H2020-ECSEL-2017-2-783162
FitOptiVis

From the cloud to the edge - smart IntegraTion and OPtimisation Technologies for highly efficient Image and VIdeo processing Systems

Deliverable:
D5.2 - Components Release V1

Due date of deliverable: 31-05-2020
Actual submission date: 04-06-2020

Start date of Project: 01 June 2018 Duration: 36 months

Responsible: Carlo Sau (Università degli Studi di Cagliari)

Revision: draft

Dissemination level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Service	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

DOCUMENT INFO

Author

Author	Company	E-mail
Carlo Sau	UNICA	carlo.sau@unica.it
Luigi Raffo	UNICA	raffo@unica.it
Francisco Barranco	UGR	fbarranco@ugr.es
Pekka Jaaskelainen	TUT	pekka.jaaskelainen@tuni.fi
Jiri Kadlec	UTIA	kadlec@utia.cas.cz
Pablo Sanchez	UC	sanchez@teisa.unican.es
Marcos Martinez	TASE	marcos.martinezalejandro@thalesaleniaspace.com
Pablo Chaves	SCHN	pablo.chaves@se.com
Enrico Grosso	UNISS	grosso@uniss.it
Tero Santti	UTU	teansa@utu.fi
Guillermo Amat Gomariz	ITI	gamat@iti.es
Claudia Rinaldi	UNIVAQ	claudia.rinaldi@univaq.it
Giacomo Valente	UNIVAQ	giacomo.valente@univaq.it
Francesca Palumbo	UNISS	fpalumbo@uniss.it
Tiziana Fanni	UNISS	tfanni@uniss.it
Roman Cecil	UWB	rcecil@kky.zcu.cz
Raul Santos de la Camara	HIB	rsantos@hi-iberia.es
Stefano Delucchi	AITEK	stefano.delucchi@aitek.it
Luis Medina Valdés	7SOLS	luis.medina@sevensols.com
Lukas Marsik	CAMEA	l.marsik@camea.cz
Zaid Al-Ars	TUD	Z.Al-Ars@tudelft.nl
Katiuscia Zedda	ABI	katiuscia.zedda@abinsula.com
David Pampliega	SCHN	david.pampliega@se.com
Roman Cecil	UWB	rcecil@kky.zcu.cz
Rob de Jong	PHL	rob.de.jong@philips.com
Geran Peeren	PHL	geran.peeren@philips.com

Document history

Document version #	Date	Change
V1	28/02/2020	Added table of content and initial inputs from D5.1
V2	10/03/2020	Integrated contributions after reminder and Prague meeting
V3	18/03/2020	Integrated new contributions for second year advancements/updates and IP info
V4	16/04/2020	Integrated new contributions for second year advancements/updates and IP info
V5	29/04/2020	Integrated contributions from AITEK, 7SOLS, CAMEA, UNIVAQ, TUD
V6	06/05/2020	Integrated contributions from UGR, TUD, UTIA, HURJA

V7	11/05/2020	Integrated PHL and TASE contributions. Modified structure of the document
V8	13/05/2020	Integrated ABI contribution, aligned references, refined introduction and conclusions
V9	13/05/2020	Last refinements, draft ready for internal review
V10	29/05/2020	Modifications after internal review, deliverable ready for submission

Document data

Keywords	Hardware, software, networking
Editor Address data	Name: Carlo Sau Partner: Università degli Studi di Cagliari Address: Piazza d'Armi, 09123 Cagliari, IT Phone: +390706755778

Table of Contents

1 Executive summary

This deliverable reports the results of WP5 during the second project year.

D5.2 Components Release V1 – Report + IP - M24 – Task Involved: All – PU - UNICA
The preliminary set of innovative processing and communication components, including novel accelerators, are reported and delivered. All the study, development and assessment activities for each of the task are involved. HW and SW physical components implementations, along with their related documentations, are also provided. A subset of the developed IPs (according to the CA and FitOptiVis exploitation plan) will be released open source.

The list of identified components is reported, with a short analysis of their state of the art and commercial positioning (if available), as well as the plans for their deployment within the FitOptiVis project use cases. The partners identified the TRL level of their components at the beginning of the project, and the expected TRL level at the end of the project. Such TRLs differ a lot from one component to another, also considering the TRL expected by the addressed UC. On the top of such plan, the status of the components at Y2 discussed in depth. Overall, at the moment, components are under refinement or under integration within PoCs and demonstrators. These latter are expected to be delivered in time by the end of the project. Information about Intellectual Property (IP) of the components is also reported in this deliverable, to provide an overview of the related protection and distribution strategies.
All the three tasks of WP5 have started and are active at Y2, so that HW, SW and communication components are discussed. All 22 partners of WP5 contributed a total of 26 of their respective component developments. All use cases of the FitOptiVis project have been covered by the contributed component developments.
This deliverable is the natural evolution of D5.1 and, therefore, the structure has been kept the same, while inputs necessary to describe components are changed is some cases due to the same components progress. With respect to D5.1, two additional sections per component have been added:

- Section 2.1.1.1 – Updates during the second year of the project: to describe the progress of the component development made during the second year of the project (if any). Here updates, modifications and advancements of the time period this deliverable is referred to, are discussed in detail.
- Section 2.1.6 – Intellectual Property information: to provide plans or information related to the protection of the component and to possible distribution strategies (open source, evaluation or commercial licenses), if any.

2 Introduction

We refer to a component as a part of the computing platform or as a part of the application. The granularity of components that are going to be presented in this document is highly variable; some of them are atomic while others represent compositions of sub-elements. In the deliverable, we will refer to the following:

- HW IPs (Section 3) are intended as physical realization of processing units and application specific accelerators, described using a HW specification language (e.g. Verilog or VHDL). The SW code to be executed may not be necessarily defined and some of them could be highly use-case dependent.
- SW IPs (Section 4) are intended as an algorithm, described using a SW specification language (e.g. C, C++, MatLab, etc.), handling a specific processing issue. The final physical platform, where they will be executed, may not be necessarily defined and some of them could be highly use-case dependent.
- Communication-oriented IPs (Section 5) are intended as gateways and on chip traffic managers.
- Miscellaneous IPs (Section 6) are intended as components that are not HW, SW or Communication-oriented, like sensor/actuators -composed components.

Progressive refinements of the components, and their composing elements or sub-components, are envisioned during the project life cycle.

In the rest of this section we present the general structure that is going to be adopted in the rest of this document.

2.1 General template to be followed [involved partner(s)]

A sentence to introduce the component is required to quickly understand its meaning.

2.1.1 Short description

This section discusses the overall component description, following the topics below:

- Purpose and mission of the components (including the reference platform or software if already defined).
- List of its principal key features and properties.
- Processed inputs and produced outputs.
- Connectivity and communication.

2.1.1.1 Updates during the second year of the project

This section is meant to report the updates, modifications and improvements performed during the second year of the project on the specific component.

2.1.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

This section describes the way the components link to the expected reference architecture discussed in WP2. Specifically, the following aspects are discussed.
Qualities:
Intended as the tuneable features of the component, upon which a Pareto analysis should be doable in order to take design-time configuration decisions on the best possible set up. Some qualities may also be tuneable at run-time, when this happen proper explanations are provided in the run-time support section below.

Set-Point(s) and Parameters:
Tuneable HW/SW features to allow the possibility of surfing among qualities, which means offering different processing knobs.

2.1.1.3 Design-time support

This section describes the way the components link to the expected design-time support discussed in WP3. This refers to commercial or academic toolsets to be used to handle components design, analysis, verification and deployments.

2.1.1.4 Run-time support

This section describes the way the components link to the expected run-time support discussed in WP4. This refers to the following aspects
Run-time adaptivity support:
Description of what can be (re-)configured or tuned at runtime. Triggers for adaptation are also mentioned.
Monitoring:
Description of what is observed during system/components execution. Monitored data in some cases represent triggers for adaptation.
Programmability and Programming API:
Programmability support of the system/components.

2.1.2 Related state of the art

Brief discussion of the state of the art and explanation of what the innovation is brought by FitOptiVis.
Success stories (usage of this component in other contexts) may also be reported here for those components that are extended in the project.

2.1.3 Commercial positioning

Possible utilization of the component in a commercial product, and possible competitor products and explanation of what is the innovation brought by FitOptiVis.
Success stories (usage of this component in other contexts) may also be reported here for those components that are extended in the project.

2.1.4 Extension within FitOptiVis

This section discusses the starting TRL of each component at the beginning of the project and the expected TRL at the end of the project.
TRL@M0:
Comments on current TRL, explaining the positioning.
FitOptiVis Extensions:
Bullet list of what you expect to add/improve.
TRL@M36:
Comments on foreseen TRL, explaining the positioning.

2.1.5 Prospective adoption

- Explain other possible applications.

- Explain in which use case and how this component is going to be used in FitOptiVis

2.1.6 Intellectual Property information

This section is meant to provide all the information related to Intellectual Property and, eventually, open source features of the component.

2.1.7 Related documents

This section lists the references.

3 Processing and acceleration components

This section describes devices developed in FitOptiVis for customized acceleration of algorithms related to the use cases of the project.

3.1 Customized parallel soft-cores [TUT]

TTA-based Co-Design Environment (TCE) is an open application-specific instruction-set toolset. It can be used to design and program customized processors based on the energy efficient Transport Triggered Architecture (TTA). The toolset provides a complete retargetable co-design flow from high-level language programs down to synthesizable processor RTL (VHDL and Verilog backends supported) and parallel program binaries.

