TULIPP: Towards Ubiquitous Low-power Image Processing Platforms

Tobias Kalb, Lester Kalms, Diana Göhringer
Ruhr-University Bochum, Germany
{tobias.kalb, lester.kalms, diana.goehringer}@rub.de

Ananya Muddukrishna, Magnus Jahre, Per Gunnar
Kjeldsberg
Norwegian University of Science and Technology, Norway
{ananya.muddukrishna, magnus.jahre}@idi.ntnu.no,
pgk@ntnu.no

Carl Ehrenstråhle, Magnus Peterson
Synective Labs AB, Sweden
{carl.ehrenstrahle, magnus.peterson}@synective.se

Antonio Paolillo, Christian Lemer, Ben Rodriguez
HIPPEROS S.A., Belgium
{antonio.paolillo, christian.lemer,
ben.rodriguez}@hipperos.com

Carlota Pons, Fabien Marty
Efficient Innovation, France
{c.pons, f.marty}@efficient-innovation.com

Boitumelo Ruf, Tobias Schuchert, Igor Tchouchenkov
Fraunhofer Institute of Optronics, System Technologies and
Image Exploitation IOSB, Germany
{boitumelo.ruf, tobias.schuchert,
igor.tchouchenkov}@iosb.fraunhofer.de

Flemming Christensen
Sundance Multiprocessor Technology Ltd., UK
flemming.c@sundance.com

Guillaume Bernard, François Duhem, Philippe Millet
Thales TED-MIS, Thales TRT, France
{guillaume.bernard, francois.duhem,
philippe.millet}@thalesgroup.com

Abstract—**Many industrial domains rely on vision-based applications which require to comply with severe performance and embedded requirements. TULIPP will develop a reference platform, which consists of a hardware system, a tool chain and a real-time operating system. This platform defines implementation rules and interfaces to tackle power consumption issues while delivering high, energy efficient and guaranteed computing performance for image processing applications. Using this reference platform will enable designers to develop a complete solution at a reduced cost to meet the typical embedded systems requirements: Size, Weight and Power. Moreover, for less constrained systems which performance requirements cannot be fulfilled by one instance of the platform, the reference platform will also be scalable so that the resulting boards can be chained for higher processing power. The instance of the reference platform developed during the project will be use-case driven and split between the implementation of: a reference hardware architecture - a scalable low-power board; a low-power operating system and image processing libraries; a productivity-enhancing tool chain. It will lead to three proof-of-concept demonstrators across different application domains: real-time and low-power medical image processing product prototype of surgical X-ray system (mobile c-arm); embedded image processing systems within Unmanned Aerial Vehicles (UAVs); automotive real time embedded systems for driver assistance. TULIPP will set up an ecosystem and will closely work with standardization organizations to propose new standards derived from its reference platform to the industry.**

Keywords—Embedded Image Processing, Low-Power Image Processing; Hardware Platform; Multicore; Real Time Operating System; Heterogeneous Architecture; FPGA; Multi-Processor System-on-Chip; MPSoC

I. INTRODUCTION

A lot of modern industrial systems rely on vision-based applications. These systems have to comply with severe performance and embedded requirements. Good examples are medical imaging, advanced automotive systems and Unmanned Aerial Vehicles (UAVs). Such vision-based applications are based on two main building blocks: image processing and image display. Sensors are then increasingly numerous, while simultaneously having to cope with a growth in smartness and data rate. Thus the complexity of image processing applications is continuously growing. Moreover, this technological challenge is also strongly resource bounded both in terms of autonomy and energy consumption. Needs from end user, customer and prospective business and technological innovation drive such a growing trend. However, in most cases non-optimized consumer electronics cannot satisfy this demand for low power with and sufficient performance.

There are two main approaches to reach the performance requirements of current algorithms working on high definition image frames. The first approach is to run these algorithms on dedicated high-performance computers. The second approach is an emerging trend based on General Purpose Graphics Processing Units (GPGPUs) utilizing their inherent parallel architecture to reduce execution time from days to hours or less. Although processing time can be reduced, the power consumption still is too high to make these approaches feasible for embedded computing.

A promising trend is the usage of low-power Graphics Processing Units (GPUs) or even more power efficient Field Programmable Gate Arrays (FPGAs). This guides computing

towards a low-power computing continuum that spans - among others - embedded systems, mobile devices, desktops and data centers. The efficiency of current architectures is not sufficient for the demands required by modern low-power computing systems. Embedded systems are often battery powered and hence performance and energy efficiency means longer battery life and better user experience.

