Chapter 10
Developing Low-Power Image Processing Applications with the TULIPP Reference Platform Instance

Tobias Kalb, Lester Kalms, Diana Göhringer, Carlota Pons, Ananya Muddukrishna, Magnus Jahre, Boitumelo Ruf, Tobias Schuchert, Igor Tchouchenkov, Carl Ehrenstråhle, Magnus Peterson, Flemming Christensen, Antonio Paolillo, Ben Rodriguez and Philippe Millet

10.1 Introduction

In today's industry, an increasing amount of applications relies on vision-based techniques. The resulting image processing systems cover a wide field of application, examples being medical imaging, automotive advanced driver assistance systems (ADAS) and unmanned aerial vehicles (UAVs). All these applications demand high computing performance, yet strict requirements and constraints of an embedded system have to be met. Therefore, both the embedded system and the image processing

T. Kalb (✉)
Ruhr-University Bochum, Bochum, Germany
e-mail: tobias.kalb@rub.de

L. Kalms · D. Göhringer
TU Dresden, Dresden, Germany
e-mail: lester.kalms@tu-dresden.de

D. Göhringer
e-mail: diana.goehringer@tu-dresden.de

C. Pons
Efficient Innovation, Castelnau-le-Lez, France
e-mail: c.pons@efficient-innovation.com

A. Muddukrishna · M. Jahre
Norwegian University of Science and Technology, Trondheim, Norway
e-mail: ananya.muddukrishna@idi.ntnu.no

M. Jahre
e-mail: magnus.jahre@idi.ntnu.no

B. Ruf · T. Schuchert · I. Tchouchenkov
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Karlsruhe, Germany
e-mail: boitumelo.ruf@iosb.fraunhofer.de

T. Schuchert
e-mail: tobias.schuchert@iosb.fraunhofer.de

C. Kachris et al. (eds.), *Hardware Accelerators in Data Centers*,
https://doi.org/10.1007/978-3-319-92792-3_10

application have to be optimized. Embedded image processing systems are expected to be a ubiquitous part of our society with a direct connection to cloud computing, servers and data centres. Today, the complexity of systems is continuously growing. The design and implementation of an image processing platform will be an even more challenging task for developers. A non-optimized design cannot satisfy the demand for low power and high performance.

Embedded platforms using a single processor do not offer enough processing power to run modern image processing applications. On the contrary, the number of sensors is growing and the system simultaneously has to deal with increasing connectivity and data, which puts even higher requirements on the processing of data and communication. The demands of modern and future low-power computing systems are not met by current architectures. Embedded vision systems are often battery powered. Therefore, an efficient architecture is needed so that performance and energy efficiency results in longer battery life and better user experience. Resource-intensive image processing can be offloaded to cloud computing or data centres, but this compromises real-time constraints and latencies of the application. Thus, the embedded system has to deliver low-power yet high-performance computing for image processing.

Regarding embedded platforms, there are two approaches to reach the performance requirements of current image processing algorithms. The first approach is to run the image processing applications on low-power graphics processing units (GPUs). This can significantly improve processing power. The second approach offers the possibility to further reduce energy consumption by using field programmable gate arrays (FPGAs) for image processing. Using state-of-the-art technology, both GPU and FPGA are connected to an embedded processor forming a system-on-chip (SoC). Heterogeneous architectures offer promising features for

