DOI 10.5162/ettc2018/10.2

PARIS - Parallelisation Architecture for Real-time Image Data Exploitation and Sensor Data Fusion

Stephan Blokzyl, Michael Nagler, René Schmidt, Wolfram Hardt
Computer Science Department, Chemnitz University of Technology, D-09107 Chemnitz, Germany
{sblo, nagm, renesc, hardt}@hrz.tu-chemnitz.de

Abstract:

The development of visual recognition systems for highly automated, mobile systems is driven by matured electro-optical sensors and application-specific, optimised computational components. Applying good-quality, high-resolution imagery exploited by powerful embedded processors, current computer vision systems provide a detailed representation of system's operational environment and contribute significantly to various perception tasks. However, the exploitation of high-resolution image data is challenging on space-, weight-, and energy-limited mobile systems, especially when respecting real-time and reliability requirements of safety critical functions.

Considering these conditions, this paper introduces a generic parallelisation architecture for real-time image processing using reconfigurable integrated circuits. The architecture supports data and task parallelisation strategies utilising the specific parallelisation capabilities of Field Programmable Gate Arrays. It provides space-, weight-, and energy-efficient parallelisation for image exploitation and information fusion.

The device- and interface-independent architecture maximises parallelism and flexibility of complex image processing applications aboard mobile systems. The modular structure of the proposed architecture enables hardware-acceleration for high-resolution sensor data exploitation, minimises processing latency, and improves the quality of the overall detection result by using multivariate detection methods.

The generic parallelisation architecture will be used for multi-copter-based high voltage transmission line inspection and vision-based localisation of an unmanned aerial vehicle during final approach phase.

Key words: Visual Recognition Systems, Mobile Systems, Parallelisation, Real-time Image Processing, Field Programmable Gate Arrays.

Introduction

The development of computer vision-based measurement and recognition systems is continuously driven by technology progress. Hardware innovations provide more powerful sensors and computing architectures. New methodologies improve sensor data acquisition and signal processing. The growth of interconnection and interaction between computing devices raises heterogeneity and interoperability of advanced computer vision solutions [1].

More powerful sensors with increasing frame rate and resolution generate considerably higher sensor data volumes. High-quality, lossless acquisition, processing, and management of the rising amount of data gets more and more demanding. Especially in automotive, transportation, avionic, and space applications, which typically realise sensor signal processing on embedded systems, and have to consider safety, reliability, and real-time aspects [2].

Devices like user-programmable integrated circuits (e. g. Field Programmable Gate Arrays, FPGAs) are increasingly popular for embedded, high-performance image data exploitation (brief overview given in [3]). They combine the parallelisation capability and processing power of application specific integrated circuits (ASICs) with the flexibility, scalability, and adaptability of software-based processing solutions. FPGAs provide powerful processing resources due to an optimal adaptation to the target application and a well-balanced ratio of performance, efficiency, and parallelisation.

DOI 10.5162/ettc2018/10.2

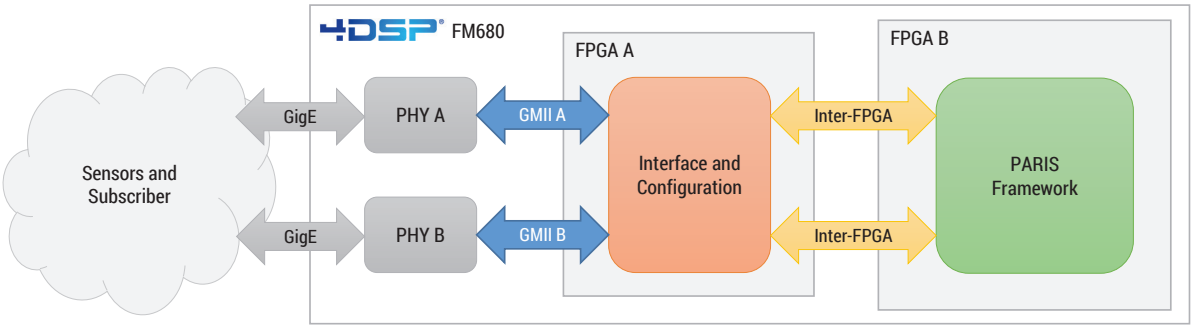

Fig. 1. Ethernet-based interconnection of the PARIS system architecture to the operational environment.