3.1.1 Short description

TCE co-processors as application-specific instruction-set processors (ASIP), which are tailored for an application, application-domain or targeted for a more general purpose use. Processor customization points include the register files, function units, supported operations, and the interconnection network.
Input:
TCE cores can have various input, depending on the application case they are tailored for. It can vary from single bit signals to data bundles transferred via wide SIMD load/store units.
Output:
Similarly, the output of a particular TCE co-processor varies according to its purpose.
Connectivity:
The connectivity depends on the system the co-processor is integrated. Typical system bus used is AXI, however, TCE cores can act as standalone processors or connected directly to ethernet interfaces. For example, customized I/O function units can be defined for hardware FIFOs or direct connections to digital-analog converters.
Composability:
TCE co-processors are meant for integrating into larger systems-on-chip (SoC) where there can be multiple co-processors or hardware accelerators, either TCE cores or IP blocks from another party. The cores itself are composed of function units and register files of which implementations can be automatically generated or pre-implemented and optimized ones can be picked from a hardware database.
Target:
TCE tools generate RTL (VHDL or Verilog). The produced RTL can be synthesized for FPGA fabrics as well as custom chips fabricated via an ASIC process.

3.1.1.1 Updates during the second year of the project

In the second year, the most visible update to the customized parallel soft-cores was heavy investment of effort to optimize the TTA-SIMD template where one can include wide vector operations in the function units. The RTL was adapted and optimized especially for FPGA soft core usage with a publication submitted to FPL 2020. On the other side, there were progress in the instruction compression mechanisms (based on dictionary compression) as well as static instruction caching mechanisms (instruction register files) of which two publications have been submitted.

3.1.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

The cores are typically customized to latency critical real-time tasks. The design-time customization points include register files, function units, operation sets, the interconnection network and number of cores. In SoC customization level the topology between co-processors is left to external tools with TCE generating integration points to the system level via e.g. AXI bus interfacing load-store units.
Run-time parameters are likewise design-specific: The co-processors are compiler programmable and thus the functionality can be changed at run-time via loading a new program and resetting the processor. The extent and quality of the code the compiler can produce to an application outside the original optimization target for the core at hand depends on the generality of the designed co-processor architecture.

3.1.1.3 Design-time support

Design-time support is performed with TCE tools. Open source releases of TCE are made available in http://openasip.org.

3.1.1.4 Run-time support

Re-programmability at run-time is supported. Additional monitoring support is developed during FitOptiVis expanding the work in ALMARVI, in which a common IP interface (AlmaIF) was developed that included break point interface and a cycle counter that was memory mapped [HSV+2019].
Programmability:

- Software compiler programmable
- Programming APIs: C, OpenCL C 1.2 + few features from OpenCL 2.0+

Monitoring:

- Cycle counters.
- Stall cycle counters.
- Instruction traces (planned in FitOptiVis).

3.1.2 Related state of the art

TCE is a state-of-the-art open source application-specific instruction-set processor (ASIP) toolset. There are several commercial ones available since 1980s, but no open source tool with such maturity and that is supported with regular releases that provide compiler upgrades etc. TCE was built as a successor of “MOVE tools” developed in TU Delft in 1990s [CM1991]. MOVE was also released as open source, but is currently unmaintained. TCE was rewritten from the scratch to include a modular object oriented C++ based code and to include improvements such as LLVM based compiler and frameworks for experimentation and research on various parts of the ASIP generation process and TTAs. TCE cores are known to have been used in commercial designs (information which is confidental) and multiple co-processor case studies in academic research projects.

3.1.3 Commercial positioning

N/A

3.1.4 Extension within FitOptiVis

In terms of technology readiness levels, since TCE is a wide toolset with extensive set of features, the produced TCE cores itself can be thought to be mature (TRL7) since they have been also commercially used. However, some features and new aspects are at TRL2 which will be refined within the project towards maturity.
Within FitOptiVis, there are plans to improve TCE co-processors at least in the following aspects:

- Optimizing TCE cores to FPGA realization; improved clock frequency and reduced resource usage: @TRL3, the goal is to get to TRL5.
- Streamlined use in FPGA soft core use: @TRL2, the goal is to get to TRL7.
- Improved compiler quality: ongoing, TRL depends on the feature being developed.

3.1.5 Prospective adoption

Will be considering for the virtual reality use case for accelerating real time photorealistic graphics rendering by means soft cores in an FPGA. One of the currently identified use cases for an ASIP / soft core is low latency video stream compression using texture compression algorithms.

3.1.6 Intellectual Property information

The TTA-SIMD template is not yet scheduled for open sourcing. However, regular TCE open source releases will be prepared in synchronization with LLVM releases.

3.1.7 Related documents

[HSV+2019] J. Hoozemans, J. van Straten, T. Viitanen, A. Tervo, J. Kadlec, Z. Al-Ars: “ALMARVI Execution Platform: Heterogeneous Video Processing SoC Platform on FPGA, Journal of Signal Processing Systems, 2019.

[CM1991] H. Corporaal, H. Mulder: “MOVE: A framework for high-performance processor design”, Supercomputing, 1991.

[TCE] TCE documentation. Online: http://openasip.org/documentation.html

3.2 Optimized HW/SW cores – 8xSIMD floating point accelerators [UTIA]

Component serves for run-time reprogrammable 8xSIMD single precision floating point computation.

3.2.1 Short description

Run-time reprogrammable 8xSIMD single precision floating point computation is performed inside of this component, that is shown in Figure 1. The component includes:

- Communication part used to receive and send data or program to Arm A9 on Zynq device or to Arm A53 on Zynq Ultrascale device via AXI-stream data interface.
- Computation part serving for execution of sequences of 8xSIMD single precision floating point operations defined in the program.

- Configuration part serving for setting up of operational mode via AXI-lite registers.

The 8xSIMD component works with

- 8 Single precision (SP) floating point (FP) units performing pipelined ADD operation which can be executed in each clock cycle.
- 8 SP FP units performing pipelined MULT operation which can be executed in each clock cycle.
- 8 SP FP units performing pipelined DIV operation which can be executed in each clock cycle.
- Auto increment address and program counter logic.
- Design time configurable set of SP FP data flow operations driven by predefined state machines.
- Logic enabling to execute programs programmed as sequence of SP FP data flow operations defined in program memory.

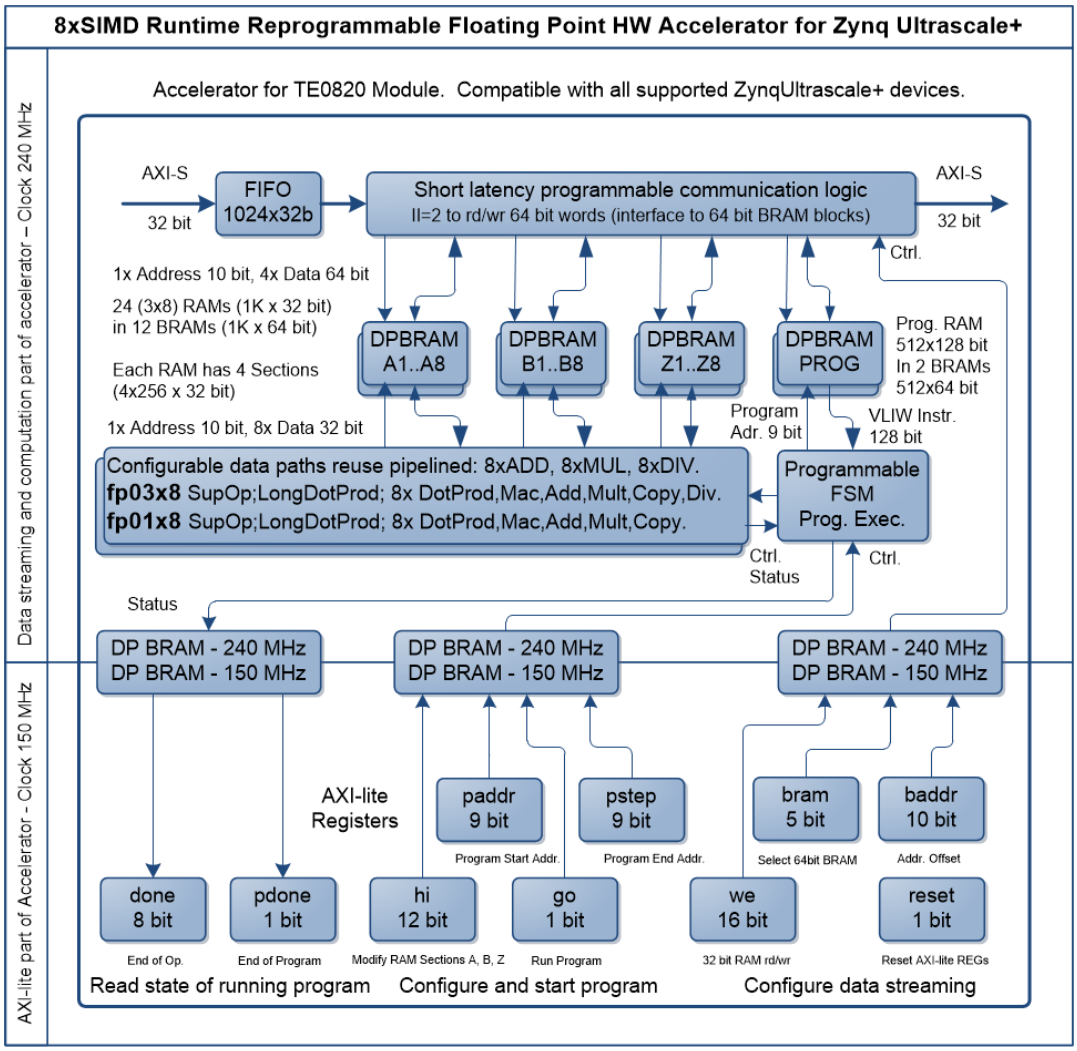


Figure 1 8xSIMD runtime reprogrammable floating point HW accelerator for Zynq Ultrascale+.

Input:

- Program firmware data received via AXI stream interface from Arm processor.