In order to achieve efficiency, heterogeneity is the key. Each type of processing element has a type of processing for which it performs best [1]. To get the best out of an efficient heterogeneous hardware platform, it is important to efficiently map the application onto the different processing elements and manage its execution at runtime. The increasing demand on vision-based systems also asks for reducing time-to-market, development and rework costs on a product as well as maximizing reuse of designs. However, there is still no standard for high-performance embedded computing systems on heterogeneous platforms in the domain of vision-based systems.

TULIPP aims to push forward a reference platform defining implementation rules to provide designers with guaranteed high-performance solutions for vision-based systems while reducing development time and costs. To validate these design rules a heterogeneous hardware reference platform, a real-time operating system and a productivity-enhancing toolchain will be developed during the project. These three components will enable high-performance image processing for modern low-power embedded systems.

The remainder of the paper is structured as follows: In Section II an overview of related work is provided. Section III describes the components of the hardware, operating system and toolchain, which will be developed in the TULIPP project. Section IV follows with a description of the use cases used for the evaluation of the developed components, and finally Section V concludes the paper with a summary and an outlook.

II. RELATED WORK

TULIPP aims to develop three main components, namely a hardware platform, a productivity-enhancing toolchain and a real-time operating system. These developments are targeted towards low-power yet high-performance image processing on embedded devices.

A. Hardware Architecture

The majority of all current image/medical embedded solutions is built on either single-core CPUs or shared memory multi-core CPUs. The term homogeneous computing refers to 100% identical CPUs running at the same speed and sharing an amount of memory. The input/output controller is the same type of CPU that does the processing and is wasteful but easy to program. For high-end automotive vision systems, heterogeneous computing is used, but each new technology generation requires huge implementation.

TULIPP will introduce the use of heterogeneous computing in image processing applications and medical solutions by incorporating different processing elements. An example might be a small 32-bit CPU for controlling in- and output, and

multiple 64-bit CPUs with additional accelerations by FPGA fabrics for processing images. The TULIPP project will produce the tools to make it easier to program and the operating system to manage the heterogeneous computing system in an energy efficient manner. The reference platform, tools and guidelines will enable medical imaging and unmanned aerial vehicles as well as automotive vision systems to overcome the ever evolving technologies.

B. Operating System and Low-Level Libraries

Regarding the RTOS part of the project, the state of the art shows a very significant gap between research and actual implementation in commercial systems. There is a variety of RTOS on the market, but most of them have been designed more than 20 years ago and do not incorporate recent scientific innovations [2]. These innovations are mainly in scheduling algorithms, resource sharing algorithms and Inter-Process Communication (IPC). With few exceptions, existing RTOS do not support power-aware scheduling at the kernel level [3]. In addition, existing RTOS have been designed with single core platforms in mind. They do not incorporate efficient support for parallel platforms. Moreover, with regard to parallel platforms, most RTOS are ported on SMP machines, meaning parallel processing units that are identical, and not on heterogeneous MPSoCs. So far, hardware and software of a real-time system were not designed together. For instance, heat dissipation thermal issues awareness at the level of the RTOS scheduler is also quasi non-existent. The consequence is that the lifetime of the system is reduced and reliability is compromised [2].

The solution proposed by the TULIPP project is a new kernel architecture specifically designed for heterogeneous multi-cores and under constraints such as small footprint, low power and good scalability. This is a combination of several features. One will be to use a master-slave micro-kernel design specifically built for better scalability. This reduces locks and allows for a small footprint [4][5]. Second, power-aware schedulers will be used (Earliest Deadline First (EDF) instead of Rate Monotonic (RM) [6]). This creates an optimal schedule using as little power as possible. Efficient and scalable mechanisms will be used for resource sharing and for IPC. The main advantage of the solution is its efficiency for parallelism and power optimization.

C. Development Tools

The current state of the art in development tools for highly heterogeneous image processing systems is that application programmers have to manually interact with vendor specific tools of components. These tools are complex and a significant amount of time needs to be invested to master them, resulting in low productivity and reduced innovation rate. TULIPP aims to remedy the problem by providing toolchain utilities that enable programmers to use multi-vendor tools productively.

