Cyprien Buron

Research Engineer
in Computer Graphics, Ph.D.

2, rue Pierre-Henri Teitgen
35000 Rennes
France
+33 6 13 26 75 10
Age : 27
cyprien.buron@gmail.com
cyprien.buron.free.fr

Scientific skills

Modeling and rendering on the GPU

Real-time generation and visualization of massive 3D environments, complex scenes lighting techniques, interactive procedural modeling, algorithms parallelization, scalable pipelines from mobile devices to high-end computers, GPU architectures, hierarchical and acceleration structures, parametric curves and surfaces.

Experience

2014 – 2015 **Post-doctoral researcher** — *Technicolor, Rennes*, France

- Definition and development of an on-mesh procedural modeling pipeline [1][2], published and presented at Siggraph.
- Experimentation of a Javascript-based system for procedural generation, aiming live editing on mobile devices.
- Analysis and research on tiled rendering algorithms for future integration in the in-house rendering plateform, and planned to be transfered.
- Development and integration of lighting and rendering features.

2010 – 2013 **Ph.D. student in Computer Graphics** — *Technicolor, Rennes*, France

Subject: Large scale generation and rendering of complex 3D models on the GPU.

- Supervision by Jean-Eudes Marvie at Technicolor and Xavier Granier at LaBRI/INRIA.
- Definition of a grammar-based procedural generation pipeline, running at interactive time using graphics hardware. Allowing creation of highly detailed geometries, comprising a full LOD system, and handling huge sceneries of architecture and vegetation [4]. Enhancement to roof generation [3]. Analysis of a procedural generation system mapping any underlying mesh.
- Realistic lighting of virtual scenes using webcam [5].
- Development and integration of latest OpenGL features in the in-house rendering platform, used in multiple technology transfers.
- Publications and presentations in international conferences : Eurographics, Pacific Graphics, SCCG.
- 4 patents filled.

2010 **Real-time 3D rendering internship - 6 months** — *Technicolor, Rennes*, France

In-house mixed reality platform enhancement.

- Implementation of a real-time realistic skin rendering algorithm in GLSL.
- Creation of a modular shader generation tool.

2010 **Study project - 6 months** — *IFSIC/ESIR - Archividéo, Rennes*, France

Development of a 3DSMax script for automatic creation of power transmission lines.

2009 **Graphics developer internship - 3 months** — *NTNU, Trondheim*, Norway

3D parametric trees generation tool for hardware tessellation use.

2009 **Study project - 3 months** — *IFSIC/ESIR, Rennes, France*

Creation of a short animation movie with Blender.

2007 **Software developer internship - 4 months** — *SIB, Rennes, France*

Update of a medicine software for new database management.

Education

2010 – 2013 **Ph.D Thesis in Computer Graphics** — *Technicolor, Rennes*, France
Joint supervision with Manao team, LaBRI/INRIA, Université de Bordeaux I, France
"Interactive generation and rendering of massive scenes : a parallel procedural approach"

2007 – 2010 **Engineering degree in Computer Science** - *IFSIC/ESIR, Université de Rennes I*, France
Major in computer graphics
Image synthesis: rasterization, ray-tracing, global illumination, geometry modeling, animation.
Image processing: Fourier transforms, nonlinear filtering, image enhancement and restoration, edge detection, region growing and segmentation, pattern recognition, video compression.

2005 – 2007 **2-year course in Computer Science** — *Institut Universitaire de Technologie, Nantes*, France

Technical Skills

Programming
- Development of stand-alone applications and integration within existing softwares
- C/C++: 8 years
- Java: 2 years
- Scripting: Javascript, Maxscript
- X3D/Vrml scene-graph, X3DOM framework

GPU
- OpenGL: 7 years
- GLSL/Cg: 7 years
- OpenCL/CUDA
- WebGL

Softwares
- IDE: Visual Studio, Eclipse
- Rendering: 3DSMax, Blender, Maya
- Mathematics: Matlab, Octave
- Use of SVN, ClearCase, Git, CMake

Languages
- Fluent English, abroad internship in 2009 and conference presentations since 2011
- Native French
- Basic Spanish

Publications

International conferences, peer reviewed

[1] Technical paper in submission.

[2] Dynamic on-mesh procedural generation. Buron Cyprien, Marvie Jean-Eudes, Guennebaud Gaël, Granier Xavier. In ACM SIGGRAPH 2014 Talks. August 2014. Vancouver. Simulation Talks Session.

[3] GPU Roof Grammars. Buron Cyprien, Marvie Jean-Eudes, Gautron Pascal. In proceedings of Eurographics 2013. May 2013. Girona - Spain. Buildings and Stereo Short Papers Session.

[4] GPU Shape Grammars. Marvie Jean-Eudes, Buron Cyprien, Gautron Pascal, Hirtzlin Patrice, Sourimant Gaël. In proceedings of Pacific Graphics 2012. September 2012. Hong Kong.