- Configuration Write registers for scalar control received via AXI-lite interface from Arm processor.
- Floating point single precision data received via AXI stream interface from Arm processor.

Output:

- Registers indicating end of program accessible to Arm processor via AXI-lite.
- Floating point single precision result data accessible via AXI stream interface for the Arm processor.

Connectivity:

- AXI stream input with input FIFO 2048x32 and support for the AXI stream side channel indicating the last transferred word sent to the component via the DMA transaction from Arm processor.
- AXI stream output with output FIFO 2048x32 bit with support for the output side channel indicating the last transferred word sent from the component via the DMA to Arm processor.
- AXI-lite input/output configuration registers.

Target:

- Zynq devices (Zynq 7000 28nm and Zynq MPSoC - UltraScale 16nm) with PL part and processor on single chip.

Communication with the accelerator is performed through different types of interface, summarized in Table 1.

Table 1 Types of interface of the 8xSIMD floating point accelerators.

type of interface	purpose	clock frequency on Zynq Ultrascale+	clock frequency on Zynq
AXI-S 32 bit	data streaming	240 MHz	115 MHz
8xSIMD_FP	computation	240 MHz	115 MHz
VLIW 128 bit	firmware program	240 MHz	115 MHz
AXI-Lite 32 bit		150 MHz	100 MHz

3.2.1.1 Updates during the second year of the project

In Y2 UTIA developed these versions of accelerators (see Table 2 and Table 3 for the complete instruction set of such accelerators):

- fp01x8 with capabilities 10, 20, 30 or 40
- fp03x8 with capabilities 10, 20, 30 or 40

for

- Zynq 7000 family of devices
- Zynq Ultrascale+ family of devices

Please note that each capability refer to a different set of SIMD OP supported by the accelerator, as specified in Table 2 and Table 3. Accelerators fp01x8 (all capabilities) do not support 8xSIMD floating point division, while accelerators fp03x8 (all capabilities) support 8xSIMD floating point division.

Table 2 Floating point functions present in all accelerators {10 or 20 or 30 or 40}.

SIMD OP	code (dec)	8xSIMD Floating Point Operation Description
VVER	0	Return capabilities of the accelerator and status of license
VZ2A	1	8xSIMD vector copy $a_m[i] <= z_m[j]$; m=1..8
VB2A	2	8xSIMD vector copy $a_m[i] <= b_m[j]$; m=1..8
VZ2B	3	8xSIMD vector copy $b_m[i] <= z_m[j]$; m=1..8
VA2B	4	8xSIMD vector copy $b_m[i] <= a_m[j]$; m=1..8
Auto-increments:		*Example: for (n=0;n<=CNT;n++){i=i+B_INC; j=j+A_INC;}*
VADD	5	8xSIMD vector add $z_m[i] <= a_m[j] + b_m[k]$; m=1..8
VADD_BZ2A	6	8xSIMD vector add $a_m[i] <= b_m[j] + z_m[k]$; m=1..8
VADD_AZ2B	7	8xSIMD vector add $b_m[i] <= a_m[j] + z_m[k]$; m=1..8
Auto-increments:		*Example: for (n=0;n<=CNT;n++){i=i+B_INC; j=j+A_INC; k=k+Z_INC;}*
VSUB	8	8xSIMD vector sub $z_m[i] <= a_m[j] - b_m[k]$; m=1..8
VSUB_BZ2A	9	8xSIMD vector sub $a_m[i] <= b_m[j] - z_m[k]$; m=1..8
VSUB_AZ2B	10	8xSIMD vector sub $b_m[i] <= a_m[j] - z_m[k]$; m=1..8
Auto-increments:		*Example: for (n=0;n<=CNT;n++){i=i+B_INC; j=j+A_INC; k=k+Z_INC;}*
VMULT	11	8xSIMD vector mult $z_m[i] <= a_m[j] * b_m[k]$; m=1..8
VMULT_BZ2A	12	8xSIMD vector mult $a_m[i] <= b_m[j] * z_m[k]$; m=1..8
VMULT_AZ2B	13	8xSIMD vector mult $b_m[i] <= a_m[j] * z_m[k]$; m=1..8
Auto-increments:		*Example: for (n=0;n<=CNT;n++){i=i+B_INC; j=j+A_INC; k=k+Z_INC;}*

Table 3 Floating point functions present in all accelerators {10 or 20 or 30 or 40}.

SIMD OP	code (dec)	8xSIMD Floating Point Operation Description
VPROD FP01, FP03: 30,40	14	8xSIMD vector products. $z_m[i] <= a_m'[j..j+nn]*b_m[k..k+nn]$; m=1..8; nn range 0..255
VMAC FP01, FP03: 20,30,40	15	8xSIMD vector MACs. $z_m[i..i+nn] <= z_m[i..i+nn] + a_m[j..j+nn] * b_m[k..k+nn]$; m=1..8; nn range 0..10
VMSUBAC FP01, FP03: 20,30,40	16	8xSIMD vector MSUBACs. $z_m[i..i+nn] <= z_m[i..i+nn] - a_m[j..j+nn] * b_m[k..k+nn]$; m=1..8; nn range 0..10
LONG_VPROD FP01, FP03: 40	17	Single long vector product . $z_m[i] <= ((a_1'[j..j+nn]*b_1[k..k+nn]+a_2'[j..j+nn]*b_2[k..k+nn]) + (a_3'[j..j+nn]*b_3[k..k+nn]+a_4'[j..j+nn]*b_4[k..k+nn])) + ((a_5'[j..j+nn]*b_5[k..k+nn]+a_6'[j..j+nn]*b_6[k..k+nn]) + (a_7'[j..j+nn]*b_7[k..k+nn]+a_8'[j..j+nn]*b_8[k..k+nn]));$ m=1..8; nn range 0..255
VDIV FP03: 10,20,30,40 FP01: not supported	20	8xSIMD vector Division. $z_m[i] <= a_m[j] / b_m[k]$; m=1..8
Auto-increments:		*Example: for(n=0;n<=CNT;n++){i=i+Z_INC; j=j+A_INC; k=k+B_INC;}*

Table 4 shows the detail of the 128 bit VLIW instruction, where the highlighting text indicates the hi 64 bit and low 64 bit. Each VLIW instruction is stored in DDR as two 64 bit words.

Table 4 Structure of the 128 bit wide VLIW program instruction.

FP01, FP03	Size	VLIW: hi lo	Description
[not_used]	*[8bit]*	*8 bit [63..56]*	*Not used by FP01 or FP03*
[not_used]	*[8bit]*	*8 bit [55..48]*	*Not used by FP01 or FP03*
[0,Z_MEM_SECTION]	[0,2bit]	8 bit [47..40]	Z_MEM SECTION (0..3)
[CNT]	[8bit]	8 bit [39..32]	Number of 8xSIMD steps (0 .. 255)
[Z_INC]	[8bit]	8 bit [31..24]	Auto increment of Z address (0 .. 255)
[Z_MEM_SADDR]	[8bit]	8 bit [23..16]	Set Z address after auto-increment overflow
[Z_MEM_ADDR]	[8bit]	8 bit [15..08]	Initial Z address
[B_INC]	[8bit]	8 bit [07..00]	Auto increment of B address (0 .. 255)
[OP]	[8bit]	8 bit [63..56]	8xSIMD vector operation
[0, B_MEM_SECTION]	[0,2bit]	8 bit [55..48]	B_MEM SECTION (0..3)
[0, A_MEM_SECTION]	[0,2bit]	8 bit [47..40]	A_MEM SECTION (0..3)
[B_MEM_SADDR]	[8bit]	8 bit [39..32]	Set B address after auto-increment overflow
[B_MEM_ADDR]	[8bit]	8 bit [31..24]	Initial B address
[A_INC]	[8bit]	8 bit [23..16]	Auto increment of A address (0 .. 255)
[A_MEM_SADDR]	[8bit]	8 bit [15..08]	Set A address after auto-increment overflow
[A_MEM_ADDR]	[8bit]	8 bit [07..00]	Initial A address

Acceleration of single precision floating point Matrix by Matrix multiplication has been prepared as an application example to evaluate the performance of released evaluation versions of accelerators. It performs: C[64,64] = A[64,64] * B[64,64].
A single instance of FP01x8 accelerator on ZynqBerry board accelerates A9 processor (with 650 MHz clock) computation of this single precision floating point Matrix by Matrix multiplication by factor of 5x.
Two serially connected instances of FP03x8 accelerator on Zynq Ultrascale+ family of devices with evaluation package for Zynq Ultrascale+ industrial module TE0820-04EV-1E (1.2 GHz clock) on TE0701 carrier board accelerate the A53 processor computation of two single precision floating point Matrix by Matrix multiplications by factor of 5x.
Two parallel connected instances of FP03x8 accelerator on Zynq Ultrascale+ family of devices accelerate the A53 processor computation of two single precision floating point Matrix by Matrix multiplications by factor of 5x.
Details of these evaluation examples can be found in D3.2 deliverable.

3.2.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Design time configurable set of SP FP data-flow operations driven by predefined state machines. Some examples are:

- 8xSIMD SP FP Vector ADD or SUB or MUL or DIV operations
- 8xSIMD SP FP MAC (chained multiply and add operation)
- 8xSIMD SP FP vector by vector dot product operation.

Firmware for the component is defined in the runtime in ARM processor. User defines sequence of available SP FP data-flow operations to achieve its goal. Some examples are:

- SP FP matrix multiplication can be implemented by several firmware programs, which can be loaded in the runtime to the component. Programs can be optimized to the matrix sizes to get best performance in terms of MFLOPS for

the currently processed matrices. Program can be composed from 8xSIMD SP FP MAC (chained multiply-add operation) or program can utilize the 8xSIMD SP FP vector by vector dot product operation.

- In case of component without Vector by vector support and without the MAC operation, (such component has significantly reduced requirements for FPGA logic), the 8xSIMD SP FP Vector ADD and MUL operations can be used to perform the matrix multiplication, with reduced performance.