Extensive reviews of heterogeneous system development tools exist [7][8][9]. We briefly review development tools that are relevant to TULIPP.

The Xilinx SDSoC development environment eases the programming of heterogeneous SoCs consisting of an ARM

processor and a Xilinx FPGA [10]. SDSoC is sold as an easy-to-use HLS tool that enables programmers to accelerate functions within C/C++ applications on FPGA devices. SDSoC is bound to systems provided by Xilinx. Nsight is a debugger and profiler for CUDA and OpenCL-based applications running on systems with NVIDIA GPUs [11]. It can debug and visualize traces of concurrent CPU-GPU execution. This enables programmers to spot performance bottlenecks and bugs more quickly. Visual Profiler is a recent performance analysis tool from NVIDIA for CUDA C/C++ programs running on systems with NVIDIA GPUs [12]. Visual Profiler can debug, profile and visually depict concurrent CPU-GPU execution similar to Nsight. In addition, it can automatically identify bottlenecks, suggest optimizations, and profile energy and clock activity. VTune Amplifier is a performance analysis and optimization from Intel [13]. VTune Amplifier supports systems with Intel processors and has limited support for ARM processors. It can automatically profile and detect problems involving CPU utilization, lock contention, memory bandwidth and cache performance. Detected problems are visualized on timelines and traced back to source code to help programmers solve problems faster.

The solution proposed by the TULIPP project allows mapping an application onto a heterogeneous, TULIPP-compliant platform, executing and profiling its performance and power efficiency and finally analyzing the results, which are used to support the developer for further optimizations.

III. CONCEPTS OF THE TULIPP COMPONENTS

The main value proposition of the TULIPP project is to develop a reference platform that defines implementation rules and interfaces to tackle power consumption for high and efficient computing performance demands for image processing applications. The associated objective is to counteract the never-ending changing process of image processing boards due to the continuous evolution of components. The overall design cost of image processing devices will then drastically be reduced by developing and integrating universal interfaces on the platform so that any new generation of components may be integrated without significant additional design costs. It will also allow high performance computing systems to be more embedded and less power consuming.

The image processing market for embedded technologies targeted by the TULIPP solution demands highest performance and mechanical flexibility to be able to comply with heat dissipation and size constraints, low cost and low power consumption. As the TULIPP concept is based on an optimized system enabling customization, its activities will first focus on defining how to use the technology to best design such a board, taking into consideration that its reference platform will evolve at the same pace than the technology to benefit as well from the improvement of future chips.

Therefore, rather than developing a generic platform solution, TULIPP proposes to capitalize on its efforts to go a step further and develop a reference platform that defines implementation rules and interfaces to tackle power consumption delivering high and efficient computing

performance for image processing applications with guaranteed real-time features and latency. TULIPP will build a reference platform through industrial consensus dedicated to low-power real-time image processing applications (Fig. 1). The project will concentrate on interfaces between the components of the platform (hardware, toolchain, operating system) as well as design and implementation rules. Following the rules, a developer will be able to produce a compliant platform and benefit from the technological advances generated by the project. The interfaces will be defined, to enable any industrial to produce any sub-part of the platform and plug it to other existing platforms to create a new compliant TULIPP platform instance. Therefore, the TULIPP project will set up and work closely with an ecosystem. Thus, valuable feedback is used during the project.
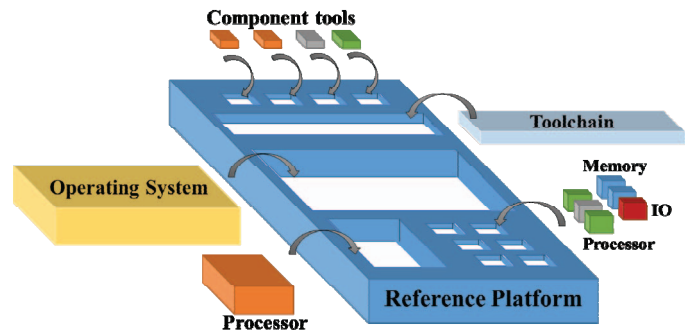