I. Tchouchenkov
e-mail: igor.tchouchenkov@iosb.fraunhofer.de

C. Ehrenstråhle · M. Peterson
Synective Labs AB, Göteborg, Sweden
e-mail: carl.ehrenstrahle@synective.se

M. Peterson
e-mail: magnus.peterson@synective.se

F. Christensen
Sundance Multiprocessor Technology Ltd., Chesham, UK
e-mail: flemming.c@sundance.com

A. Paolillo · B. Rodriguez
HIPPEROS S.A., Ottignies-Louvain-la-Neuve, Belgium
e-mail: antonio.paolillo@hipperos.com

B. Rodriguez
e-mail: ben.rodriguez@hipperos.com

P. Millet
Thales, Toulouse, France
philippe.millet@thalesgroup.com

modern vision-based applications. Each type of processing element has a type of processing for which it performs best. 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 for 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 [1, 2] aims to push forward a reference platform defining implementation rules to provide designers with guaranteed high-performance solutions for vision-based systems. TULIPP also considers flexibility and scalability for applications demanding more processing power. Therefore, the platform will be set up for heterogeneous multicore and multiprocessor architectures. Several instances of the platform can be combined to further increase processing performance. During the project, a heterogeneous hardware reference platform, a real-time operating system (RTOS) and a productivity-enhancing set of development utilities will be developed. These three components will enable high-performance image processing for modern low-power embedded systems and, in addition, will allow validating the implementation rules and guidelines. The TULIPP project will publish a reference platform handbook, which allows developers to easily follow and apply the implementation rules to find an optimal solution for their image processing application. Thus, TULIPP-compliant custom platforms and applications can be designed at reduced development time and costs.

10.2 The TULIPP Use Cases

The development of the TULIPP reference platform as well as the TULIPP reference platform handbook are use case driven. The three use-case scenarios feature applications for medical imaging, for automotive systems and for UAVs. The applications cover different scenarios, yet the requirements for the embedded systems as such are similar. All three use cases have to operate their image processing applications with high computational performance, low power consumption as well as reduced overall system size and weight. In addition, these applications have to comply with short deterministic latencies, which requires to process images within real-time constraints. Using these three use-case scenarios, the developments of the project, namely the hardware platform, the real-time operating system, and the productivity-enhancing utilities, can be evaluated at the best to provide optimal implementation rules and guidelines for the TULIPP reference platform handbook.

10.2.1 Medical X-Ray Imaging

In medical imaging, mobile equipment is expected to replace high-end infrastructure devices. 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 through veins and arteries. Complex imaging systems have to be used to achieve this.

Dedicated to X-ray instruments, the work of the TULIPP project is highly relevant to a significant part of the market share, in particular through its Mobile C-Arm use case, which is a perfect example of a medical system that improves surgical efficiency. In real time, during an operation, this device displays a view of the inside of a patient's body, allowing the surgeon to make small incisions rather than larger cuts and to target the region with greater accuracy. This leads to faster recovery times and lower risks of hospital-acquired infection. Current X-ray sensors are able to provide live images and video in real time. The drawback of this is the radiation dose: 30 times what we receive from our natural surroundings each day. This radiation is received not only by the patient but also by the medical staff, week in, week out.

While the X-ray sensor is very sensitive, lowering the emission dose increases the level of noise on the pictures, making it unreadable. This can be corrected with proper processing.

From a regulatory point of view, the radiation that the patient is exposed to must have a specific purpose. Thus, each photon that passes through the patient and is received by the sensor must be delivered to the practitioner; no frame should ever be lost. This brings about the need to manage side by side strong real-time constraints and high-performance computing.

The medical use case of the TULIPP project deals with this problem in real-time X-ray image processing. The goal is to reduce the radiation dose to much safer levels while keeping the image quality and the required latency and rate. The image processing is done by an embedded system within a mobile device called 'C-arm' because of its 'C' shape. The X-ray source is located at one end of the C and the sensor at the other end. This shape allows to have the patient in the middle and to take images of any of its body parts through any angle and direction. The C-arm is used during surgery and delivers an X-ray video stream in real time. The chosen algorithms reduce noise from the sensors and enhance the image in order to provide enough details to the surgeon. The algorithms require high-performance processing.

In the TULIPP project, our aim is to try to 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 theatres are confined environments crowded with staff and equipment, the device needs to be small and mobile. A system that integrates the processing close to the sensor is ideal to help reduce extraneous wires and improves the mobility of the equipment. The system needs to be compact but also has a low power draw since heat and other RF emissions could disturb the sensors and eventually actually add more noise to the signal. When we add up the hard real-time constraints to which the system must comply due to 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 (Fig. 10.1).

Fig. 10.1 Digital radiography

10.2.2 Advanced Driver Assistance

In the automotive domain, more electronic devices are going to be integrated into cars in the future. One of the most promising segments for embedded vision systems are the advanced driver assistance systems (ADAS). A steep growth is expected for the next five to ten years. Here, the automotive industry puts a strong focus on driving safety and pedestrian safety with vision-based systems as one of the enablers for many new and innovative solutions. This includes both passive and active safety systems. The most interesting fields of application for embedded vision systems include vehicle, pedestrian and object detection, traffic sign recognition, lane detection, night vision, surround view and driver monitoring. Data from optical sensors is often combined with data from other sensors 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 10 years from now.

