Institution: Imperial College London		
Unit of Assessment: 11 Computer Science and Informatics		
Title of case study: GraphicsFuzz: Reliable and Secure Graphics Programming		
Period when the underpinning research was undertaken: Nov 2011 – Jun 2018		
Details of staff conducting the underpinning research from the submitting unit:		
Name(s):	**Role(s) (e.g. job title):**	**Period(s) employed by submitting HEI:**
Alastair F. Donaldson	Professor	Nov 2011 – present
Hugues Evrard	Research Associate	Oct 2016 – May 2018
Jeroen Ketema	Research Associate	Apr 2012 – Mar 2015
Paul Thomson	Research Associate	Sep 2016 – Apr 2017
Period when the claimed impact occurred: 1 Aug 2013 – 31 Dec 2020		
Is this case study continued from a case study submitted in 2014? No		

1. Summary of the impact

GraphicsFuzz, a tool that automatically finds bugs in compilers for graphics processing units (GPUs), is based on fundamental advances in *metamorphic testing* pioneered by Donaldson's Multicore Programming research group at Imperial College London. GraphicsFuzz has found serious, security-critical defects in compilers from *all* major GPU vendors, leading to the spin-out of GraphicsFuzz Ltd., to commercialise the research. GraphicsFuzz Ltd. was acquired by Google in 2018 for an undisclosed sum, establishing a new London-based team at Google, led by Donaldson, to focus on Android graphics driver quality. Google have open-sourced the GraphicsFuzz tool, and it is now being used routinely to find serious defects in graphics drivers that affect the Android operating system (estimated 2.5 billion active devices worldwide) and the Chrome web browser (used by an estimated 66% of all internet users). GraphicsFuzz was enabled by two previous research projects from Donaldson's group that have also led to industrial impact: GPUVerify, a highly scalable method for static analysis of GPU software, which ARM have integrated into their Mali Graphics Debugger tool, and CLsmith, an automated compiler testing tool that has found bugs in OpenCL compilers across the industry. The global GPU market is predicted to grow to grow from USD20 billion in 2019 to USD200 billion in 2027, and the industrial uptake of these tools – all based on fundamental research from Imperial College London – is having major impact on the industry, making GPU software safer and more secure for end users, and reducing development costs of GPU software and drivers.

2. Underpinning research

Nature of the research insights and findings relating to the impact: Graphics processing units are massively parallel, making them well-suited to accelerating computationally-intensive workloads, such as the machine learning and computer vision tasks required for autonomous driving, as well as enabling high-fidelity computer graphics. GPU programs and device drivers *always* need to be highly optimized: speed of execution is the sole reason for using GPU hardware. It is fundamentally challenging to build highly-optimised software with a high degree of reliability, yet GPU technology must be reliable: errors can be catastrophic when GPUs are used in safety-critical domains (such as autonomous vehicles), and malfunctioning graphics drivers may render consumer devices useless. The work underpinning this case study was motivated by the exciting challenge of how to design tools to aid in building correct and efficient GPU software and compilers.

Donaldson's group have contributed to solving this challenge via (1) methods for reasoning about the correctness of programs designed for GPU acceleration, and (2) techniques for automatically testing GPU compilers to check that they faithfully translate high-level GPU programs into equivalent GPU-specific machine code. The main challenge associated with (1) is that GPU programs are typically executed by *thousands* of threads, requiring highly scalable analyses. The difficulty associated with (2) is that graphics programming languages are deliberately under-specified, so that no *oracle* is available to determine whether a given program has been correctly compiled.