Fig. 1. TULIPP Reference Platform

A. Hardware Architecture

The hardware architecture will be a template of the heterogeneous computing system. Such a trend can currently be noticed from chip manufacturers producing heterogeneous systems for embedded applications. NVIDIA, for instance, delivers high performance with a cluster of ARM processors and a GPGPU with the Tegra-K1 [14]. Xilinx builds similar SoC solutions with the Zynq and UltraScale+ MPSoC devices [15]. While one of such chips will be efficient at processing a specific part of an application, another might be better for a different part of the same application. The TULIPP hardware platform template will allow combining a variable number of chips and different types of chips on a single board. However, having such a heterogeneous system brings a drawback: when an application only needs one chip at a given time, then the unused chips will still consume power. The TULIPP platform will enable chips or communication infrastructures to be switched off when unused. Thus, for each part of an application, the best suitable chip will be selected while the other parts of the platform will be switched off.

The TULIPP project will define how to select SoCs suitable to build a TULIPP platform instance, how to interconnect several SoCs and how to manage the selection and switch-off mechanisms at run-time. Thus, it will enable the reduction of the power consumption of the whole system. The hardware platform will be fine-tuned and configured for each application. The operating frequencies will also be adjustable when possible; when implementing the application on an FPGA, the frequency will be adjusted to the lowest value that fits the

needs of the application. Further features for low power consumption and adaptivity will be introduced by the utilization of dynamic partial reconfiguration (DPR) [16].

B. Operating System and Low-Level Libraries

The operating system and low-level libraries designed for low power and image processing will have to match the application requirements. It will run on the instantiated processors of the hardware layer, which means communication and synchronization mechanisms have to be implemented between the processors. The footprint – i.e. the binary size – of the operating system must be small (a few tens of kilobytes) because each hardware component embeds only a small local memory. Bigger memories like DDR will also be available but the access time will not be compatible with running application in real-time. Some of the targeted components that will be implemented on the TULIPP reference platform will come with some primitives to be integrated or wrapped in the library available to programmer. Standard APIs will be extended or shortened to cope with low power and image processing. Extensions or modified standards will be proposed as pre-norm. The real-time operating system will be designed by HIPPEROS and based upon the HIPPEROS family of RTOSs [5][17]. The role of the RTOS as a TULIPP component will be to efficiently handle hardware resources (multi-core CPUs and FPGA circuitry) and to provide real-time guarantees and interfaces easy to program to the reference platform user. The operating system is based on a reliable micro-kernel architecture that provides hard real-time scheduling of hardware/software tasks, virtual memory management to isolate processes and efficient IPC mechanisms. The operating system interfaces will be built to easily integrate other TULIPP components (hardware and tool chain). Standard tools will be supported to interface with hardware (boot-loaders, debuggers, etc.). APIs well suited for the embedded and image processing domain (such as POSIX, OpenMP, OpenCV, etc.) will be supported by low-level libraries shipped with the operating system.

C. Toolchain

A TULIPP compliant platform (hereafter called TULIPP platform) can have components from different vendors, each with their own specific toolchains and Integrated Development Environments (IDEs). Expertise in multi-vendor tools is therefore required to develop high-performance, low-power applications efficiently on TULIPP platforms. Such expertise takes a long time to develop, is beyond the abilities of average programmers and narrow experts, and inhibits productivity. Lack of expertise in a particular vendor tool may also prevent platform builders from choosing an otherwise fitting hardware component.

The toolchain component of the TULIPP platform is a set of utilities that enhance programmer productivity while using multi-vendor tools. The set of utilities is called STHEM - Supporting uTilities for Heterogeneous EMbedded image processing. STHEM wraps around, extends and connects existing vendor tools to present a seamless mapping and performance/energy analysis interface to programmers. Using STHEM interfaces, programmers can easily map application parts to suitable components, employ useful primitives and library routines to control and communicate between the components, analyze performance and energy consumption of the application at whole platform and component granularities, and identify optimization opportunities and problem areas within the application.