The automotive use case of the TULIPP project is focused on pedestrian detection. The purpose of pedestrian detection algorithms is to recognize humans in an image collected by an optical sensor. Detected pedestrians can then be used to trigger further processing, e.g. automated braking. In TULIPP, the implemented algorithm performs the pedestrian detection by feeding a set of trained classifiers various processed forms of an input image.

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 (Fig. 10.2).

Fig. 10.2 ADAS object recognition

10.2.3 Autonomous UAVs

In the domain of unmanned aerial vehicles (UAVs), onboard image processing in real time is a key technology for autonomous operation [3]. Small UAVs have entered a large range of applications as their underlying technology has improved. This also allowed for the exploration of a new and large range of applications. Today, applications for surveillance, search and rescue, video production, logistics and research are just a small subset of possible scenarios [4]. With the growing amount of UAVs, however, the number of crashes and problems with controlled operation is increasing. Problems can be caused by several sources, e.g. operator error as well as mechanical or electrical malfunction. In a worst-case scenario, this error not only involves the UAV itself but also humans, goods or infrastructure [5]. 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 goal of the UAV use case in TULIPP is to estimate depth images from a stereo camera set-up. Orientated in the direction of flight, the depth images are used to detect objects in the path of the UAV. In further stages, detected object is then used for collision avoidance. This approach creates a more autonomous and more intelligent solution. The problem is that more intelligence needs more computing power, which is very limited especially on small UAVs. Yet, image processing has to be onboard in real time, also considering weight and power constraints.

The TULIPP solution aims to fill this processing gap by using its good performance-to-weight and power consumption-to-weight figures. We aim to use computer vision algorithms such as stereo and depth estimation to detect obstacles and evaluate the

Fig. 10.3 Autonomous unmanned aerial vehicle

surroundings in order to make the UAV more intelligent. For this purpose, we attach the TULIPP reference platform with a stereo camera set-up orientated in direction of flight to a UAV. Our goal is to use stereo algorithms to automatically detect obstacles in real time that are within dangerous vicinity in front of the UAV and to avoid a collision. Therefore, we will provide a processing chain which is quite common to any stereo vision-based application. The processing chain contains stereo image acquisition, preprocessing, like image rectification, a depth estimation algorithm based on semi-global matching, similar to [6], an obstacle avoidance algorithm as well as an interface to an external system (UAV) (Fig. 10.3).

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 utilities used to design, develop and deploy an embedded high-performance low-power image processing application.

10.3 The TULIPP Reference Platform Instance

The TULIPP project is developing and will provide a reference platform. The reference platform is presented in the context of the starter kit, a conceptual package consisting of a platform instance, project applications and the reference platform handbook. The aim of the starter kit is to provide engineers with a generic evaluation platform that serves as a base for productively developing low-power image processing applications. The platform instance is a physical processing system consisting of hardware, Operating System (OS), and application development tools. This platform instance demonstrates the results of the project using the applications of the presented use cases.

The reference platform handbook is a set of guidelines for low-power image processing embedded systems. We use guidelines as shorthand for the reference platform handbook. Guidelines recommend application implementation methods supported by the platform instance. A guideline is a goal-oriented, expert-formulated encapsulation of advice and recommended implementation methods for low-power image processing. A vendor platform that enables guidelines by providing suitable implementation methods is called an instance. An instance is fully compliant if it provides recommended implementation methods for all the guidelines that it supports. We envisage that compliance with guidelines will be judged and certified by an independent body identified by the ecosystem of stakeholders.

The reference platform is used to define implementation rules and interfaces to tackle power consumption for high and efficient computing performance demands for image processing applications. The main objective is to provide a new approach to find an optimal solution for a vision-based system. The complex task of designing and evaluating different, interleaving and evolving hardware and software components is then eased so that the overall cost of image processing devices will be reduced drastically. The universal and well-defined interfaces of the reference platform offer the possibility to include new generations of hardware devices and software components without significant overheads and costs for redesigning the system. It allows developers to efficiently design more embedded and less power consuming image processing platforms.