Outline of underpinning research, and associated dates: The research at Imperial was conducted between November 2011 and March 2017, with entrepreneurship activity taking place between January 2017 and October 2017. The GPUVerify tool is based on a novel method for transforming a highly-parallel GPU program into a *sequential* program that models the execution of an arbitrary *pair* of threads such that data race-freedom (an important property of concurrent programs) for the parallel GPU program can be established by analysing the radically simpler sequential program [R1, R2]; this allows decades of work on sequential program verification to be leveraged. While GPUVerify can provide formal guarantees about the source code of a GPU program written in Open Computing Language (OpenCL), these guarantees are undermined if the downstream compiler that turns OpenCL into GPU-specific machine code is defective. This resulted in the CLsmith project for automated testing of OpenCL compilers (extending the Csmith project for testing of C compilers, from the University of Utah). CLsmith uses novel methods to generate OpenCL programs in a randomised fashion such that each generated program is guaranteed to compute some well-defined (albeit unknown) result [R3]. This approach enables *differential testing*: if two different OpenCL compilers yield *different* results on one of these well-defined programs, at least one of the compilers must exhibit a bug. Automated program reduction techniques can then be used to provide small OpenCL programs that expose the root cause of compiler bugs [R4]. Inspired by the success of CLsmith, GraphicsFuzz (originally called GLFuzz) is an automated testing tool for compilers for the OpenGL Shading Language (GLSL) [R5, R6]. The key innovation behind GraphicsFuzz is to apply *metamorphic testing* to compilers, based on the fact that we should expect two equivalent, deterministic programs to yield *equivalent*, albeit *a priori* unknown, results when compiled and executed. The tool takes an initial graphics program and uses program transformation techniques to yield many equivalent programs, looking for divergences in behaviour. Such divergences indicated compiler bugs. To shed light on the *root causes* of such bugs, transformations can be iteratively reversed, to search for pairs of equivalent programs that differ only slightly, but for which the compiler under test generates semantically inequivalent code.

3. References to the research

[R1] A. Betts, N. Chong, A.F. Donaldson, S. Qadeer, P. Thomson: GPUVerify: a verifier for GPU kernels. ACM OOPSLA 2012, pp. 113-132. Google Scholar cites: 165. DOI: 10.1145/2384616.2384625, Open access version

[R2] A. Betts, N. Chong, A.F. Donaldson, J. Ketema, S. Qadeer, P. Thomson, J. Wickerson: The Design and Implementation of a Verification Technique for GPU Kernels. *ACM Trans. Program. Lang. Syst.* 37(3): 10:1-10:49, 2015. Google Scholar cites: 41. DOI: 10.1145/2743017, Open access version

[R3] C. Lidbury, A. Lascu, N. Chong, A.F. Donaldson: Many-core compiler fuzzing. ACM PLDI 2015, pp. 65-76. Google Scholar cites: 92. DOI: 10.1145/2737924.2737986. Open access version

[R4] M. Pflanzer, A.F. Donaldson, A. Lascu: Automatic Test Case Reduction for OpenCL. ACM IWOCL 2016, pp. 1:1-1:12. Google Scholar cites: 11. DOI: 10.1145/2909437.2909439, Open access version

[R5] A.F. Donaldson, A. Lascu: Metamorphic testing for (graphics) compilers. ACM MET, pp. 44-47. Google Scholar cites: 30. DOI: 10.1145/2896971.2896978, Open access version

[R6] A.F. Donaldson, H. Evrard, A. Lascu, P. Thomson: Automated testing of graphics shader compilers. *Proc. ACM Programming Languages* 1 (OOPSLA): 93:1-93:29, 2017. Google Scholar cites: 40. DOI: 10.1145/3133917, Open access version

4. Details of the impact

Google's Director of Engineering for Android Gaming and Graphics stated: *"The cumulative effect of this impact on the Android and Chrome platforms is an improvement in graphics driver reliability and security for billions of users around the world. [...] Having something like GraphicsFuzz to aid in the discovery of vulnerabilities and issues can save companies like Google billions of dollars."* Savage also calls out the relationship between this impact and the underpinning research: *"The impact is a* ***direct result*** *of the first-class research undertaken at Imperial College London that lies behind the GraphicsFuzz approach."* [I1]. It is estimated that there are more than 2.5 billion active Android devices worldwide, and that 66% of all internet users use the Chrome browser.

Impact of the GraphicsFuzz tool. The original GLFuzz tool was highly successful in applying metamorphic testing to finding bugs in GLSL compilers from all seven major GPU providers [R5, R6]. Donaldson disseminated this to a wide audience by writing a series of blog posts describing these findings (AMD post, Apple post, ARM post, Imagination Technologies post, Intel post, NVIDIA post, Qualcomm post). These bugs included three serious security vulnerabilities for which the Imperial team received public recognition – an Apple iOS information leak [I2], an NVIDIA kernel mode vulnerability [I3], and data-capture exploit caused by a defect in an ARM graphics driver [I4]. Donaldson and team promoted GraphicsFuzz on social media, reaching the front page of HackerNews and being featured by sites such as Phoronix [I5].