A simplified view of the intended workflow for STHEM is shown in Fig. 2. The programmer assumes the role of a director in all stages of the workflow except the first. Directions given by the programmer implemented in the background by expert-written mechanisms. The workflow is iterated until desired performance and energy profile is reached. In the first stage of the workflow, programmers write application code. In addition to codifying application algorithms, optimizations understood in the previous iteration of the workflow are made at this stage. Programmers are also assisted with platform-specific primitives and library routines that abstract away commonly used domain-specific functionality. In the second stage, programmers direct the mapping of application to components using easy-to-use, expert-written mapping directives. Programmers can also refine existing mapping directives and create new ones. After mapping, programmers proceed to the third stage where they direct platform configuration, application execution, and profiling. Sensible default configurations are provided to minimize programmer effort in setting up the platform for execution and profiling. The fourth stage is the analysis stage. Profiling data collected in the previous stage is analyzed for problems and visually highlighted to programmers on Roofline plots, Grain graphs [18], thread time-lines, and call graphs. For the highlighted problems, programmers are also suggested general optimization strategies, which they can customize for the application by going back to the first stage. STHEM is built as an Eclipse 4 RCP plugin to facilitate an integrated workflow with popular vendor tools that integrate into the Eclipse IDE [10][11][13].
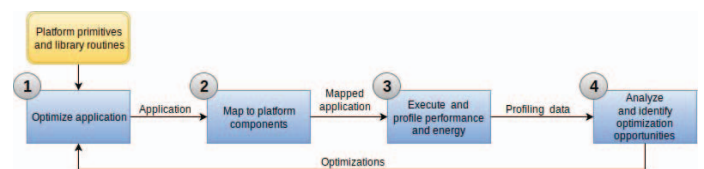

Fig. 2. High-level overview of the iterative workflow using STHEM

With the three components described above the TULIPP project aims to provide rules and definitions for a reference hardware platform, which enables a developer to design a high performance yet low-power image processing application. Furthermore, a real-time operating system is used, which guarantees a defined runtime behavior and significantly improves low-power features of the combination of hardware platform and software application. In addition, the development process is supported by a toolchain that provides valuable assistance for developing and deploying the application on a heterogeneous hardware design. Thus, the TULIPP project provides an extensible set of components for embedded high performance low-power image processing applications.

IV. TULIPP USE CASES

For the evaluation of the components developed by the TULIPP project three different use case scenarios are defined. The use cases are described by applications for medical imaging, for automotive and for UAVs. Even though the applications are different, the constraints of such embedded systems are similar: performance, power consumption, size or volume and costs. In addition to those constraints, for most of the applications the image is used at real-time and often with short or limited latency.

A. Medical X-Ray Imaging

In medical imaging, associated challenges to manage large volumes of data hinder the growth of this market. However, mobile imaging equipment is expected to replace high-end infrastructure devices, which will help the medical imaging market to grow. Modern day surgery requires that the surgeon has precise control of their movements and at times is able to see the path that blood flows though veins and arteries. This can only be seen with the use of complex imaging systems. Current X-Ray sensors, for example, are more like digital cameras than the plate and films of old systems, and as such can provide live images and video in real time. One problem with such technology is that when the radiation levels are set low to reduce the dose involved, the digital sensors are very sensitive to noise. Increasing the radiation level dose does reduce the level of noise, but can have serious adverse effects on both the patient and the surgeon. Keeping the radiation dose at safe levels and at the same time producing a clear live real time image requires significant processing power.

In the TULIPP project our aim is to try and reduce the level of radiation by 75%. As a result of this, more powerful image processing will be required in order to still be able to see small details in the human body that are crucial during surgery. Since most operating rooms are small, the device needs to be small and mobile. A system that integrates the processing close to the sensor would be ideal to help reduce extraneous wires. The system needs to be compact but also have a low power draw since heat and other RF emissions could disturb the sensors and eventually actually add more noise to the signal. The system must comply with hard real-time constraints as part of regulatory constraints regarding devices used in medical environments. This combination of requirements makes this use case a challenge to design and develop a matching solution.

B. Advanced Driver Assistance

In the automotive domain, the technology available today in high-end cars (collision avoidance, danger alert, pedestrian recognition or drowsy driver detection) will be integrated in middle range cars within the next ten years. More and more electronic devices are then going to be integrated in cars. Advanced Driver Assistance Systems, ADAS, are currently one of the most promising segments for image processing with a steep expected growth rate for the next five to ten years. Driving safety, in parallel with pedestrian safety, has a large focus in the automotive sector, with vision-based systems as one of the enablers for many new and innovative solutions. Also automotive safety organizations, like Euro NCAP, are updating their standards and will in a step-wise fashion over the next years require more and more active safety systems in the cars. Some of the most interesting application areas include Vehicle, pedestrian and object detection, Traffic sign recognition, Lane detection, Night Vision, Surround view and Driver monitoring. The data gathered from the systems very often are used in combination with data from other systems to either guide or assist the driver, or to take control of the vehicle by automatic braking, automatic lane keeping, park assist etc. These applications will over the years be refined and enhanced, resulting in fully autonomous driving solutions some ten years from now.

