Synthesizing High-Performance Image Processing Applications with Hipacc

M. Akif Özkan, Oliver Reiche, Bo Qiao, Richard Membarth, Jürgen Teich, and Frank Hannig
Hardware/Software Co-Design, Department of Computer Science,
Friedrich-Alexander University Erlangen-Nürnberg (FAU), Germany.

***Abstract*—Programming heterogeneous platforms to achieve high performance is laborious since writing efficient code requires tuning at a low level with architecture-specific optimizations and is based on drastically differing programming models. Performance portability across different platforms can be achieved by decoupling the algorithm description from the target implementation. We present Hipacc (http://hipacc-lang.org), a framework consisting of an open-source image processing DSL and a compiler to target CPUs, GPUs, and FPGAs from the same program. We demonstrate Hipacc's productivity by considering real-world computer vision applications, e.g., optical flow, and generating target code (C++, OpenCL, C-based HLS) for three platforms (CPU and GPU in a laptop and an FPGA board). Finally, we showcase the real-time processing of images acquired by a USB camera on these platforms.**

Image processing underpins many of today's smart systems ranging from medical imaging up to advanced driver assistance systems. Yet, there is not any *best* computing platform that always meets the requirements of target systems regarding performance, energy efficiency, and power. Mobile devices like phones and tablets strive for efficient implementations to save battery live; driver assistance systems require image processing to be in time; and systems in medical imaging have to process extremely high data volumes fast. A dedicated microcontroller, a central processing unit (CPU) or graphics processing unit (GPU) embedded in a system-on-a-chip (SoC), or a field programmable gate array (FPGA) might be possible target architectures. Yet, implementation code needs to be rewritten in a programming language that is suitable for a selected target. Furthermore, high performance can mostly be achieved only with platform-specific optimizations, which make the code lengthy and hardly reusable or even not portable. This is a time-consuming task even for platform experts, yet almost insurmountable for algorithm developers who are often mainly interested in mathematical models.

As a solution, the Heterogeneous Image Processing Acceleration (Hipacc) framework [1], [2] decouples the algorithm description from low-level implementation details utilizing a domain-specific language (DSL), thus enables highly optimized and efficient target code generation via source-to-source compilation. Initially developed to target GPUs from Nvidia and AMD, Hipacc was subject to multiple extensions over the years. These extensions involve code generation for other accelerators, such as embedded GPUs [3] and FPGA devices [2], [4] through high-level synthesis for Xilinx and Intel/Altera FPGAs. Figure 1 provides a visual overview of the framework and its target architectures. Our approach allows compact algorithm descriptions (i.e., a high productivity), portability between different target platforms, as well as excellent execution speed (performance) compared to state-of-the-art frameworks.

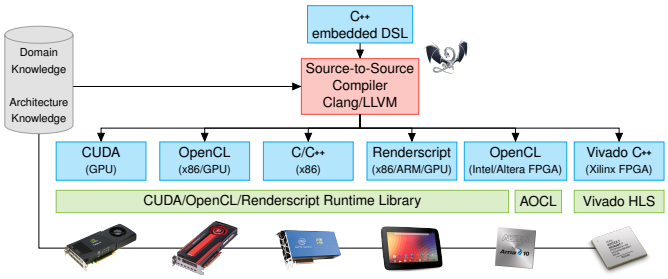

Figure 1. Overview of the Hipacc framework and its target architectures.

In this demonstrator, we present Hipacc's ease of image processing system design. We provide DSL application codes for various image processing algorithms, including Harris corner detection and optical flow, from which code is generated for any of the selected target platforms. Thereby, we target the CPU and GPU in the demonstrator laptop as well as an FPGA that is connected via ethernet. Additionally, we show that other C++ libraries, such as the socket library that is used for the Ethernet connection, can be orchestrated with Hipacc as a benefit of having a DSL embedded in C++.

References

[1] R. Membarth, O. Reiche, F. Hannig, J. Teich, M. Körner, and W. Eckert, "HIPAcc: A domain-specific language and compiler for image processing", *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 210–224, Jan. 1, 2016. DOI: 10.1109/TPDS.2015.2394802.

[2] O. Reiche, M. A. Özkan, R. Membarth, J. Teich, and F. Hannig, "Generating FPGA-based image processing accelerators with Hipacc", in *Proceedings of the International Conference On Computer Aided Design (ICCAD)*, (Irvine, CA, USA), IEEE, Nov. 13–16, 2017, pp. 1026–1033. DOI: 10.1109/ICCAD.2017.8203894.

[3] R. Membarth, O. Reiche, F. Hannig, and J. Teich, "Code generation for embedded heterogeneous architectures on Android", in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, (Dresden, Germany), European Design and Automation Association (EDAA), Mar. 24–28, 2014, 86:1–86:6. DOI: 10.7873/DATE.2014.099.

[4] M. A. Özkan, O. Reiche, F. Hannig, and J. Teich, "FPGA-based accelerator design from a domain-specific language", in *Proceedings of the 26th International Conference on Field-Programmable Logic and Applications (FPL)*, (Lausanne, Switzerland), IEEE, Aug. 29–Sep. 2, 2016, 9 pp. DOI: 10.1109/FPL.2016.7577357.