The TULIPP image processing solution aims for scalable high performance and mechanical flexibility to be able to comply with heat dissipation and size constraints, low cost and low power consumption. The inherent idea of customization guides the activities of TULIPP to set up guidelines and definitions on how to use and combine heterogeneous technology at its best. Thus, an optimal solution—in terms of performance, energy efficiency and development costs—for a customized image processing system can be found. Furthermore, TULIPP takes into consideration that the reference platform will evolve at the same pace as modern technology. This ensures that developers can benefit from the improvements future technologies and devices offer but it is a challenge at the same time because the reference platform has to assimilate new technology as it comes up.

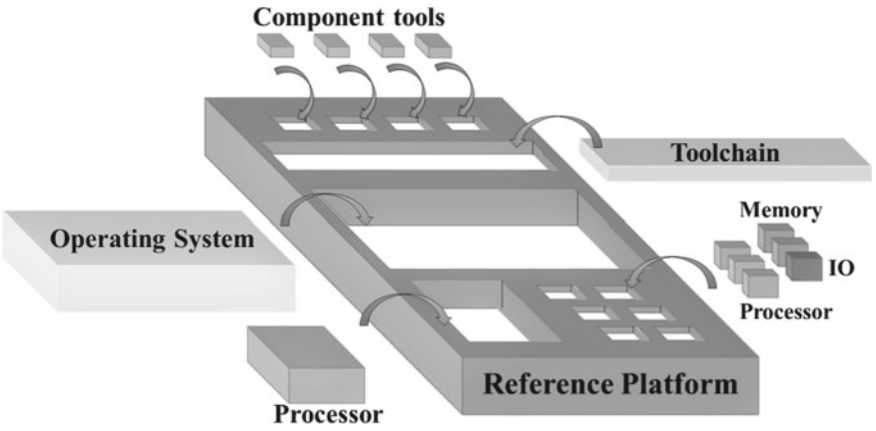

Fig. 10.4 TULIPP reference platform

Instead of developing a generic platform solution that should fit all the applications, TULIPP proposes to focus its efforts on developing a guide for a reference platform that helps the designer in making choices for the components and interfaces. This reference platform is a versatile ideal platform described through a set of guidelines. By following the guidelines, one can implement features to tackle power consumption while delivering high and efficient computing performance for image processing applications under real-time constraints on processing rate and latencies. This solution is much more beneficial and future-proof including all aspects of the development of an image processing platform. TULIPP will build the reference platform through industrial consensus dedicated to low-power real-time image processing applications (Fig. 10.4).

The project will concentrate on interfaces between the components of the platform (hardware, utilities, operating system and middleware libraries) as well as design and implementation processes. Following the guidelines, a developer will be able to produce a compliant platform and benefit from the technological advances generated by the project. In addition, vendors will be able to produce a compliant part and plug it into an existing platform. Thus, a TULIPP-compliant platform instance benefits both to the developers and the vendors. To achieve this goal, the TULIPP project will set up and work closely with an ecosystem formed with platform-part providers (e.g. chip, processing board, operating system, processing libraries, toolchain, etc.) and application developers. This allows incorporating valuable feedback during the project lifetime.

10.3.1 TULIPP Hardware Architecture

Current designs in vision-based embedded solutions are built on single-core CPUs or shared memory architectures. Homogeneous approaches to modern embedded image processing systems are easy to programme but are not an optimal solution regarding energy efficiency and processing performance. High-end vision systems in the automotive industry feature heterogeneous architectures. The drawback is that each iteration during design time and each new generation of technology requires a huge implementation effort.

TULIPP will focus its work on heterogeneous systems in image processing applications. Different processing elements will be combined in a hardware architecture, where each processing element is best suited for certain parts of an image processing application. As an example, a small 32-bit CPU can be used for controlling in- and outputs. The processing of images can then be executed on multiple 64-bit CPUs with additional acceleration by FPGAs or embedded GPUs.