ADAS vision systems require real time, low-latency processing, at high to very high computational load. They need to be robust and reliable, and will often be treated as safety critical systems. The TULIPP project addresses all these questions. By offering a toolset and standardization, it will help the designers to focus on the image processing application rather than platform details. The TULIPP ADAS use case shows how a typical automotive vision application, pedestrian detection, can be facilitated by the TULIPP platform and how characteristics like low power, high performance and robustness are natively supported.

C. Autonomous UAVs

In the domain of UAVs and robotic vision, onboard image processing in real-time is one of the key technologies for autonomous operation [19]. Small UAVs have entered a large range of applications as their underlying technology has improved and more possibilities for use have been explored. Now, applications such as surveillance, search and rescue, video production, logistics and research are just a small subset of their uses [20]. Their operation in the entertainment domain is rapidly growing as the result versus cost ratio becomes more competitive. However, with the growing number of UAVs in use the number of crashes and problems with their control are also increasing. These problems can be caused by operator error or malfunction. In the worst-case scenario these errors can cause damage to more than just the UAV involved and end up harming people, goods or infrastructure [21]. Therefore, UAVs need more intelligent control and interaction systems, such as automatic collision avoidance or more robust pose estimation, to minimize risks of failure. The problem is that more intelligence needs more computing power, which is very limited especially on small UAVs.

The TULIPP solution aims to fill this processing gap by using its good performance-to-weight and power-consumption-to-weight figures. Similar to [22] we aim to use computer vision algorithms such as stereo and depth estimation to detect obstacles and evaluate the surroundings in order to make the UAV more intelligent. For this purpose, we attach the TULIPP-board with a stereo camera setup orientated in direction of flight to a UAV. Our goal is to use stereo algorithms to detect obstacles automatically that are within dangerous vicinity in front of the UAV and to avoid a collision. Amongst others [23], [24] and [25] describe popular stereo algorithms that are able to be run in real-time.

These three use cases represent an ideal combination of applications for the TULIPP project. Each use case requires

embedded high performance low-power image processing, but the constraints differ for each use case scenario. Striving for optimization for each use case, the developments of the TULIPP project will then provide a flexible and extensible solution including rules and definitions for the hardware platform, the operating system and the tool chain used to design, develop and deploy an embedded high performance low-power image processing application.

V. CONCLUSION

The TULIPP project is going to leverage the utilization of heterogeneous embedded computing platforms for image processing applications. This will be accomplished by setting up rules and definitions for hardware platforms and real-time operating systems. Thus, the TULIPP project aims to pave the way for standards for embedded high performance low-power image processing in industrial applications. The design, development and deployment of the hardware platform and real-time operating system will be improved by a productivity-enhancing toolchain. Three use cases, covering medical, automotive and UAV applications, will be used to validate the developed solutions provided by the project. The TULIPP project is also going to establish a valuable advisory board and ecosystem to further leverage and promote its developments towards industrial usage and standardization. The state and progress of the project will be available to the public at the website of the TULIPP project [26].

REFERENCES

[1] Göhringer, D., Birk, M., Dasse-Tiyo, Y., Ruiter, N., Hübner, M., Becker, J., "Reconfigurable MPSoC versus GPU: Performance, power and energy evaluation", 9th IEEE International Conference on Industrial Informatics (INDIN), 2011

[2] Brandenburg, B., "Scheduling and locking in multiprocessor real-time operating systems", Ph.D. dissertation, The University of North Carolina, 2011

[3] Mentor Graphics, "Nucleus RTOS", https://www.mentor.com/embeddedsoftware/nucleus/, Accessed on 23.05.2016

[4] Cerqueira, F., Vanga, M., Brandenburg, B., "Scaling Global Scheduling with Message Passing", Proceedings of the 20th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2014), pp. 263-274, April 2014

[5] Paolillo, A., Desenfans, O., Svoboda, V., Goossens, J., Rodriguez, B., "A new configurable and parallel embedded real-time micro-kernel for multi-core platforms", In Proceedings of the ECRTS Workshop on Operating Systems Platforms for Embedded Real-Time applications (ECRTS-OSPERT '15), July 2015

[6] Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., Weglarz, J., "Scheduling Computer and Manufacturing Processes", Springer (Berlin), 2001, ISBN 3-540-41931-4

[7] Jeffers, J., Reinders, J., "High Performance Parallelism Pearls Volume Two: Multicore and Many-core Programming Approaches", Morgan Kaufmann, 2015.