These features are fundamental for the hardware-accelerated real-time image exploitation concept, presented by Blokzyl *et al.* in [4]. With a two-dimensional parallelisation methodology (data vs. task parallelisation), the approach intends to accelerate the image processing procedure and improves the overall processing and result quality. The concept focuses on specific requirements on system components for safety-critical perception applications:

- Flexibility through modularisation and re-configurability
- Result quality enhancement by multivariate exploitation
- Process acceleration by execution parallelisation
- Result predictability by use of deterministic algorithms
- Determinability of worst-case-execution-time
- Result safety and confidence qualification
- Certifiability for safety-critical applications

The powerful realisation of the two-dimensional parallelisation methodology requires specific attributes of the hosting hardware. Beside flexibility, scalability, and a high degree of parallelisation capabilities, properties like power consumption, energy efficiency, size, and weight play an important role in embedded applications.

Hence, this work introduces the PARIS[1] system architecture, a framework for easy integration of data and task parallelisation for sensor data exploitation, data fusion, system management, data flow, and sensor control (compare Fig. 1).

System Architecture

The PARIS system model comprises the following sub-components: Network integration and network management, system and sensor management, as well as image data exploitation and fusion. All elements and their detailed functions are explained in the following sections.

A. Network Integration and Network Management

Network integration and network management includes both physical and logical interconnection of the PARIS system architecture with the operational environment. The PARIS interfaces use Gigabit Ethernet standard to connect all sensors and result subscribers, but the free configurability of FPGAs allows the utilisation of alternative communication standards and protocols (compare Fig. 1).

Popular digital signal processing FPGAs (e. g. XILINX Virtex® or Altera Stratix® series devices) provide ready-to-use intellectual property (IP) cores for the Data Link layer to enable ease Gigabit Ethernet communication. This IP for Media Access Control (MAC IP) supports Ethernet-based communication without additional user implementation. Layer 1 functions including message transmission (Physical Layer) are realised by external transceivers (see PHY A/B in Fig. 1) and functions of Layer 3 and higher (compare Open System Interconnection Model, OSI model [8]) can be customised with the respect to the target application.

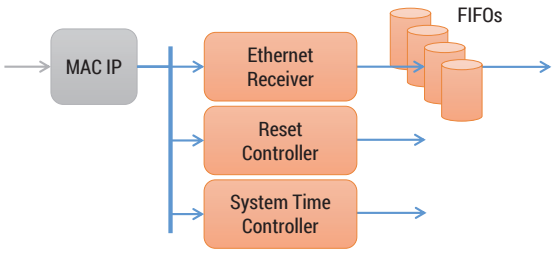

Fig. 2 Basic receiver components of the PARIS system architecture.

[1] **P**arallelisation **A**rchitecture for **R**eal-time **I**mage data exploitation and **S**ensor data fusion