The TULIPP reference platform provides a template for heterogeneous computing architectures and systems. Modern SoCs demonstrate the potential of heterogeneous systems. The NVIDIA Tegra-K1 [7] provides high performance by combining an ARM processor with a GPGPU. Similarly, Xilinx puts great efforts on the Zynq devices. Here, an ARM processor is combined with an FPGA. Xilinx UltraScale+ MPSoCs takes one step further and combines 64-bit and 32-bit ARM architectures together with dedicated real-time cores and an FPGA [8]. The goal of the TULIPP reference platform is not only to optimally utilize a single SoC. Moreover, the project aims to connect different SoCs. Different parts of an image processing application can then be run on the best-suited computing architecture. An adaptive system allows running an application energy efficient yet high performant. The hardware platform will be fine-tuned and configured for each application. Therefore, TULIPP defines how to select SoCs suitable to build a TULIPP platform instance and how to efficiently interconnect several SoCs. Switch-off mechanisms, adjustable operating frequencies and dynamic partial reconfiguration (DPR) [9] further reduce the cost of unused system resources during runtime.

As of the time of writing, TULIPP uses Xilinx Zynq SoCs to thoroughly test different combinations of hardware and interfaces on- and off-chip. The inherent heterogeneity of these devices is used to derive definitions, implementation rules and guidelines for the TULIPP reference platform and the TULIPP reference platform handbook. The TULIPP hardware architecture uses the PC/104 form factor. This form factor is already supported by many vendors, which are committed to the ongoing development of this specification [10]. It is mature, modern, open standard and expandable but also capable for stand-alone applications. These features are perfectly suited for a scalable and heterogeneous TULIPP hardware platform.

10.3.2 TULIPP Operating System and Low-Level Libraries

Today, there is a significant gap between research and commercial implementation of an RTOS. Research results feature scheduling algorithms, resource sharing algorithms and inter-process communication (IPC) protocols. Yet these innovations are rarely incorporated in commercial systems. Existing RTOS are mainly targeted towards single-core designs, and multicore approaches focus on homogeneous SMP architectures. In addition, the need for energy efficiency is rarely supported. This includes, for example, power-aware scheduling at the kernel level [11]. As a result, system lifetime and reliability are reduced [12].

In TULIPP, the design of the operating system and low-level libraries is targeted towards low power consumption and image processing. The RTOS kernel supports heterogeneous architectures and power-aware features. Communication and synchronization mechanisms are implemented so that the operating system correctly operates on the instantiated processing elements of the hardware platform. The footprint—i.e. the binary size—of the RTOS is kept small for hardware components only embedding a small local memory. Frequent accesses to bigger memories like DDR are not suitable for running image processing applications in real time and consume more energy. The TULIPP reference platform provides primitives so that components can be integrated or wrapped in the low-level library available for the programmer. In addition, standard APIs will be modified and extended to comply with the requirements of low power consumption and high-performance image processing. Thus, extensions and modified standards are proposed as a pre-norm.

The RTOS in TULIPP is developed to efficiently handle heterogeneous hardware processing resources of multicore CPUs and FPGAs with a strong focus on finding the right balance between low power footprint and high computing performance.

The RTOS solution proposed by the TULIPP project is a new master–slave microkernel architecture specifically designed for heterogeneous multicores. It features a small footprint, low power consumption and good scalability. This is a combination of several features. Power-aware schedulers—i.e. extended earliest deadline first (EDF) instead of rate monotonic (RM) schedulers [13]—reduce the overall power consumption of applications. Moreover, it has been shown in the real-time literature that schedulers based on a parallel task model are well suited to be extended to power-aware scheduler [14]. Therefore, the TULIPP platform includes an RTOS capable of scheduling parallel real-time tasks (software or hardware) associated with the right runtime libraries allowing to easily design parallel workload to be run on the different heterogeneous components of the target platform. This, combined with optimizations provided by the offline utilities of TULIPP presented in the next section will result in an image processing embedded system suited to the user requirements in terms of power consumption and computing performance.

The hard real-time scheduling of hardware/software tasks is combined with virtual memory management to isolate processes and efficient IPC mechanisms to allow these processes to communicate. Developers can then use the reliable real-time guar-

antees and easy programmability of the provided RTOS for optimized low-power image processing applications.

The TULIPP real-time operating system is designed by HIPPEROS and based on its family of RTOSes [15, 16]. It supports standard tools to interface with hardware (bootloaders, debuggers, etc.), and low-level libraries shipped with the operating system support APIs validated for low-power embedded image processing applications (POSIX, OpenCV, OpenMP, etc.) [17]. The operating system interfaces are developed to be optimally integrated with the TULIPP hardware platform and the supporting development toolchain and utilities. Additionally, the RTOS environment is adapted to suit the modern advantages of the heterogeneous platforms [18]. For example, for heterogeneous platforms as the Zynq device, the HIPPEROS RTOS provides APIs to partially reconfigure at runtime what is running in the FPGA, enabling the Xilinx dynamic partial reconfiguration feature and exposing it to the image processing application developer (the user of the TULIPP platform).