3.2.1.3 Design-time support

The design time support of the 8xSIMD floating point accelerators is presented in [KPK2020]. It describes the design flow for automated generation of data streaming HW (data movers):

- Zero Copy (HW data mover without DMA unit)
- DMA (HW data mover with DMA unit)
- SG DMA (HW data mover with interrupts)

In particular UTIA DTRiMC tool serves for automatic generation of DMA data movers for the component for Arm A9 SoC systems and for Arm A53 Ultrascale+ SoC systems, while UTIA DTRC tool serves for automated generation of Petalinux kernel, Debian OS with Xilinx SDSoC compiler support and Xilinx Vivado HLS compiler support.
This design time support is based on the Xilinx SDSoC 2018.2 system level compiler. D3.2 provides also an overview of the released design time resources.

3.2.1.4 Run-time support

UTIA develops API supporting Arm A9 SoC systems and for Arm A53 Ultrascale+ SoC systems running Debian OS with:

- API functions download of firmware programs to the component
- API functions for DMA of SP FP data to and from the component.
- It is using one of these three supported HW interfaces to the component, (these HW alternatives are generated in design time by Xilinx SDSoC 2018.2 compiler:
  - DMA HW (Arm processor API is waiting for end of DMA by pooling)
  - SG DMA HW interrupt-based API.
  - Zero Copy. It is using data mover generated by Xilinx HLS. (Arm processor API is waiting for end of DMA by pooling)
- Monitoring:
- Percentage of load of the AXI-Stream
- Cycle counters indicate ARM clock cycles needed for given task executed in the component

Programmability and Programming API
Firmware is re-programmable in run-time by data streaming. Computation & data streaming can be performed in parallel. See WP4 run-time support. See D4.2 and D4.3 for description of this run-time support.
Several AXI-lite registers can be controlled by the ARM application:

- reset, 1 bit: “1” Reset AXI lite Registers; “0” NOP
- we, 16 bit: Write from stream to blocks 0 .. 13
- baddr, 10 bit: Stream will rd/wr from addr=baddr
- bram, 5 bit: Read from Block 0 .. 13 to stream
- paddr, 9 bit: Program start address
- pstep, 9 bit: Program stop address
- go, 1 bit: “1” go from paddr to pstep; “0” NOP
- hi, 12 bit: SubBank prog. mod: 00zz00bb00aa (bits)

- done, 8 bit: Read only. “0” => Instruction runs
- pdone, 1 bit: Read only. “0” => Program runs

The accelerators have two main kinds of memories:

- 12 dual-ported 1024x64 bit BRAMs Blocks (0 .. 11) are used as 24 Data RAM 1024x32 bit A1..A8, B1..B8 and Z1..Z8 (see Table 5)
- 2 dual-ported 512x64 bit BRAMs Blocks (12, 13) are used as Program RAM 512x32 bit P1..P3 (see Table 5)

Table 5 Internal block rams of accelerators.

SIMD A 32 bit	Block 64 bit	SIMD B 32 bit	Block 64 bit	SIMD Z 32 bit	Block 64 bit	VLIW Prog	Block 64 bit
A1	0	B1	4	Z1	8	P1	12
A2		B2		Z2		P2	
A3	1	B3	5	Z3	9	P3	13
A4		B4		Z4		P4	
A5	2	B5	6	Z5	10		
A6		B6		Z6			
A7	3	B7	7	Z7	11		
A7		B8		Z7			

The stream data from/to ARM DDR memory has the following features:

- Maximal data streaming block is 2048 x 32 bit
- Data streaming block can have variable size: Min 2 x 32bit; Max 2048 x 32 bit
- Mode of operation (same for Data/Program):
- Write to a block defined by we from address baddr
- Broadcast Write by more bits in we (from baddr)
- Read from block bram from address baddr
- Write or Broadcast Write and Read in parallel
- See through the Accelerator we = 0; bram =16.

The ARM has several SW API for such streaming of data. Calls are unblocking, trigger HW threads. HW threads synchronize with SW in blocking `sds_wait()`. Calls are similar to the pthread `barrier()`. See D4.2 and D4.3 for description of this run-time support.

An example of serial streaming SW API for single accelerator and for single serial chain of multiple chained accelerators can be (len is the number of 32 bit words in the stream):

```
data2hw_wrapper((unsigned*)src_P1_P2, len);       //1
capture_wrapper((unsigned*)dest_P1_P2, len);      //2
…
sds_wait(1);
sds_wait(2);
```

While an example of parallel streaming SW API for two parallel accelerators and for two serial chains of several chained accelerators can be:

```
data2hw_wrapper( (unsigned*)src_P1_P2,  len);     //1
data2hw1_wrapper((unsigned*)src_P1_P2,  len);     //2
capture_wrapper( (unsigned*)dest_P1_P2, len);     //3
capture1_wrapper((unsigned*)dest_P1_P2, len);     //4
…
sds_wait(1);
sds_wait(2);
sds_wait(3);
sds_wait(4);
```

3.2.2 Related state of the art

The accelerators can be compared with these alternative solutions:

- EDKDSP family of accelerators developed by UTIA in frame of ECSEL projects Almarvi [EDKDSPa], [EDKDSPb] and EMC2 [EDKDSPc].
  (+) The EDKDSP accelerators where limited to Artix, Kintex and Zynq 7000 family of devices, due to the used design flow. Design flow selected for the 8xSIMD accelerators can target also the 16nm Zynq Ultrascale+ devices.
  (+) The EDKDSP accelerators have used for data communication with Arm DDR memory the Xilinx MicroBlaze soft-core with data and instruction cache. The data transfer was slow and this have often negatively impacted the acceleration results and the scalability (number of instantiated EDKDSP 8xSIMD accelerators managed by the same MicroBlaze soft-core). These limitations of EDKDSP accelerators are solved in released 8xSIMD accelerators by the data streaming interface and the auto-generated HW data movers.
  (-) The automated generation of HW data movers for 8xSIMD accelerators requires in design time the Xilinx SDSoC 2018.2 compiler license.
  (+) User of 8xSIMD accelerators compiles the C++ top level of the application in standard g++ compiler and does not need the Xilinx SDSoC 2018.2 compiler license for compilation, or for execution of the compiled code.
  - TTA-based Co-Design Environment (TCE) is an open application-specific instruction-set toolset [TCE]
  (-) TCE based processors have much stronger support for optimized C compiler.
  (-) TCE based processors have OpenCL API implemented in the pocl framework, and this is not supported at this stage by the released 8xSIMD accelerators.
  (+) 8xSIMD accelerators support fast streaming data interfaces to arm A9 or A53 processors which are not supported at present by TCE based processors.
  (+) 8xSIMD accelerators support overlapped (parallel) streaming data transfer with computation on the accelerator
  (+) 8xSIMD accelerators are supported by the auto-generated HW data movers supporting HW ZeroCopy, DMA and SG DMA with interrupt based data transfer infrastructure compatible with Debian OS common simple API.
  (+) 8xSIMD accelerators firmware can be defined and compiled in runtime SW app with compilation by “make” and g++ compiler directly on the target embedded device (ZynqBerry or Zynq MPSoC – UltraScale module TE0820).

3.2.3 Commercial Positioning

The commercial version of accelerators is available starting from 1.4.2020. UTIA will offer this license on commercial base. Contract has to be signed with UTIA. For information about details of the commercial license write to Jiri Kadlec kadlec@utia.cas.cz.

3.2.4 Extension within FitOptiVis

TRL@M0: TRL4 SDSoC support and results from Almarvi, but only with slow AXI-lite interface and only for standalone Zynq processor without Linux (Debian) OS.
TRL@M36: TRL6 Support of integration of the component for 64bit Zynq UltraScale with PetaLinux and Debian file OS. Support by UTIA DTRC tool and DTRiMC tool.

3.2.5 Prospective adoption

This component will be used as HW accelerator for calibration of robot arm based on video processing.

3.2.6 Intellectual Property information

In Y2 UTIA released these evaluation versions of accelerators:

- fp01x8 capabilities = 40
- fp03x8 capabilities = 40

for

- Zynq 7000 family of devices with evaluation package for ZynqBerry board.
- Zynq Ultrascale+ family of devices with evaluation package for Zynq Ultrascale+ industrial module TE0820-04EV-1E on TE0701 carrier board

The license for the evaluation versions of accelerators enables execution of certain large number of floating point operations before it expires. If this happens, the board has to be switched off and switched on again to restart the evaluation license again. Starting from 1.4.2020, the evaluation versions of accelerators can be publicly downloaded for free from UTIA www page: http://sp.utia.cz/index.php?ids=projects/fitoptivis
The commercial version of accelerators is available starting from 1.4.2020. UTIA will offer this license on commercial base. Contract has to be signed with UTIA. For information about details of the commercial license write to Jiri Kadlec kadlec@utia.cas.cz

3.2.7 Related Documents

[KPK2020] J. Kadlec, Z. Pohl, L. Kohout, “Two serial connected evaluation versions of FP03x8 accelerators for TE0820-03-4EV-1E module on TE0701-06 carrier board”, Application Note, 2020.
[EDKDSPa] Online: http://sp.utia.cz/index.php?ids=results&id=t20i2pm4
[EDKDSPb] Online: http://sp.utia.cz/index.php?ids=results&id=t30e3hm4
[EDKDSPc] Online: http://sp.utia.cz/index.php?ids=results&id=s30i1hm4
[TCE] TCE repository. Online: https://github.com/cpc/tce

3.3 NEURAghe [UNICA]

NEURAghe defines a component for accelerating neural networks execution on programmable heterogeneous platforms.