DOI 10.5162/ettc2018/10.2

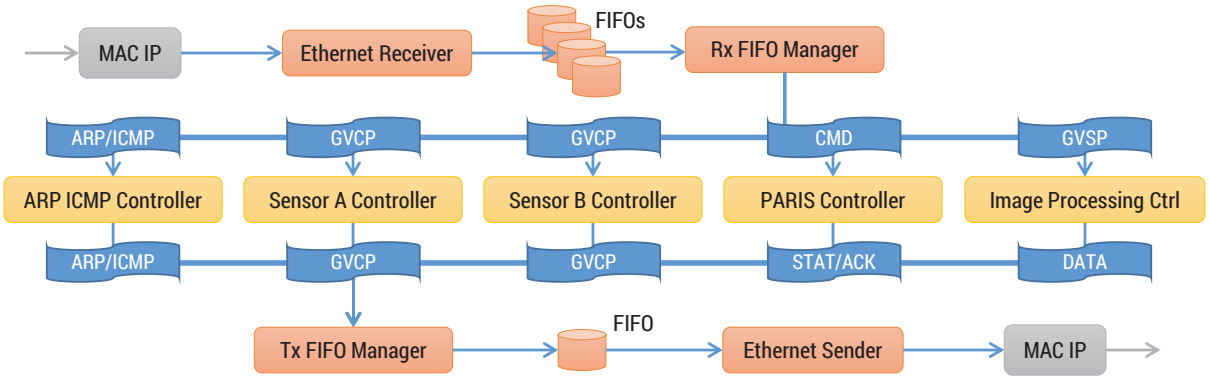

Fig. 3. Overview of receiver, sender, and controller structure of the PARIS system architecture.

The communication between PARIS and PHY transceivers uses the Gigabit Media Independent Interface standard (GMII). GMII is independent from the communication medium and a change from e. g. copper wire-based to fibre-optic transmission does not affect the link between PARIS and the peripherals. Only copper wire-based transceiver modules have to be replaced by devices supporting fibre-optic interconnection. Furthermore, changing Ethernet communication to an alternative technology, like e. g. Peripheral Component Interconnect (PCI) requires the modification of communication IP core only (here MAC IP, Ethernet Receiver in Fig. 2). It has no impact on other system components of the PARIS system model.

This flexibility maximises the integration capability of PARIS in comparison to conventional system architectures. PARIS is not limited to specific FPGA devices or series and can be operated on various boards with arbitrary form factors, physical connections, and interfaces.

PARIS is equipped with two independent Gigabit Ethernet interfaces (A/B) providing a total bandwidth of 2 Gbit/s for sensor data acquisition and result dissemination. As shown in Fig. 2, incoming messages arrive at the MAC IP that extracts the message payload (Ethernet frame) and simultaneously forwards the data to the Ethernet Receiver, Reset Controller, and System Time Controller. The parallelised setup of these three subsequent modules is necessary to ensure global system reset in any error case (e. g. rest of the PARIS is stalled) and to minimise the latency of system time synchronisation. These modules are connected unidirectional to the MAC IP. They have no control access to MAC IP and have to be ready to receive incoming Ethernet frames anytime. Reset and System Time Controller exclusively accept Reset Protocol (RP) respectively Simple Network Time Protocol (SNTP) messages and process these packets in real-time. The Ethernet Receiver accepts all the other message types and caches them in different receive buffers (First-In-First-Out buffers, FIFOs).

If the MAC IP advices the reception of a new message, the Ethernet Receiver selects an empty receive buffer and saves the minimum packet payload. Destination and checksum information of Ethernet frames is not saved as the PARIS discards misaddressed or damaged incoming messages. Source address data is saved in cases when it is necessary to reply to original sender (e. g. command acknowledge). Saving only minimum payload reduces memory consumption, minimises stall and wait times of concurrently reading modules, decreases processing latency, and prevents data loss.

The application of multiple message buffers allows the parallel processing of incoming data, which accelerates the overall system performance. If only one single buffer was used, the retrieval of large-sized messages would result in longer wait times for non-active parallel modules. Subsequently, already completely received data could not be processed simultaneously because the receive buffer data output is already blocked. As the PARIS predominantly processes image data, the number of large-sized packets is very high. The byte count of image packets (so called jumbo frames) exceeds the 1.5 KB Maximum Transmission Unit (MTU) of standard Ethernet frames. Fetching and exploitation of image jumbo frames takes considerably longer compared to the processing of e. g. small-sized network management messages like Address Resolution Protocol (ARP) or Internet Control Message Protocol (ICMP). Reply and validity times could be neglected. The application of multiple receive buffers reduces this bottleneck and enables parallel data readout and exploitation, which leads to significantly less delay between data reception, processing, and result dissemination.