[8] Mittal, S., Vetter, J. S., "A survey of CPU-GPU heterogeneous computing techniques", ACM Computing Surveys (CSUR), vol. 47, no. 4, p. 69, 2015

[9] Cong, J., Liu, B., Neuendorffer, S., Noguera, J., Vissers, K., Zhang, Z., "High-level synthesis for FPGAs: From prototyping to deployment", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 30, no. 4, pp. 473–491, 2011

[10] Xilinx Inc., "SDSoC Environment User Guide (UG1027)", http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_4/ug1027-sdsoc-user-guide.pdf, Accessed on 14.03.2016

[11] NVIDIA Nsight | NVIDIA, http://www.nvidia.com/object/nsight.html, Accessed on 13.05.2016

[12] NVIDIA Visual Profiler NVIDIA Developer, https://developer.nvidia.com/nvidia-visual-profiler, Accessed on 07.04.2016

[13] Profiling OpenMP* applications with Intel® VTune™ Amplifier XE | Intel® Developer Zone, https://software.intel.com/en-us/articles/profiling-openmp-applications-with-intel-vtune-amplifier-xe, Accessed on 27.10.2015

[14] NVIDIA Corporation, "Whitepaper: NVIDIA Tegra K1 - A New Era in Mobile Computing", https://www.nvidia.com/content/PDF/tegra_white_papers/tegra-K1-whitepaper.pdf, January 2014, Accessed on 20.04.2016

[15] Xilinx Inc., "White Paper: Zynq UltraScale+ MPSoCs - Unleash the Unparalleled Power and Flexibility of Zynq UltraScale+ MPSoCs (WP470)", http://www.xilinx.com/support/documentation/white_papers/wp470-ultrascale-plus-power-flexibility.pdf, November 2015, Accessed on 20.04.2016

[16] Xilinx Inc., "White Paper: Partial Reconfiguration of Xilinx FPGAs Using ISE Design Suite (WP374)", http://www.xilinx.com/support/documentation/white_papers/wp374_Partial_Reconfig_Xilinx_FPGAs.pdf, May 2012, Accessed on 31.05.2016

[17] HIPPEROS S.A., http://www.hipperos.com, Accessed on 26.04.2016

[18] Muddukrishna, A., Jonsson, P. A., Podobas, A., Brorsson, M., "Grain Graphs: OpenMP Performance Analysis Made Easy", Proceedings of the 21st ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, New York, NY, USA, 2016, pp. 28:1–28:13

[19] Sung, C.-K. and Segor, F., "Onboard pattern recognition for autonomous UAV landing", Proc. SPIE 8499, Applications of Digital Image Processing XXXV, 84991K, 2012

[20] Kuntze, H. B. et al., "SENEKA - sensor network with mobile robots for disaster management", Homeland Security (HST), 2012 IEEE Conference on Technologies for, Waltham, MA, 2012, pp. 406-410

[21] Tchouchenkov, I., Segor, F., Schoenbein, R., Kollmann, M., Bierhoff, T., Herbold, M., "Detection And Protection Against Unwanted Small UAVs", Proceedings of the Eleventh International Conference on Systems ICONS, 2016

[22] Barry, Andrew J., Tedrake, R., "Pushbroom stereo for high-speed navigation in cluttered environments", IEEE International Conference on Robotics and Automation (ICRA), 2015

[23] Hirschmüller, H., "Accurate and efficient stereo processing by semi-global matching and mutual information", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2., 2005

[24] Hrabar, S. et al., "Combined optic-flow and stereo-based navigation of urban canyons for a UAV", 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2005

[25] Yang, R., Pollefeys, M., "Multi-resolution real-time stereo on commodity graphics hardware", Proceedings. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1. IEEE, 2003

[26] Tulipp, "Tulipp: Towards Ubiquitous Low-power Image Processing Platforms – High, efficient and guaranteed computing performance for image processing applications", http://www.tulipp.eu, Accessed on 31.05.2016