[5] Grabbing Real Light - Toward Virtual Presence. Gautron Pascal, Jean-Eudes Marvie, Cyprien Buron. In proceedings of the 27th Spring Conference on Computer Graphics. April 2011. Viničné - Slovak Republic.

Misc 4 patents filled.

Technology transfers

Rendering engine

I was involved in the Technicolor's in-house rendering platform development. I was responsible for the renderer refactoring: separation of the core rendering algorithm from the graphics API; provides the ability to change the renderer profile at any time, including for mobile devices; allows for better performances according to the available hardware. I integrated some latest OpenGL features such as the tessellation shaders allowing the creation of thousands of polygons using dedicated cores of the graphics hardware; and transform feedback operation to capture and reuse geometry generated on the GPU. I also developed rendering features such as cube shadow maps. All these developments have been transfered to the Moving Picture Company (London, Los Angeles, Vancouver), Technicolor Animation and Games (Bangalore) through the pre-viz real-time rendering transfers.

Projects summary

2010 **Real-time skin rendering.** *Challenge:* High-quality skin rendering for real-time use in production. *Contributions:* Implementation of a sub-surface scattering algorithm in the in-house platform. Multiple texture-space passes harnessing the graphics hardware capabilities. *Application:* Realistic, production-ready, human skin rendering.

2010 **Modular shaders.** *Challenge:* Definition of modular shader bricks to further generic reassembly. *Contributions:* Decomposition of complex rendering shaders into unitary shader bricks (lighting, texturing, shading, fog, etc). Providing generic entry points and main structure to ensure maximum performances. *Application:* Modular shader prototype for the in-house rendering platform.

2011 **Real-time realistic lighting of virtual scenes.** *Challenge:* Efficient relighting of virtual scenes using dynamic and real environment. *Contributions:* Capturing the real lighting in dynamic environments with simple webcams instead of using high-end digital cameras and mirror balls in still scenes. Efficient filtering of the captured environment data using graphics hardware according to the reflectance functions of the virtual surfaces. Simple method to extrapolate the incoming lighting outside the webcam range (limited aperture). *Application:* Direct lighting of virtual objects in all virtual reality applications such as virtual worlds, games etc. Lighting design tool integrated for post-production, using either environment map combination or semi-automatic light source extraction.

2011 – 2012 **Interactive generation and rendering of massive 3D sceneries.** *Challenge:* Interactive generation, tuning and visualization of massive environment elements for both video games and production rendering. *Contributions:* Generation of detailed procedural models without explicit geometry storage. Reformulation of the grammar expansion for generation of detailed models at the tessellation control and geometry shader stages. Generation of massive, highly detailed models using the geometry generation capabilities of modern graphics hardware. Integration within a scalable framework by introducing automatic generation of levels of detail at reduced cost. *Application:* Interactive generation and rendering of scenes containing thousands of buildings and trees.

2013 **Parallelization of context-aware procedural models.** *Challenge:* Dynamic generation of complex structures depending on internal contexts, such as roofs. *Contributions:* Decomposition of the internal contexts to parallelize complex computation. Starting from a consistent internal context computed on the initial global structure, splitting into local context parameters per input segments compliant with a GPU architecture. Introduction of new grammar rules for efficient parallelization, bringing the massive parallelism capabilities of graphics hardware to the benefit of coherent generation of global structures. *Application:* Roof generation with straight skeleton computed from the building footprints, decomposed into independent local contexts.

2013 – 2014 **Dynamic on-mesh procedural generation.** *Challenge:* Interactive control of on-mesh growing shape grammars. *Contributions:* Introduction of a novel approach based on a marching rule on the GPU. Encoding of environment contexts as geometry texture atlases, on which indirection pixels are computed around each chart borders. Marching rule to walk through the texture atlas at run-time, and efficiently jump from chart to chart using indirection information. The underlying surface is thus followed during the grammar development. Use of additional texture information to easily constrain the grammar interpretation. Smooth geometry deformation using cubic Bezier curves. *Application:* Procedural growing ivy model. Dynamic on-mesh painting of authorized growth areas or leaves density with on-the-fly model adaptation.

2014 **Script-based procedural generation on mobile devices:** *Challenge:* Extension of the procedural engine to low-end mobile hardware with interactive performances. *Contributions:* Definition of a script-based procedural engine. Native interaction between kernel and grammar rules to generate Vrml data. Use of the x3dom framework to create WebGL scenes. *Application:* Procedural generation and rendering of hundreds of buildings on tablets at interactive time.

2015 **Tiled-rendering pipeline for heavy lighting computation.** *Challenge:* Interactive computation of very complex lighting scenes. *Contributions:* Integration and development of tiled-rendering algorithms within the in-house rendering platform. By dividing screen-space into multiple tiles or clusters, identification of light to tile/cluster contributions. At render time the lighting computation is then dramatically reduced. Efficient parallelization of the multi-pass algorithm using either rasterization or compute shaders, depending on the graphics hardware. *Application:* Interactive rendering of scenes comprising thousand of light sources.