All error-free frames in the receive FIFOs are forwarded by the System and Sensor Manage-

ment to the appropriate sub-components of the PARIS system architecture (see Fig. 3).

B. System and Sensor Management

As introduced in section A, all received and valid frames are cached in receive buffers (FIFOs) by the Ethernet Receiver. A connected Receive Buffer Manager (Rx FIFO Manager, compare Fig. 3) controls the fetching and distribution of the received data across the PARIS system architecture. The Rx FIFO Manager analyses the message types and distinguishes the following message classes:

1. Address Resolution Protocol (ARP)
2. Internet Control Message Protocol (ICMP)
3. Internet Group Management Protocol (IGMP)
4. GigE Vision Control Protocol (UDP/GCSP)
5. GigE Vision Stream Protocol (UDP/GVSP)
6. System Command Protocol (CMD/UDP)

The Rx FIFO Manager monitors cyclically all receive buffers and checks one buffer at each clock cycle. With e. g. four receive buffers and a system operation frequency of 125 MHz, a new and completely received Ethernet frame is recognised with a maximum latency of 24 ns. According to the message type (indicated by the buffer state), the associated sub-component is connected to the corresponding buffer and starts data readout. The receive buffer is released as soon as the connected module completes data transmission and the Rx FIFO Manager resets the buffer for receiving new data frames. As a result, all messages are forwarded to the different PARIS modules (controllers) depending on their message type:

ARP ICMP IGMP Controller The ARP ICMP IGMP Controller implements the OSI model Layer 3. This module is instantiated two times and manages the network setup and communication parameters (addressing, routing, and traffic control). The two controllers operate the Gigabit Ethernet interfaces of the PARIS system architecture as introduced in section A.

Sensor Controller The Sensor Controllers manage all external imaging devices based on the GigE Vision standard. The GigE Vision standard is an interface and communication standard for Ethernet-based electro-optical sensors (GigE Vision Control Protocol, GVCP). The GVCP has been developed by the Automated Imaging Association (AIA) in the year 2006. As the PARIS framework is equipped with a stereo camera system with two independent electro-optical sensors, two entities of the Sensor Controller exist (A/B). The controllers are duplicable and replaceable with alternatives which enables the application of arbitrary sensors for different recognition and measurement tasks.

PARIS Controller The PARIS Controller monitors and controls the operation of the PARIS system model and handles all incoming CMD messages. The CMD message type contains system configuration data (system parameter) and commands the available work modes. Successful processing of CMD messages is replied by acknowledgement packets (ACK). Continuous status messages (STAT) report the PARIS system (health) and operation status.

Image Processing Controller The Image Processing Controller represents the core module of the PARIS system architecture. The module hosts all image processing units and utilises data and task parallelisation strategies. The Image Processing Controller obtains the image data fragments from the electro-optical sensors (GigE Vision Streaming Protocol, GVSP) and implements pre-processing as well as imagery exploitation (more details given in section B). The result data of the Image Processing Controller is encapsulated in Ethernet frames (DATA) and addressed to the result subscribers. The index of signed in result subscribers is managed by the PARIS Controller.

The processing results of the different controllers (ARP, ICMP, IGMP, GVCP, ACK, STAT, and DATA) are enclosed in Ethernet frames and stored in a wait buffer for sending. A Send Buffer Manager (Tx FIFO Manager) controls the write access to the send FIFO with the help of a token system. If a PARIS sub-component demands write access, it requests a token from the Tx FIFO Manager. The module starts data transmission as soon as the Tx FIFO Manager grants write rights with the help of a handshake mechanism. The module releases the token after transmission completion and a competitive PARIS component can access the send buffer. The send FIFO is realised as singular buffer for each Ethernet interface, because the small lengths and size variances of the outgoing messages lead to significantly less competitive access compared to the receive buffers. An average size of approximately 60 bytes per packet causes a mean wait time of approximately 500 ns until a token is released again[2].