Acquisition of GraphicsFuzz Ltd. by Google, and open sourcing. Encouraged by this media interest, Donaldson and his post-docs Hugues Evrard and Paul Thomson secured funding from the DCMS/Innovate UK ICURe programme to investigate commercialisation opportunities for the work, leading to them founding GraphicsFuzz Ltd., incorporated in December 2017 and spun-out from Imperial in February 2018 [I6]. The team promoted the technology via an innovative web-app that would run a set of GraphicsFuzz tests on a user's device, allowing them to tweet a summary of the test results [I7]. This raised the profile of GraphicsFuzz quickly, leading to the acquisition of the company by Google in August 2018 for an undisclosed sum [I6]. Google open-sourced the GraphicsFuzz tool in September 2018 [I8].

Ongoing impact of GraphicsFuzz on graphics driver quality. Donaldson, Evrard and Thomson joined Google UK, and now lead a team specialising in methods to improve the quality of Android graphics device drivers, deployed on billions of devices worldwide. GraphicsFuzz is being used to find defects in Android graphics drivers from ARM, Qualcomm, Imagination Technologies and NVIDIA, who collectively produce GPUs for more than 79% of the smartphone and tablet GPU market. Tests that expose these bugs are being integrated with the Android Compatibility Test Suite, meaning that devices can only ship future versions of Android if these bugs are fixed in their drivers. The impact of this ongoing use of GraphicsFuzz is detailed in the supporting letter from Google's Director of Engineering, Android Gaming and Graphics [I1].

Impact on Chrome. The WebGL technology brings graphics rendering capabilities to web applications but exposes widely-used browsers such as Chrome – used by most of the world's internet users – to graphics-related security vulnerabilities. GraphicsFuzz has been integrated with the ClusterFuzz project for continuous fuzz testing of Chrome, and since January 2019 has found more than a dozen security vulnerabilities in Chrome's WebGL implementation that had been missed by other testing methods. These vulnerabilities have now been fixed [I9].

Impact on open source graphics drivers. Google has used GraphicsFuzz to rigorously test AMD's open-source driver for Vulkan, finding a large number of driver bugs, which have been now fixed, including bugs that turned out to affect the widely-used LLVM compiler framework [I10]. GraphicsFuzz has also been used to find defects in the open-source Mesa drivers for Intel and NVIDIA GPUs [I11].

Tools for the Vulkan ecosystem. GraphicsFuzz has found more than 100 bugs collectively in a suite of tools that underpin the Vulkan graphics programming model: the glslang shader translator, the spirv-opt and spirv-val optimizer and validator that ship as part of the SPIRV-Tools framework, and the spirv-cross cross-compiler [I12].

Integration with Vulkan CTS. Every Vulkan driver must pass the Vulkan Conformance Test Suite (CTS), which now features a new **graphicsfuzz** test group, comprising tests generated by GraphicsFuzz that exposed bugs in previously-conformant Vulkan drivers. Adding these tests to the Vulkan CTS *requires* fixes to any defective drivers in order for them to *remain* conformant. The Vulkan CTS now features more than 300 GraphicsFuzz tests [I13].

Impact of GPUVerify on ARM. The collaboration between Imperial College London and ARM during the CARP EU project (coordinated by Donaldson) led to ARM integrating GPUVerify into their Mali Graphics Debugger [I14]. Leveraging fundamental results on how to scale analysis to thousands of parallel threads [R1, R2], this allows software developers targeting ARM's Mali GPU series – which leads the smartphone and tablet GPU market – to reason rigorously about the OpenCL programs that they write.