Thus, the main advantage of the TULIPP RTOS solution is its efficiency for heterogeneous parallelism and power optimization.

10.3.3 TULIPP Toolchain

The current state of the art in development tools for heterogeneous image processing systems requires a lot of interaction and experience with several different vendor-specific tools. Each component is accompanied by its own complex tools. The developer has to spend a significant amount of time to master these tools. This results in low productivity and a reduced innovation rate. Extensive reviews of such heterogeneous system development tools exist [9, 19, 20]. A TULIPP-compliant platform can feature hardware components from several different vendors. In general, each component is supplied with its own specific toolchain and Integrated Development Environment (IDE). To efficiently develop low-power, high- performance applications on TULIPP hardware platforms, the programmer needs to gain expertise in several tools. This inhibits productivity, as such expertise takes a long time to develop. In addition, lack of knowledge or expertise in a particular vendor—device or tool—may also prevent developers from selecting hardware components, which would be best suited for the power and performance requirements of a specific image processing application.

TULIPP proposes a solution using toolchain utilities that allow developers to use multivendor tools more efficiently and productively. The focus of the utilities is put on an improved system set-up, its analysis and optimization. This includes mapping an application optimally onto a heterogeneous, TULIPP-compliant platform. 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. Developers are able to map parts of an application onto suitable components using STHEM interfaces. Primitives and library routines can be used to handle control and communication of the components. Problem areas of the application can be

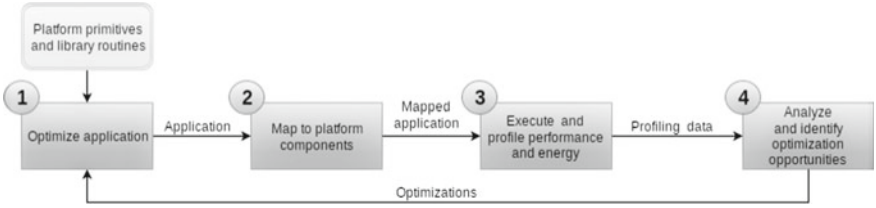

Fig. 10.5 High-level overview of the iterative workflow using STHEM

identified with STHEM's utilities for performance and energy consumption analysis. Furthermore, the utilities also identify optimization opportunities. Thus, the developer is guided towards an energy-efficient and high-performant image processing application and platform.

An overview of the workflow is given in Fig. 10.5. STHEM provides efficient usability of expert-written mechanisms for the improvement of an application. The workflow is iterated until desired performance and energy profile is reached. In the first stage of the workflow, programmers write application code. The optimizations identified in the previous iteration are also applied in this first stage. Programmers are also assisted with platform-specific primitives and library routines that abstract away commonly used domain-specific functionality. In the second stage, developers are supported in the mapping of an application to components using easy-to-use, expert-written mapping directives. The third stage is used for platform configuration, application execution and profiling. Analysis of the execution and profiling is provided in the fourth stage. Visualizations are used to highlight profiling data and problem areas [21]. In addition, STHEM suggests optimization strategies to programmers and offers to automatically explore the design space.

STHEM is built as an Eclipse 4 RCP plugin to facilitate an integrated workflow with popular vendor tools that integrate into the Eclipse IDE [22–24]. The first implementation of STHEM is designed for Xilinx SDSoC [22]. Thus, all the components of the TULIPP reference platform—heterogeneous hardware platform, multicore RTOS and supporting utilities—can be evaluated and tailored towards optimal energy efficiency and high computing performance for image processing applications.

With the three components described above—hardware architecture, operating system and toolchain—the TULIPP project aims to provide rules and definitions for a reference hardware platform. Thus, the TULIPP project does not only provide an extensible set of components for embedded high-performance low-power image processing applications. Moreover, the insights and knowledge gained by developing these components is comprehensively written down in the TULIPP reference platform handbook.