[2] With a system operation frequency of 125 MHz.

DOI 10.5162/ettc2018/10.2

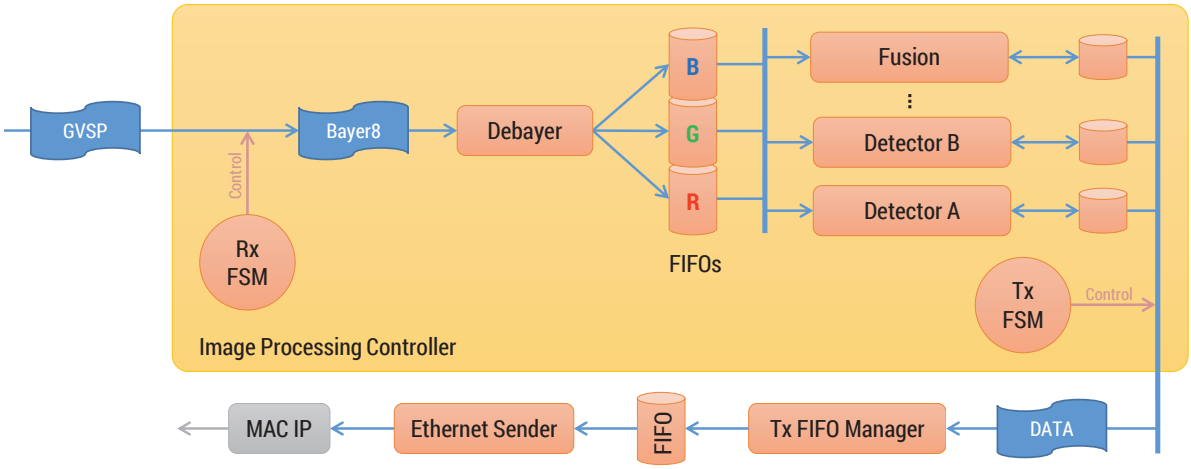

Fig. 4. Data flow and two-dimensional parallelisation inside the Image Processing Controller.

If the PARIS framework hosts applications which generate result data with obviously different-sized result messages, the send buffer can be optionally extended with additional FIFO memories. Multiple send buffers allow simultaneous sending of small, completely cached Ethernet packets while large-sized messages are continuously buffering. This reduces bilateral blocking of PARIS sub-components.

The send buffer status is steadily supervised by the Ethernet Sender. Completely cached Ethernet frames are transmitted to the MAC IP as soon as the MAC IP is ready to accept data and the send medium is free.

C. Image Processing and Fusion

The Image Processing Controller is the core module of the PARIS system architecture. It hosts the parallelisation framework for real-time image data exploitation and fusion. The controller involves all steps for sensor data acquisition, image pre-processing, and image data exploitation. Data acquisition names the (image) payload de-fragmentation of the incoming GVSP messages (image fragments). The extracted raw image stream is typically a Bayer mosaic image that has to be converted to a three-channel colour image, which can be processed by subsequent image processing modules (detectors). The colour image generation is realised by the Debayer module (see Fig. 4). The three resulting partial images represent the image channels Blue, Green, and Red, which are stored in image synchronisation buffer (B/G/R FIFOs).

Parallel image processing modules start execution as soon as enough image data is available in the image synchronisation buffers. The following module with the highest data requirement determines the time to start processing. If one module requires 49 pixels and another (parallel) module requires only nine pixels to start execution, both modules stall the processing until 49 pixels are available in the image synchronisation buffer (start synchronisation). This earliest start strategy avoids the buffering of the complete image and reduces both processing latency and memory consumption. As result, the image buffer size decreases significantly. The PARIS hardware realisation uses only 12.3 KB of FPGA-BRAM instead of 2,073.6 KB DDR-SDRAM which are required for the software-based processing of two megapixel colour images (high definition resolution). The PARIS system architecture saves more than 99% of image buffer size compared to conventional, software-based image processing solutions.