3.3.1 Short description

NEURAghe is an accelerator for Convolutional Neural Network (CNN) based on the Xilinx Zynq SoC. It fully exploit heterogeneity of Zynq platforms (see Figure 2): in the programmable logic a Convolution-Specific Processor (CSP) with a convolution engine and a programmable soft-core take care of the bulk of CNN workload; while the ARM General Purpose Processor (GPP) is in charge of executing hard-to-accelerate parts of the computational graph without the need of supervising acceleration thanks to the CSP soft-core. An overview of the system architecture is depicted in the figure below, where separation between GPP and CSP is highlighted.
The key features of NEURAghe are related to some critical aspects of CNN execution:

- support the deployment of arbitrary CNN topologies,
- acceleration of critical compute-bound operations (i.e. convolutional layers),
- hiding of memory-bound operations (i.e. fully connected layers) by overlapping them with the compute-bound ones.

NEURAghe offers to the developer several degrees of freedom.

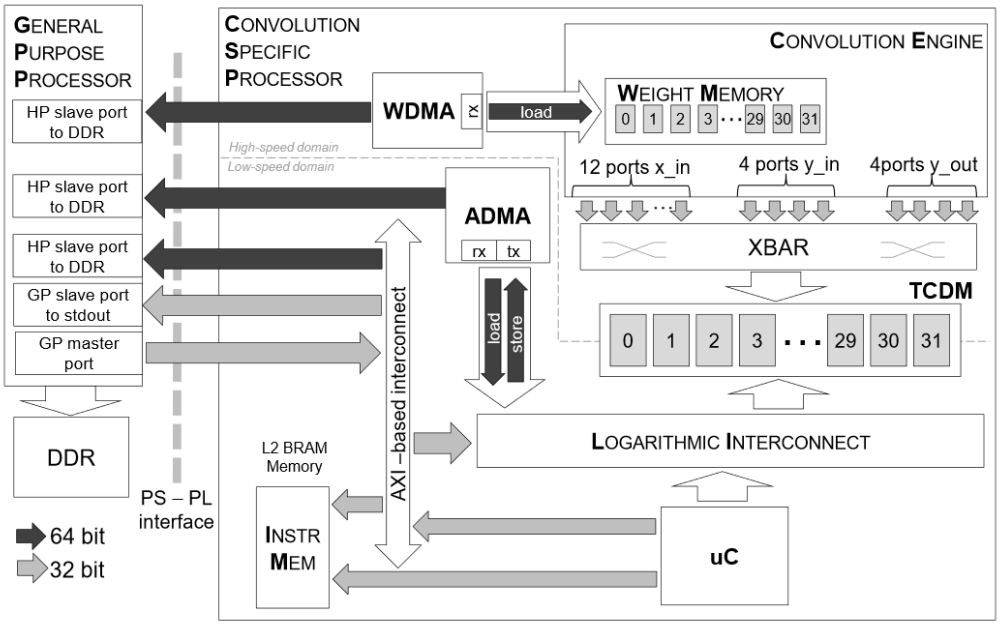


Figure 2 NEURAghe architectural template.

Input:
Being accelerating CNN, the input of the system is data (images, sound, sensor data) under recognition/ classification/ analysis.
Output:
The output is a classification/detection response.
Target:
Indeed, besides the CNN to be accelerated (it is generic in this sense), it is possible to target a different Xilinx Zynq SoC device according to the context. In particular,

NEURAghe is already compliant with Z7010, Z7010, Z7045 and Ultrascale+, while support for new platforms can be envisioned during FitOptiVis, if needed.
Connectivity:
The connectivity depends on the specific board selected for development. It may be ethernet-based or wireless-based for smaller nodes. The board can be equipped with sensors and cameras by means of the native I/O interfaces.
Composability:
Multiple NEURAghe instances can be composed in a single system using the previously mentioned connectivity. Multiple CSPs can be implemented on the same chip if this may fit the use-case.

3.3.1.1 Updates during the second year of the project

Different improvements have been done during the second year of the project. On the architecture side, we have implemented a flexible memory addressing module that is needed to support free strides and dilations, especially in Temporal Convolutional Networks, which are used in sequence recognition/classification and in generative audio/streaming tasks. Moreover, we have changed the architectural configuration of memory modules and of multiply-and-accumulate blocks, to support arbitrary kernel sizes.
On the software side, we have worked on the programming support. We have created a python tool, called ONNX2Neuraghe that translates a ONNX network description into a C code, instrumented with calls to NEURAghe APIs, that is ready to be compiled for the ARM processors in the Zynq or Zynq Ultrascale+ processing system.
Finally, we have worked on the porting on Zynq Ultrascale+ devices, to support more recent reconfigurable platforms and all-programmable SoCs.

3.3.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Qualities:

- Gops/s: number of Giga operations per second, depending on the data size and working frequency,
- Gops/w: number of Giga operations per second, depending on the data size and working frequency,
- Accuracy: precision versus performance trade-off can be exploited changing CNN algorithm at runtime.

Set-Point(s) and Parameters:

- network configuration: CNN to be executed and accelerated with NEURAghe,
- data size (8-16 bit): number of bits for fixed point numbers used to process the CNN,
- working frequency: clock frequency of the accelerator (arbitrary).

3.3.1.3 Design-time support

Design time support tools and methods:

- Xilinx Vivado (later than 18.1): commercial tool necessary for the deployment of NEURAghe, properly configured according to the application context, on the target Xilinx Zynq SoC,
- Python scripts: academic set of scripts used for CNN (ONNX) to C conversion to map the desired CNN on a NEURAghe deployed system.

3.3.1.4 Run-time support

Run-time adaptivity support:

- network configuration: CNN to be executed and accelerated which can be changed at run-time among the CNNs that have been generated (Python scripts conversion) at design-time,
- data size (8-16 bit): number of bits for fixed point numbers can be changed at run-time according to the desired Gops/s, Gops/w and accuracy,
- working frequency: clock frequency of the system can be tuned at run-time according to the desired Gops/s and Gops/w.

Monitoring:

- clock cycles: it is possible to measure clock cycles needed for execution of the CNN layers and use them to trigger adaptation, e.g. working frequency.

Programmability and Programming API:

- network configuration (CNN to be accelerated) can be changed by software to any already generated corresponding C description,
- NEURAghe is provided with custom low level C API that are used by the GPP to delegate computation to be accelerated on the CSP.

3.3.2 Related state of the art

Due to the intrinsic parallelism in convolutions, FPGAs are a very promising target technology for the implementation of hardware accelerators for different kind of neural networks. Exploiting dedicated hardware, FPGAs allow realizing very flexible architectures. They can integrate a high number of parallel Digital Signal Processing (DSP) units, enabling efficient implementation of MAC operations, delivering high operational throughput at limited clock frequencies. Moreover, FPGAs offer a significant amount of memory resources, which can be used to map efficient temporary storage buffers to store partial results of the convolutions, and other hardware primitives, such as registers and look-up tables (LUTs), suitable to be used to implement the glue logic around the accelerators.
Thanks to the extensibility added within FitOptiVis NEURAghe is currently outperforming all existing architectures [M2018] on Z-7045 using 8-bit and 16-bit data (delivering up to 335 GOps/s or 173 GOps/s, respectively), achieves state-of-the-art performance on Z-7020 using 8-bit data (up to _85 GOps/s), and can be used even on tiny devices such as Z-7007s [MLD+2019].
In FitOptiVis, we extend the NEURAghe template-based architecture to guarantee design-time scalability to multi-processor SoCs with vastly different cost, size and power envelope such as Xilinx's Z-7007s, Z-7020 and Z-7045.

3.3.3 Commercial positioning

N/A

3.3.4 Extension within FitOptiVis

TRL@M0:
TRL3: NEURAghe proof of concept has been already demonstrated in lab.
FitOptiVis Extensions:

- Improving efficiency, in terms of Gops/s and Gops/w, and implementability in different SoCs
- Exploration of new operating points for data size (e.g. 4 bit accuracy)
- Extension to less mainstream CNN patterns

TRL@M36:
TRL4: NEURAghe will be validated in laboratory.

3.3.5 Prospective adoption

NEURAghe will be exploited in water supply use case and, in particular, it will be adopted to improve AITEK video processing software modules, that are now based on traditional foreground and background separation. Here CNN algorithms can bring enhanced performance especially for tasks related to target classification, such as single shot object detectors where it is required to detect and classify multiple objectives within a video frame.

3.3.6 Intellectual Property information

The NEURAghe architecture and development environment is already available through an open source repository (https://github.com/neuraghe/NEURAghe). In particular, the repo contains all the source code, hardware and software, necessary to run CNN applications on Zynq or Zynq Ultrascale+ platforms taking advantage from the NEURAghe CNN hardware accelerator. It involves also initial documentation material mainly intended for installation and getting started with the accelerator and related supporting software. A website, proper documentation material and tutorials to facilitate the usage and the diffusion of NEURAghe technology will be set up in the next months. Within the repository, the last version of NEURAghe comprehensive of latest improvements, as described in Section 2.1.1.2, can be found under the neuraghe_2.0 folder.

3.3.7 Related documents

[M2018] S. Mittal, “A survey of FPGA-based accelerators for Convolutional Neural Networks”, Neural Computing and Applications, 2018.

[MLD+2019] P. Meloni, D. Loi, G. Deriu, M. Carreras, F. Conti, A. Capotondi, D. Rossi, “Exploring NEURAGHE: A Customizable Template for APSoC-based CNN Inference at the Edge”, IEEE Embedded System Letters, 2019.

3.4 AIPHS [UNIVAQ]

In recent years, embedded applications have been characterized by increasingly stringent requirements, both from functional and non-functional point of view. This led to the adoption of complex hardware platforms (multi-core and many-core architectures), able to guarantee high computational power with low energy consumption and reduced footprint. An efficient characterization of these platforms can be problematic, given the large number of hardware resources and the complexity of software applications.

