animated visualization of spatial–temporal trajectory data for air-traffic analysis. *The Visual Computer*, 1–11.

CARUCCI, F., AND STUDIOS, L. 2005. Inside geometry instancing. In *GPU Gems 2*, M. Pharr and R. Fernando, Eds. Addison-Wesley Professional, 47–67.

KEMEN, B., 2012. Procedural grass rendering. URL: http://outerra.blogspot.de/2012/05/procedural-grass-rendering.html.

OVERVOORDE, A., 2014. Geometry shaders. URL: https://open.gl/geometry.

PURNOMO, B., BILODEAU, J., COHEN, J. D., AND KUMAR, S. 2005. Hardware-compatible vertex compression using quantization and simplification. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, ACM, New York, 53–61.

What languages and coding conventions are expected?

The preferred language for CPU code is C/C++ and for GPU code it is HLSL or GLSL. The target audience will have experience with a variety of coding conventions. The most common and familiar convention is likely Hungarian notation, applied to both CPU and GPU code. See previous previous editions of *ShaderX*, *GPU Pro*, and *GPU Zen* for examples.

What graphics hardware should be targeted?

You can target any commercially-available hardware for your technique, whether for desktop computers, mobile devices, or game consoles.

What are the expectations for sample code?

Every article must include some sample source code. If it is possible to build the code into a running application, you should all files necessary to build and run it using a commercially-available development tools. This includes CPU source code, shader source code, solution / make files, art assets, etc. Pre-compiled binaries are optional.

To assist with the development process, we recommend using The Forge, a cross-platform rendering framework available here: https://github.com/ConfettiFX/The-Forge.

All source code will be publicly shared on GitHub. See https://github.com/wolfgangfengel/GPUZen for examples.

What are the expectations for demo videos?

Demo videos are not mandatory but certainly helpful. They should be rendered in a format compatible with YouTube; H.264/MP4 at a 16:9 aspect ratio is preferred.

How do I share my manuscript for review?

You will be put in touch with the section editor. You can email the files or use a file-sharing service like Google Drive or Dropbox to share files.

What is the typical workflow for an article?

All articles proceed through the following phases:

1. **Proposal phase.** The author presents an articles proposal based on the template provided.
2. **Writing begins.** Following acceptance of the proposal, the section editor will work with the author on goals and deadlines.
3. **First draft.** This draft should be “feature-complete” and from the perspective of the author in a publishable state.
4. **Review and editing.** The section editor and reviewers will go through the article and provide feedback on the draft. This may involve simple corrections or substantial rework depending on the quality of the draft.
5. **Final draft.** The final draft is sent from the section editor to the main editor. The main editor will review the article before it gets printed and might come back to the author with questions and suggestions.

Please note that the only guaranteed feedback phase is step 4 above.