As shown in Fig. 4, an arbitrary number of various, parallel image processing modules can be connected to the data output bus of the image synchronisation buffer. The different run- and execution times are synchronised by individual result buffers for each image processing module (result synchronisation). This structure qualifies the PARIS framework to use data and task parallelisation strategies as well as static and dynamic in-system-reconfiguration.

Data Parallelisation Multiple instances of independent detectors enable the distribution of incoming sensor data to multiple, homogeneous processing entities. Slow image processing algorithms with a longer execution time compared to the validity time of incoming sensor data (update rate) are accelerated by workload sharing. The detection results of the different image processing modules are aggregated subsequently in a common synchronisation buffer (result aggregation).

Task parallelisation The parallel application of heterogeneous, simultaneously working detec-

tors increases the overall detection result quality. All image segments are analysed by heterogeneous, weak detectors realising different low-level image processing methods on equal image segments. The diverse intermediate results are merged in a subsequent fusion step that generates detections on a higher level of abstraction contributing to the global detection task.

Data Aggregation and Information Fusion All partial and intermediate results are combined in result buffers and validated by a confidence score (compare Fig. 4). This step comprises the synchronisation and aggregation of data parallelisation results and the information fusion of diverse intermediate results from task parallelisation. More details of the concept and the realisation of the fusion approach, including the integration of data from other sources and sensors, are given in [4] and [5].

In-System-Reconfiguration The parallel and modular architecture of the image processing controller enables both static and dynamic reconfiguration without complete deactivation or stalling of the image processing system. The replacement, integration, or exclusion of dedicated detectors enables the PARIS framework to respond to changing recognition conditions and varying detection tasks during run-time. The PARIS is able to adapt the degree of parallelisation inside the image processing core. This guarantees an anytime optimal compromise between energy consumption (power dissipation respectively heat development), computing performance, and acceleration rate.

All partial, aggregation, and fusion results are encapsulated in Ethernet frames after successful processing. The Ethernet messages are pushed to the send buffer (compare section B) which is supervised by the Ethernet Sender. As soon as the send buffer status indicates a completely buffered Ethernet frame and the MAC IP reports readiness to send, the Ethernet Sender transmits the pre-buffered frame to the MAC IP. The MAC IP forwards the message subsequently to the connected PHY transceiver via GMII (see Fig. 4).

Results

The introduced PARIS system architecture is realised on the hardware acceleration board FM680 of the US-American company 4DSP, worldwide leading experts in the area of Commercial-of-the-Shelf (COTS) board-level electronics and FPGA intellectual property. The board is equipped with two XILINX Virtex® family FPGAs (see Fig. 1). One Virtex®-5 XC5VLX20T FPGA hosts the interface implementation for Gigabit Ethernet communication and manages the system boot-up process with an external power-on configuration flash memory. The large-scale Virtex®-6 XC6VLX550T FPGA with 86,000 slices, 550,000 logic cells, 687,000 flip-flops, and approximately 3.6 Megabyte on-board-RAM [6] realises system and sensor control as well as the sensor signal processing (image data exploitation). These powerful large-scale FPGAs enable high acceleration potential for a wide range of highly parallel image exploitation and data fusion applications.

Logic Resources	Utilisation 6VLX550T *PARIS Framework*	Utilisation 5VLX20T *Interface Device*
Slice Registers	9,401 (1.4%)	2,022 (16.2%)
Slice LUTs	13,364 (3.9%)	1,952 (15.6%)
Occupied Slices	4,491 (5.2%)	1,011 (32.4%)
RAM Blocks	44 (2.3%)	4 (15.4%)

Tab. 1 FPGA Utilisation for the PARIS System Architecture

The hardware circuitry of the PARIS system implementation achieves a maximum operation frequency of 125.881MHz with an appropriate logic resource consumption as shown in Tab. 1. The economic, logic resources saving hardware realisation of the PARIS system architecture allows efficient integration of a large number of data and task parallelised image exploitation components and the implementation of complex image processing applications on FPGA devices.