3.4.1 Short description

AIPHS, acronym of *AdaptIve Potential Hardware profiling System*, is a library of elements to compose hardware monitoring systems for reconfigurable architectures, in order to support on the characterization of platforms from the point of view of different metrics. It is written in VHDL and it allows developing hardware monitors for different architectures. The goal is to provide the basic elements to perform a low intrusive monitoring action.

HW Type:

MicroBlaze, ARM Cortex A9 and Leon3 processors have been targeted with AIPHS.
Target:
FPGA, VHDL.
Input:
There are two interfaces: initialization/collection results interface and monitoring interface. The former receives commands and provides results, while the latter receives low-level transactions.
Output:
Measures related to selected metrics. The output is parsable in Common Trace Format using a Python script.

3.4.1.1 Updates during the second year of the project

Different improvements have been done during the second year of the project. The framework has been adapted to monitor accelerators produced with the support of MDC. AIPHS has now a graphical interface to interact with the user, allowing the introduction of the monitoring system in the general system, providing three new metrics: the number of exchanged messages with the accelerators, the latency of the acceleration, and the runtime verification of accelerator properties. Detailed information is provided in the D4.3 deliverable.

3.4.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

N/A

3.4.1.3 Design-time support

AIPHS requires tool to implement HW architectures on FPGA and Python 3 interpreter to parse results in Common Trace Format. Starting from them, it allows obtaining a custom monitoring system that can be connected to different on-chip interfaces. Supported interfaces depend on the content of some libraries that can be updated with new ones.

3.4.1.4 Run-time support

Programmability:
Programming APIs available

- Hardware description:
  - Technology: VHSIC Hardware Description Language (VHDL)
  - Version: IEEE 1076-2008
- Interaction with monitoring system with embedded Linux:
  - Technology: Native POSIX Threads Library (NPTL)
  - Version: 2.19
- Parse of trace data:
  - Technology: Common Trace Format (CTF)
  - Version: v1.8.2

Monitoring:

- Number of read and/or write operations in a given period of time for a specified memory area for a specified memory.
- Elapsed time between two memory accesses on a specified memory location for a specified memory, expressed in number of clock cycles.
- Total number of cache hits and misses in a given period for a specified cache memory.

- Number of transactions occurring during a specified monitoring period over a monitored bus. A transaction is defined depending on the monitored bus.
- Size of data transferred over a monitored bus during a specified monitoring period.
- Amount of time required for burst data transfer, expressed in number of clock cycles. The definition of burst data transfer is implied by the selection of the targeted bus.
- Amount of time required for a single transaction execution. A transaction is defined depending on the monitored bus.
- Time required from address generation to the first service of data in a single bus transaction, expressed in number of clock cycles. A transaction is defined depending on the monitored bus.
- Overall run-time of specific parts of applications (indicated by means of segments of code) executed on a single core. Support for multi-core is work-in-progress.
- Percentage of instruction covered by a set of runs.

3.4.2 Related state of the art

A monitoring system can be useful in different applications, such as support for software developers to debug systems obtained through HLS tools [GW2017]. Debug of extra-functional properties is required and there exist tools that allow it for the performance in reconfigurable logic scenarios [SMD+2015]. In ASIC processors, there are various example of smart monitoring solutions: AMD Lightweight profiling feature [AMD] and Intel Processor Tracing [INTEL] are composed of hardware facilities (such as hardware performance counters) and software able to use information acquired at low-level. They are supported by necessary libraries to interpret collected data. In Leon3 based scenarios, Nam et al. [HKP2014] proposed a performance monitoring unit integrated with perf_event API. Xilinx supports system level profiling using SDSoC environment [XILINXa], using performance counters in ARM Cortex A9 and performance monitoring units in programmable logic side. Moreover, it offers profiling solutions also in MicroBlaze soft-processor [XILINXb]. In order to define a custom profiling system for embedded applications, a solution based on specific metric definition and implementation of necessary parts has been considered [MFV+2015]. This technique conducted to a definition of a library of elements, to be used to compose a hardware profiling system tailored for specific applications [VMP+2016].

3.4.3 Commercial positioning

N/A

3.4.4 Extension within FitOptiVis

The future way to perform monitoring solutions is based on hardware/software collaborative approaches where hardware implements the profiling action (profiling hooks and mechanisms) and software implements the functionalities that use collected data (algorithms, complex heuristics, etc.). The proposed extension within FitOptiVis focuses on different points.
The first is the possibility to tailor and customize the monitoring system for the system under exam, considering also accelerators: the customization will depend on when to use the monitoring action (i.e. during the lifecycle to characterize the system or during development phases to support the designer). It will depend on platform selected for the

system (ASIC, reconfigurable logic). And it will depend on non-functional properties of the system itself (how much overhead can be introduced with the monitoring system, if a real-time profiling action is requested, etc.).
The second is the development of a framework able to support the designer on the selection of profiling solution.

3.4.5 Prospective adoption

The water supply (Use Case 1) will be the FitOptiVis UC adopting this component.

3.4.6 Intellectual Property information

The AIPHS architecture and development environment will be made available through an open source repository at M24 under MIT license.

3.4.7 Related documents

[GW2017] J. Goeders, S. J. E. Wilton, "Signal-Tracing Techniques for In-System FPGA Debugging of High-Level Synthesis Circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2017

[SMD+2015] L. Shannon, E. Matthews, N. Doyle, A. Fedorova, “Performance monitoring for multicore embedded computing systems on FPGAs,” 2nd Int. Workshop FPGAs for Software Programmers, 2015

[AMD] AMD Lightweight profiling specification. Online: https://developer.amd.com/resources/43724-2/

[INTEL] Intel, “Intel 64 and IA-32 Architectures Developer's Manual”.

[HKP2014] N. Ho, P. Kaufmann, M. Platzner "A hardware/software infrastructure for performance monitoring on LEON3 multicore platforms," International Conference on Field Programmable Logic and Applications, 2014.

[XILINXa] Xilinx SDSoC Development Environment. Online: https://www.xilinx.com/products/design-tools/software-zone/sdsoc.html

[XILINXb] Xilinx MicroBlaze processor reference guide (v 2015.4). Online: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_4/ug984-vivado-microblaze-ref.pdf

[MFV+2015] A. Moro, F. Federici, G. Valente, L. Pomante, M. Faccio, V. Mutillo, “Hardware Performance Sniffers for Embedded Systems Profiling,” Workshop on Intelligent Solutions for Embedded Systems, 2015.

[VMP+2016] G. Valente, V. Mutillo, L. Pomante, F. Federici, M. Faccio, A. Moro, S. Ferri, C. Tieri, "A Flexible Profiling Sub-System for Reconfigurable Logic Architectures," 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, 2016.

3.5 High Performance Space Image Collection and Processing [TASE]

The High Performance Space Image Collection and Processing component developed by TASE consists on the main interface between a CMOS Sensor and the video processing chain with a preliminary processing performed by this component. This component will be in charge of collecting the images coming from the CMOS Sensor and make a pre-processing of them in order to make them usable for the video-processing chain.

3.5.1 Short description

The Image Collection Interface will be a hardware IP developed in FitOptiVis in order to obtain images from the CMOS sensor IMX274 on the LI-IMX274MIPI-FMC [LIIMX] board provided by Leopard Imaging. The IMX274 sensor is produced by Sony and has an active pixel array of 3684Hx2196V. The component will also be in charge of doing some pre-processing related with image.

Input**:**

- MIPI CSI-2 at 1.440 Gbps/Lane with 10/12-bit/pixel output at 30/60fps

Output:

- Output Data Format configuration interface (bits per pixel and data rate)
- Image data

Target Technology:

- Main target: Zynq UltraScale+
- Side target: UltraScale and UltraScale+ FPGA's families

3.5.1.1 Updates during the second year of the project

This component was originally thought to work with the CMV12000 sensor. Which is a High-Performance sensor with an output of up to 300 fps. For using this sensor, an *ad hoc* board was developed for interfacing with this sensor. After several months of development within the FitOptiVis project, several signal integrity issues were found. These issues made that the target sensor changed from the CMV12000 to the LI-IMX274MIPI-FMC by Leopard Imaging. The development of the High Performance Space Image Collection and Processing is currently on-going with some delay due to the switch on the target sensor that required to start again the development of the component very recently.

3.5.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Readout mode:

- 0: All pixels are readout with 12-bit output. Max frame frequency: 32.81 fps
- 1: All pixels are readout with 10-bit output. Max frame frequency: 62.52 fps
- 2: Horizontal and vertical direction 2-line weighted binning readout of pixels of the same colour at the all-pixel scan area, 12 bit/pixel. Max frame frequency: 65.82 fps
- 3: Horizontal and vertical direction 2-line weighted binning readout of pixels of the same colour at the all-pixel scan area, 10 bit/pixel. Max frame frequency: 124.80
- 5: 2 of every 3 lines in the vertical direction at the all-pixel scan area are added. Then, 3 pixels of the same colour in the horizontal direction are added and output. Max frame frequency: 185.73
- 6: 2 of every 8 lines in the vertical direction at the all-pixel scan area are added. Then, 3 pixels of the same colour in the horizontal direction are added and output. Max frame frequency: 185.73

3.5.1.3 Design-time support

Design time support tools and methods: Xilinx's Vivado (2018.1 or later) commercial tool used in order to develop the IP and to instantiate it.

3.5.1.4 Run-time support

Run Time Adaptivity:
The IP can change the configuration of the image collection on run-time being able to change the quality of the obtained images.
Programmability:

- Programming APIs:
  N/A
- Reconfiguration:
  - The IP could be reconfigured in case of malfunction due to failures induced by radiation conditions present in space and affecting SRAM Configuration Memory of the FPGA.

Monitoring:

- Selected Configuration.
- Watchdog for the camera.
- Power Measurement: The energy consumption of this block can't be determined but the changes on the power that the different configurations produce on the aggregate will be tracked.