Impact of CLsmith on OpenCL compilers. The Imperial team who developed CLsmith initially used it to find more than 50 bugs in OpenCL compilers from companies including Altera, AMD, ARM, NVIDIA and Intel [R3, R4]. CLsmith has been used by Codeplay Software Ltd., a UK-based compiler company, for continuous integration of their OpenCL compilers; see letter of support from Codeplay's Principal Software Engineer, AI Parallelism [I15]. Via a TETRACOM technology transfer project, CLsmith has since been incorporated into the set of tools used by UK-based start-up company dividiti for analysis of many-core platforms [I16].

Beneficiaries of the impact: The immediate beneficiaries of GPUVerify, CLsmith and GraphicsFuzz are GPU technology developers, who are now able to construct more reliable software and drivers as a result of the impact of these tools. End users of computer systems (i.e. the general public) are ultimately beneficiaries due to more reliable applications that use computer graphics.

Nature of the impact: By finding bugs in GPU technology automatically, these tools (a) reduce the manual effort required for GPU quality assurance, (b) increase test rigour by finding defects that human testers would be unlikely to uncover, and (c) increase the coverage of graphics driver compatibility test suites.

5. Sources to corroborate the impact

[I1] See letter of support from the Director of Engineering, Graphics and Gaming, at Google.

[I2] Apple: "About the security content of Safari 10.1" (link archived here) March 2017. (Search for "GLFuzz" for confirmation of the claim that GraphicsFuzz (then called GLFuzz) found a security vulnerability affecting iOS. Filed as CVE-2017-2424 (link archived here).)

[I3] Nvidia: “Security Bulletin: NVIDIA GPU display driver contains multiple vulnerabilities in the kernel mode layer handler”, July 2017. (Search for “Alastair Donaldson” for confirmation of the claim that GraphicsFuzz found a security vulnerability affecting NVIDA graphics drivers. Filed as CVE-2017-6259.) Link archived here.

[I4] bugs.chromium.org: “Issue 675658: Security: Malicious WebGL page can capture and upload contents of other tabs”, December 2016. (Supports claim that GraphicsFuzz (then called GLFuzz) uncovered a security vulnerability affecting the Chrome web browser.) Link archived here.

[I5] phoronix.com: “Fuzzing OpenGL Shaders Can Lead To Some Wild Results”, December 2016. (Supports claim about press coverage of blog posts.) Link archived here.

[I6] Imperial College : “GraphicsFuzz Acquired by Google”, August 2018. (Supports claim that GraphicsFuzz Ltd. was an Imperial College spinout acquired by Google.) Link archived here.

[I7] Examples of users tweeting results from the GraphicsFuzz web app: 1, 2, 3, 4, 5, 6, 7, 8. (Supports claim of the app’s existence and the public’s engagement.) PDF of the Tweets available here.

[I8] GraphicsFuzz on GitHub: https://github.com/google/graphicsfuzz. (Supports claim that Google open-sourced the GraphicsFuzz tool.) Link archived here.

[I9] Security bugs found by GraphicsFuzz that affected Chrome and have now been fixed for sufficiently long that Google have made them public. Link archived here.

[I10] Impact on open-source AMD driver: example bugs in the widely-used LLVM framework that affected AMD’s driver and that have been fixed 1, 2, 3, 4; other bugs that affected AMD’s driver and have been fixed: 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17. PDF available here.

[I11] Impact on open-source Mesa drivers: example bugs in Intel Mesa driver: 1, 2, 3, 4; example bugs in Nouveau open source NVIDIA driver: 1, 2. PDF available here.

[I12] Impact on the Vulkan tooling ecosystem: bugs in SPIRV-Tools, bugs in SPIRV-Cross, bugs in glslang all found by GraphicsFuzz. Link archived here.

[I13] Impact on Vulkan Conformance Test Suite: GraphicsFuzz tests that have been added to the Vulkan CTS. Link archived here.

[I14] S. Barton: “Debugging OpenCL Applications with Mali Graphics Debugger V2.1 and GPUVerify”, April 2015. (Supports claim that GPUVerify has been integrated into the ARM Mali Graphics Debugger.) Link archived here.

[I15] See letter of support from the Principal Software Engineer, AI Parallelism, at Codeplay Software Ltd.

[I16] Exploitation of compiler testing work from Imperial College London is listed on the Success Stories page of the dividiti Ltd. website. Link archived here.