10.4 The TULIPP Reference Platform Handbook

The TULIPP reference platform handbook provides guidelines on developing and optimizing a low-power high-performance image processing application and embedded vision system. As described in the previous chapters, it is a complex task to develop an image processing system that complies with the constraints of embedded systems and the demand for low power consumption and high computing performance. Embedded vision systems feature a complex combination of different design steps, including component selection and platform set-up, application and algorithm optimization as well as application mapping.

The guidelines provided by the TULIPP reference handbook support the developer by managing the complexity of designing modern low-power image processing embedded systems. Therefore, guidelines are derived from the expert knowledge gained during the development of the TULIPP reference platform. This knowledge is split into advices and recommended implementation methods. Advices support the developer on what to do so that an image processing application and platform can be optimized. The recommended implementation methods describe in detail how to achieve the suggested optimizations.

The guidelines of the handbook consider all the components of the TULIPP reference platform. Developers are supported in the selection of TULIPP-compliant hardware components and operating system settings best suited for their application. In addition, guidelines are formulated on how to rewrite the application and how to use APIs, libraries and pragmas achieving a more efficient image processing application. The guidelines always put the focus on low power consumption, high and heterogeneous computing performance and real-time image processing.

As a long-term goal, the TULIPP handbook aims to improve productivity as well as influence new standards for heterogeneous embedded vision systems operating under real-time constraints with low power consumption and high computing performance. The expert insights of the guidelines then not only support developers by optimizing their application, but also vendors by providing TULIPP-compliant hardware and software. For developers, this has the potential to significantly improve productivity. Costly mistakes are avoided and vast design and implementation spaces are pruned. Vendors are encouraged to strive for compliance by providing suitable hardware and software. Thus, the efforts of the TULIPP project pave the way for future standards in embedded low-power image processing systems.

10.5 Future Goals and Outlook

The main objective of the TULIPP project is to develop a reference platform for high-performance and energy-efficient embedded systems for image processing applications. In addition to this reference platform, TULIPP is going to support developers

with a reference platform handbook. The handbook provides guidelines on developing and optimizing low-power high-performance embedded vision systems.

To achieve its goals, the TULIPP project develops and provides a starter kit. This Starter Kit is used to demonstrate the potential of the projects' approaches to the design, development and implementation of embedded vision systems. The TULIPP starter kit allows the project to track its efforts and demonstrate its use cases. Furthermore, the kit also allows developers to design high-performant yet low-power image processing platforms and applications. The starter kit will feature a first version of the TULIPP reference platform and handbook. The hardware component consists of a PC/104 board with a small system-on-module (SoM). Here, the SoM features a Xilinx Zynq device for heterogeneous computing performance. The toolchain component consists of a collection of application analysis utilities referred to as support utilities for heterogeneous embedded image processing (STHEM). STHEM will augment existing vendor toolchains by automating analysis procedures and supporting more efficient application design. This reduces time-to-market, improves developer productivity and system quality by making it easier and faster to arrive at an implementation that meets the requirements. The hardware platform and STHEM will be completed with the HIPPEROS real-time operating system and its low-level libraries, allowing to design and run highly efficient embedded vision-based applications and platforms.

The reference platform and handbook cover all the aspects of designing embedded systems for vision-based applications: from computing hardware, operating system and low-level libraries to programming toolchain and utilities. The reference platform also defines a set of interfaces between basic components and implementation rules to facilitate prospective system design. The implementation rules and guidelines of the reference platform handbook support developers designing embedded image processing platforms. This significantly contributes to a reduction of time-to-market and development costs. The TULIPP project is going to leverage the utilization of heterogeneous embedded computing platforms for image processing applications. Thus, the TULIPP project aims to pave the way for standards for embedded high-performance low-power image processing in industrial applications.

The TULIPP project establishes a valuable advisory board and ecosystem. The advisory board consists of leading experts from industry and academia with a strong focus on image processing applications and platforms targeted towards embedded systems. Thus, valuable feedback is used during the project to further leverage and promote its developments. Incorporating high-quality feedback and utilizing its developments and guidelines the TULIPP project puts great efforts into industrial usage and standardization of ubiquitous low-power embedded systems for image processing platforms. The state and progress of the project will be available to the public at the website of the TULIPP project [1].

Acknowledgements The project is funded by European Commission under the H2020 Framework Programme for Research and Innovation under grant agreement No 688403.