3.5.2 Related state of the art

CCD sensors, which are much more expensive to manufacture than CMOS sensor dominate the space industry in earth observation, astronomy and exploration [M2015]. This is because is a mature technology and with excellent electro-optical performance. On the other hand, CMOS sensors are considered the future by the European Space Agency that is investing in order to develop a European CMOS foundry industry for development of high performance sensors.

3.5.3 Commercial positioning

N/A

3.5.4 Extension within FitOptiVis

TRL@M0:
TRL1
FitOptiVis Extensions:
Full development of the module on several Xilinx architectures
TRL@M36:
TRL6

3.5.5 Prospective adoption

The IP will be used in the Use Case 10 of this project.

3.5.6 Intellectual Property information

This component will be based on the integration of several sub-components provided by Xilinx which are commercial IPs. The integration of all of them will be distributed as Open-Source but a commercial license for some of the sub-components will be needed.

3.5.7 Related documents

[LIMIX] Leopard Imaging "LI-IMX274MIPI-FMC", Rev 1.3, [Revised March 2020.

[M2015] Kyriaki Mnoglou, "CMOS Image Sensor developments supported by the European Space Agency", EIRO Forum Topical Workshop: CMOS Sensors, 2015.

3.6 Image transmission interface [TASE]

The Image Transmission Interface Hardware component by TASE is in charge of taking the images produced by the High-Performance Space Image Collection and Processing and to send them through different interfaces in order to make them available for the different parts of the whole video-processing system.

3.6.1 Short description

An *ad hoc* IP will be developed in order to transmit the data produced by the component described in Section 3.5 and is able to send them out of the system through different interfaces: HDMI (to a display, for demonstration purposes) and Ethernet/UDP (in order to send the images to an external board with a CNN that will process the images).

Input:

- Compressed 1080p images.

Output

- HDMI video stream
- UDP video stream

3.6.1.1 Updates during the second year of the project

The development of this component has been postponed to the third year of the project.

3.6.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

This block won't have any tuneable set-point nor run-time parameters.

3.6.1.3 Design-time support

Design time support tools and methods:

- Xilinx's Vivado (2018.1 or later) commercial tool used in order to develop the IP and to instantiate it.

3.6.1.4 Run-time support

Run Time Adaptivity:
No runtime adaptivity
Programmability:

- Programming APIs:
  - N/A
- Reconfiguration:
  - The IP could be reconfigured in case of malfunction due to failures induced by radiation conditions present in space and affecting SRAM Configuration Memory of the FPGA. This reconfiguration is done through

an external scrubber. The goal of reconfiguring the component is not changing its functionality but healing the component from malfunctioning due to the accumulation of bitflips on the configuration memory produced by radiation.

Monitoring:

- Power Measurement: The energy consumption of this block can't be determined, but the changes on the power that the different configurations produce on the aggregate will be tracked.
- Data throughput on the different channels (UDP and HDMI)

3.6.2 Related state of the art

N/A

3.6.3 Commercial positioning

N/A

3.6.4 Extension within FitOptiVis

TRL@M0:
TRL1
Extension within FitOptiVis:
Full development of the component on the project
TRL@M36:
TRL4

3.6.5 Prospective adoption

The IP will be used in the Use Case 10 of this project.

3.6.6 Intellectual Property information

This component is targeted to be distributed as open source. When the development of the component is finished this information will be confirmed.

3.6.7 Related documents

N/A

3.7 License plate detector [CAMEA]

License plate detector is designed on the bases of state-of-the-art object detection algorithm trained particularly for license plate. It can be used for smart camera based on hybrid platform (e.g. Xilinx Zynq) solution.

3.7.1 Short description

License plate detector achieves a real-time performance on multiple scales, without the requirement of external DRAM memory. It is fully accelerated and synthesized in form of FPGA firmware primarily for hybrid platform Xilinx Zynq. This component can be then

used in smart cameras equipped by compatible hardware, e.g. the smart camera developed by CAMEA.
Detector receives images from camera sensor (possibly composed HDR) and processes every frame and tries to find LP in it. The results are then passed to post-processing engine (usually outside the FPGA but inside Zynq platform – running in ARM). Parts of detector processing chain can be eventually separated or re-implemented for various platforms.
Input:

- Video interface – AXI Stream data video, maximum image width 2048 pixels, 8 bits per pixel
- Configuration Interface – AXI Lite

Output:

- Data interface - AXI Stream data, 64 bits per detection (position x, y, width, height and likelihood)

Target:

- Main target: Xilinx Zynq Z-7020
- Side target: Xilinx Zynq, Zynq UltraScale and UltraScale+ FPGA's families

3.7.1.1 Updates during the second year of the project

During the second year of the project, the detector was completely re-designed and it offers better detection performance, speed and lower FPGA resource requirements. The latest version of detector is now evaluating modern Aggregated Channel Features (ACF), replacing the memory demanding Local Binary Patterns (LBP) features. The lower memory requirements enable to implement multiple models within one device (e.g. different detection window sizes, multiple object classes, etc.) which makes the new design more flexible and applicable to different tasks.

3.7.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Qualities:
License Plate detector component has following qualities:

- Detection success rate – the main quality of detector is detection success rate of individual license plates. Precision versus performance trade-off can be exploited changing detector configuration.
- Power consumption – has to be kept low allowing battery operation and operation in sealed box (low heat dissipation)

Set-Points:

- Design-time
  - Lowering resolution of input image - limiting number of scales -> saving resources
  - Lower-frame rate -> computationally weaker platform can be used
  - Known (or reduced) scale of LP saving resources
- Run-time
  - Day/night classifier (switched depending on day time)
  - Scale number reduction (or framerate reduction) enabling more classifiers running, e.g. for 1 line and 2 line LPs

Available model:
N/A

3.7.1.3 Design-time support

- The FPGA firmware is designed using current commercially available tools (mostly VHDL oriented) such as Xilinx Vivado.
- C++ reference design for co-simulation

3.7.1.4 Run-time support

Monitoring:
Profiling of units in FPGA and software, displaying in special tool and looking for hiccups and conflicts

- FPS counter
- load/power consumption estimator

RT adaptivity:
N/A
Programmability:
FPGA firmware with programmable units via registers (classifier configuration, LP size, searching window, …).
Programming API:
on IP core without proprietary API (only AXI bus access).

3.7.2 Related state of the art

Object detection in embedded systems is an important task that many applications of computer vision and scene analysis benefit from. Industrial quality control systems address various markers, traffic monitoring uses detection of cars and license plates, biometric systems detect faces and facial features, driver assistance systems detect cars and pedestrians. The detection is especially important in applications that directly rely on it, such as recognition or tracking, and in these applications, the speed, accuracy, power consumption, and/or robustness of detection matters most.
The object detection in embedded devices typically belongs to one of the three detection method categories.

1. AdaBoostbased detectors – cascades of boosted classifiers [VJ2014] or soft cascades [DAB+2014]. They typically use Haar image features [HV2011], or LBP [ZJM+2013].
2. Support Vector Machines (SVM) with Histograms of Oriented Gradient features (HOG) [KBT2016], [KTT2011], [MTM2011].
3. Other methods implementing detection with background subtraction [YLM+2017], keypoints [BNP2010], neural networks [FMA+2010], or custom detection algorithms [SK2013].

In this project we focus on boosted detectors which analyze sub-windows of an input image by a classifier composed from weak classifiers based on simple image features such as Haar [VJ2014] or Local Binary Patterns (LBP) [ZJM+2013]. Multi-scale detection is solved by scaling and processing of the input image in multiple resolutions – image pyramid. Embedded object detectors are often implemented directly in software using libraries such as OpenCV [OPENCV]. While this approach is easy and straightforward, it often is quite slow as detection is computationally demanding task and embedded processors tend to be simpler and slower than desktop CPUs. Another approach is to implement a custom detection algorithm exploiting various acceleration resources of the target platform – CPU [JHZ2010], GPU [HJJ+2011] or Field Programmable Gate Array (FPGA) [ZJM2013], [KBT2016], [KTT2011], [MTM2011], [HV2011], [KT2011] units. This is advantageous in many areas where the deployment

of standard PC-based or embedded software solution is not possible, e.g. because of resource consumption, physical dimensions, industrial or military conditions, etc..

3.7.3 Commercial positioning

N/A

3.7.4 Extension within FitOptiVis

TRL@M0:
Within FitOptiVis project, license plate detector component started in form of proof of concept (TRL3/4).
Expected Extensions:
Cutting edge algorithms, HDR extension, power aware redesign.
TRL@M36:
At the end of the project, thanks to the extensions of detector, the TRL6 is expected. Test and validation in simulated and optionally real environment will be done.

3.7.5 Prospective adoption

The license plate detector component will be adopted in Road Traffic Surveillance use case within the FitOptiVis project.

3.7.6 Intellectual Property information

Trial and academic use for free. Paid license for commercial use and service for fee.

3.7.7 Related documents

[VJ2014] P. Viola, M. J. Jones, “Robust real-time face detection”, IJCV, 2004.
[ZJM+2013] P. Zemcik, R. Juranek, P. Musil, M. Musil, M. Hradis, “High performance architecture for object detection in streamed videos”, International Conference on Field-Programmable Logic and Applications, 2013.
[OPENCV] OpenCV library. Online: http://opencv.org
[JHZ2010] R. Juranek, A. Herout, P. Zemcık, “Impelementing Local Binary Patterns with SIMD instructions of CPU”, in International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 2010
[HJJ+2011] A. Herout, R. Josth, R. Juranek, J. Havel, M. Hradis, P. Zemciık, “Real-time object detection on cuda”, Journal of Real-Time Image Processing, 2011.
[KBT+2016] C. Kyrkou, C.-S. Bouganis, T. Theocharides, M.M. Polycarpou, “Embedded hardware-efficient real-time classification with cascade support vector machines”, IEEE Transactions on Neural Networks and Learning Systems, 2016.
[KTT2011] C. Kyrkou, C. Ttofis, T. Theocharides, “Fpga-accelerated object detection using edge information”, International Conference on Field-Programmable Logic and Applications, 2011.
[MTM2011] S. Martelli, D. Tosato, M. C. V. Murino, “Fast fpga-based architecture for pedestrian detection based on covariance matrices,” International Conference in Image Processing, 2011.