Conclusion and Outlook

The novel PARIS system architecture provides a solution for modular, flexible, and comfortable parallelisation of real-time image data exploitation. The architecture supports both data and task parallelisation strategies, static and dynamic in-system-reconfiguration, and considers fundamental requirements for embedded signal processing architectures like e. g. power consumption, energy efficiency, size, and weight restrictions. The PARIS framework is implemented on a COTS hardware acceleration board, will be proved in a stereo vision-based localisation system of an unmanned aerial vehicle during final approach phase [5][7], and for multi-copter-based high voltage transmission line inspection [9].

Acknowledgment

The results have been supported by the program OPIRA - Open Innovation for Remotely Piloted Aircraft Systems (RPAS), funded by the Federal Ministry for Economic Affairs and Energy, Germany.

Within OPIRA, a collaboration of Airbus Defence and Space, German universities, and research institutes intends to develop holistic solutions to increase air traffic safety and the operation of RPAS for civil applications.

References

[1] M. Vodel, S. Blokzyl, W. Hardt, Hochauflösende Videodatenanalyse in Echtzeit - FPGA Hardwarebeschleunigung durch Daten- & Task-Parallelisierung, *Proceedings of the 2015 Innosecure 15 - Innovationen in den Sicherheitstechnologien*, April 2015; ISBN: 978-3-86359-299-8

[2] S. Blokzyl, W. Hardt, Generisches Framework zur Parallelisierung von Echtzeitfähiger Bilddatenauswertung für Rekonfigurierbare Integrierte Schaltkreise, *Proceedings of the Studierendensymposium Informatik 2016 der TU Chemnitz (TUCSI2016)*, 145-148 (March 2016); ISBN: 978-3-944640-85-3

[3] S. Blokzyl, W. Hardt, FPGA-basierte Hardwarebeschleunigung für Echtzeitbildverarbeitung und Fusion, *Proceedings of the Informatiksymposium Chemnitz 2012 (TUCSIS2012), Chemnitzer Informatik-Berichte CSR-12-01 (2012),* 102-105 (July 2012); ISSN: 0947-5125

[4] S. Blokzyl, M. Vodel, W. Hardt, A Hardware Accelerated Real-Time Image Processing Concept for High-Resolution EO Sensors, *Proceedings of the 61. Deutscher Luft- und Raumfahrtkongress (2012)*, September 2012; DOI: urn:nbn:de:101:1-201211239114

[5] S. Blokzyl, M. Vodel, W. Hardt, FPGA-based Approach for Runway Boundary Detection in High-resolution Colour Images, *Proceedings of the 2014 IEEE Sensors & Applications Symposium (SAS2014)*, 59-64 (February 2014); ISBN: 978-1-4799-2180-5

[6] XILINX, Product Specification DS 150 - Virtex-6 Family Overview, August 2015, Available online: http://www.xilinx.com/support/documentation/datasheets/ds150.pdf

[7] S. Blokzyl, M. Vodel, W. Hardt, Generischer Ansatz zur kantenbasierten Landebahndetektion in hochauflösenden Farbbildern, *Proceedings of the 62. Deutscher Luft- und Raumfahrtkongress (2013)*, September 2013; DOI: urn:nbn:de:101:1-201312099786

[8] Study Groups of ITU's Telecommunication Standardization, Information technology - Open Systems Interconnection - Basic Reference Model: The basic model, July 1994; DOI: http://handle.itu.int/11.1002/1000/2820

[9] U. Tudevdagva, S. Blokzyl, B. Battseren, E. Khaltar, The beginning of Unmanned Aerial Vehicle-based Inspection of HVTL, *Power Energy and Engineering Journal*, 23-27 (February 2017); ISBN: 978-99929-73-24-2