References

1. Tulipp (2017) TULIPP: towards ubiquitous low-power image processing platforms—high, efficient and guaranteed computing performance for image processing applications. http://www.tulipp.eu. Accessed on 11 May 2017
2. Kalb T et al (2016) TULIPP: towards ubiquitous low-power image processing platforms. In: Proceedings of the international conference on embedded computer systems: architectures, modeling and simulation (SAMOS XV)
3. Sung C-K, Segor F (2012) Onboard pattern recognition for autonomous UAV landing. In: Proceedings of the SPIE 8499, applications of digital image processing XXXV, 84991K
4. Kuntze HB et al (2012) SENEKA–sensor network with mobile robots for disaster management. In: 2012 IEEE conference on technologies for homeland security (HST). Waltham, MA, pp 406–410
5. Tchouchenkov I, Segor F, Schoenbein R, Kollmann M, Bierhoff T, Herbold M (2016) Detection and protection against unwanted small UAVs. In: Proceedings of the eleventh international conference on systems ICONS
6. Ruf B, Schuchert T (2016) Towards real-time change detection in videos based on existing 3D models. In: Proceedings of SPIE, Edinburgh, UK, vol 10004, pp 100041H–100041H-14
7. NVIDIA Corporation (2014) Whitepaper: NVIDIA Tegra K1—a new era in mobile computing. https://ww.nvidia.com/content/PDF/tegra_white_papers/tegra-K1-whitepaper.pdf. Accessed on 28 April 2017
8. Xilinx Inc. (2015) 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. Accessed on 28 April 2017
9. Cong J, Liu B, Neuendorffer S, Noguera J, Vissers K, Zhang Z (2011) High-level synthesis for FPGAs: from prototyping to deployment. IEEE Trans Comput Aided Design Integr Circuits Syst 30(4):473–491
10. PC/104 Consortium (2017) PC/104 Consortium—supporting legacy technology while developing new solutions for the future PC/104 Consortium. http://pc104.org. Accessed on 11 May 2017
11. Mentor Graphics (2017) Nucleus RTOS. https://www.mentor.com/embeddedsoftware/nucleus/. Accessed on 04 May 2017
12. Brandenburg B (2011) Scheduling and locking in multiprocessor real-time operating systems. PhD dissertation, The University of North Carolina
13. Blazewicz J, Ecker KH, Pesch E, Schmidt G, Weglarz J (2001) Scheduling computer and manufacturing processes. Springer, Berlin. ISBN 3-540-41931-4
14. Paolillo A, Goossens J, Hettiarachchi PM, Fisher N (2014) Power minimization for parallel real-time systems with malleable jobs and homogeneous frequencies. In: The 20th IEEE international conference on embedded and real-time computing systems and applications, Chongqing, China, August 2014
15. Paolillo A, Desenfans O, Svoboda V, Goossens J, Rodriguez B (2015) 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
16. HIPPEROS S.A. http://www.hipperos.com. Accessed on 09 May 2017
17. Paolillo A, Rodriguez P, Veshchikov N, Goossens J, Rodriguez B (2016) Quantifying energy consumption for practical Fork-Join parallelism on an embedded real-time operating system. In: The 24th ACM international conference on real-time networks and systems, Brest, France, October 2016
18. Martin C, Antonio P, Goossens J, Rodriguez B (2017) Research and implementation challenges of RTOS support for heterogeneous computing platforms. In: HARTS-ULB, Brussels, Belgium, May 2017

19. Jeffers J, Reinders J (2015) High performance parallelism pearls volume two: multicore and many-core programming approaches. Morgan Kaufmann
20. Mittal S, Vetter JS (2015) A survey of CPU-GPU heterogeneous computing techniques. ACM Comput Surv (CSUR) 47(4):69
21. Muddukrishna A, Jonsson PA, Podobas A, Brorsson M (2016) Grain graphs: OpenMP performance analysis made easy. In: Proceedings of the 21st ACM SIGPLAN symposium on principles and practice of parallel programming, New York, NY, USA. pp 28:1–28:13
22. Xilinx Inc. (2017) SDSoC environment user guide (UG1027). http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_4/ug1027-sdsoc-user-guide.pdf. Accessed on 09 May 2017
23. NVIDIA Nsight|NVIDIA. http://www.nvidia.com/object/nsight.html. Accessed on 09 May 2017
24. 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 May 2015