[HV2011] C. Huang F. Vahid, “Scalable object detection accelerators on FPGAs using custom design space exploration”, IEEE Symposium on Application Specific Processors, 2011.

[KT2011] C. Kyrkou, T. Theocharides, “A flexible parallel hardware architecture for adaboost-based real-time object detection”, IEEE Transactions on Very Large Scale Integration Systems, 2011.

[DAB+2014] P. Dollar, R. Appel, S. Belongie, P. Perona, “Fast feature pyramids for object detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014.

[YLM+2017] C. H. Yeh, C. Y. Lin, K. Muchtar, H. E. Lai, M. T. Sun, “Three pronged compensation and hysteresis thresholding for moving object detection in real-time video surveillance”, IEEE Transactions on Industrial Electronics, 2017.

[BNP2010] D. Bouris, A. Nikitakis, I. Papaefstathiou, “Fast and efficient fpga based feature detection employing the surf algorithm”, IEEE Symposium on Field-Programmable Custom Computing Machines, 2010.

[FMA+2010] C. Farabet, B. Martini, P. Akselrod, S. Talay, Y. LeCun, E. Culurciello, “Hardware accelerated convolutional neural networks for synthetic vision systems”, IEEE International Symposium on Circuits and Systems, 2010.

[SK2013] Y. D. Sato, Y. Kuriya, “Multi-scale elastic graph matching for face detection”, EURASIP Journal on Advances in Signal Processing, 2013.

3.8 MDC-generated coarse grained reconfigurable HW accelerators [UNISS, UNICA]

In heterogeneous platforms the programmable logic is tightly coupled with on board microprocessors running embedded operating system (OS) [ABG+2016]. This makes FPGA devices suitable to run highly demanding applications at reasonable power cost, while guaranteeing high performance. In such architectures, the embedded OS manages HW tasks in a classical SW like manner [BSB2013], easing the access to the configurable logic and making the HW-SW approach more appealing.

3.8.1 Short description

Purpose and Mission
MDC tool, as described in D3.1, provides coarse-grain reconfiguration (CGR) support, generating application-specific reconfigurable substrates and packing them within accelerators ready to be used in Xilinx FPGA platforms.
Principal key features and properties
CGR is applied at the word/data level, rather than at bit level. The accelerator can switch among a set of pre-defined functionalities or surf among different available profiles of a given application. MDC-compliant systems follow an application to architecture approach: the CGR substrate is shaped according to a set of desired applications, resulting in an application specific platform capable of achieving strong execution efficiency, but limited to a fixed set of applications.
Processed inputs and produced outputs
Application-Specific (user defined)
Connectivity and communication
MDC tool offers different processor to co-processor communication infrastructures. The user can select the type of coupling among memory-mapped and stream-based, and

the possibility of adopting DMA to manage data transfers. Proper APIs are automatically built upon the setting preventing the user from the burden of low-level infrastructure details management.

Composability and granularity of the component

MDC tool exploits horizontal composability properties to build accelerators. It adopts a high-level synchronous dataflow representation of the kernels to be accelerated; starting from them, datapath merging techniques are used to create the reconfigurable substrate. Crossbar switches (SBoxes) are inserted at the cross-roads of different paths, and data are driven according to the specified configuration patterns.

3.8.1.1 Updates during the second year of the project

During the second year of the project, MDC-generated coarse grained reconfigurable HW accelerators have been improved by adding the possibility of exposing to the external world signals useful for monitoring interesting aspects of the accelerators systems (e.g. latency, faults). In particular, the user will be able to select the level of monitoring, that is the level on which monitored signals are placed: it can be at system bus level (first level), at accelerator atomic execution level (second level) and/or at accelerator datapath level (third level). Please note that, MDC-generated coarse grained reconfigurable HW accelerators have been only made capable of exposing signals to be monitored, while monitoring activities are performed by the evolution of the AIPHS component (see Section 3.4).

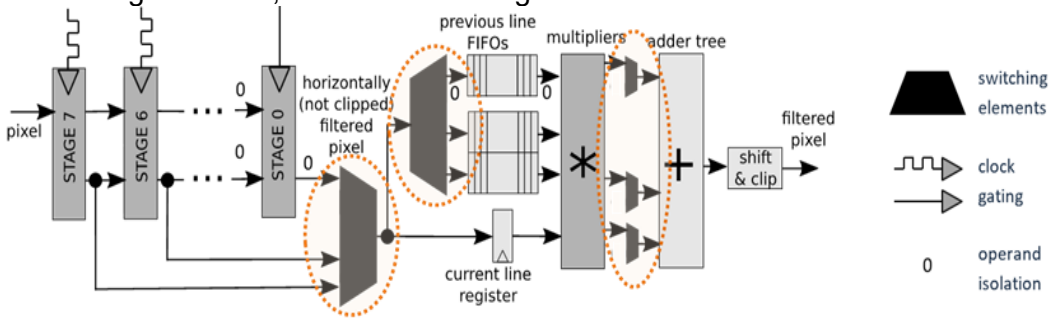
3.8.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Qualities:

MDC-compliant CGR accelerators are particularly suitable to offer run-time reconfiguration among different system execution profiles. Implementations are application-specific.

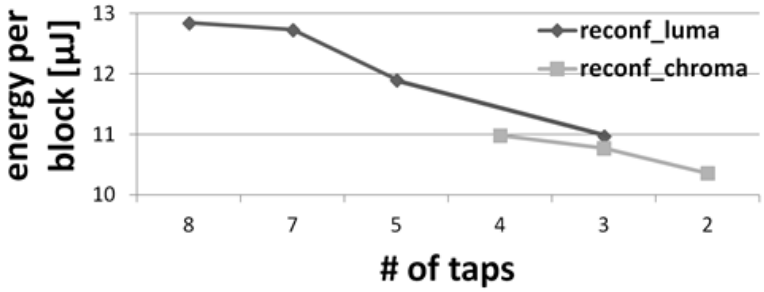
Approximate HEVC interpolation filters - Energy versus Quality Trade-Off

High Efficiency Video Coding (HEVC) exploits the temporal redundancy in a video; in a sequence of frames it codes only the reference frame, while in the following only the movement of the objects is coded. If the movement of the objects involves fractional values of pixels, a block is predicted by interpolating its reference block. Exploiting functional approximate computing it is possible to save up to 28% of energy, with a small degradation of the video quality, obtained by reducing the number of taps in the filter [NMP2019]. In HW, when the quality can be downgraded, part of the computational stages in a CGR infrastructure can be excluded from the computation by properly driving the switching elements, as shown in the figure below.



Approximate HEVC interpolator.

MDC-compliant CGR filter guarantees different trade-off in terms of energy versus quality, by changing at runtime the number of taps involved in the computation [PSF+2017, SPP+2017]. The figure below shows a clear trade-off between energy consumption and image quality. This variability can be exploited on a smart device equipped with a proximity sensor: when the user is close to the device, quality should be high, but when he/she is far, and cannot distinguish details, the quality can be lowered.



Quality versus Energy Trade-Off in the HW Approximate HEVC interpolator.

Set-Point(s) and Parameters:
Surfing among configurations in MDC-compliant CGR accelerators is made possible by configuring the accelerator through proper APIs, selecting the network identifier *ID*. The table below summarizes the available trade-off for the HEVC interpolator.

Design	Working points	Trade-off	Target Technology
HEVC	Number of taps involved in the computation	Energy vs Quality **High quality** - 8 taps luma and 4 taps chroma. **Low Energy** - 3 taps luma and 2 taps chroma.	Artix-7 FPGA

3.8.1.3 Design-time support

- Design and customization of MDC-compliant CGR accelerators is done using the MDC tool (details are provided in D3.1) [PFS+2017, SMR+2015].
- The outcome of the MDC tool, intended as CGR stand-alone datapath, is a target independent HDL description. However, synthesis and deployment of MDC-compliant CGR systems is target dependent, since vendor specific flows have to be used for synthesis
- As explained in D3.1, MDC allows the automated deployment of ready-to-be-used accelerators over Xilinx FPGA platforms. The Vivado design environment is used in this specific case.

3.8.1.4 Run-time support

In MDC-compliant accelerators the cost of reconfiguration is minimized, both in terms of time (word-based reconfiguration is performed in one clock cycle) and power (no need of downloading a new big bitstream through dedicated channels).
MDC tool handles programmability, keeping trace of the SBoxes configuration patterns while merging the input specifications. Those patterns are saved into dedicated Look-Up Tables. A configuration module in the HDL properly sets the SBoxes selectors according to the network ID.

Run-time adaptivity support:
In CGR systems different paths of data can be activated at run-time, excluding the un-necessary resources from the computation. In MDC-compliant accelerators different kernels/profiles have different IDs, which activates different paths of data.
Monitoring:
The monitoring approach we are investigating is compatible with the Performance Application Programming Interface (PAPI) [PAPI]. To access the monitors a PAPI-MDC Component, compatible with the already existing PAPI components, has been defined in [MF19]. This component is automatically configured, according to the number of events to be monitored, using an XML file.
Programmability and Programming API
The MDC design suite offers advanced features related to system integration. Each CGR HW accelerator can be easily programmed by means of simple APIs that expose to the users C functions to run accelerator with a certain set point, hiding the details of the accelerator, like coupling with the processor or DMA usage.
Dealing with monitoring features, HW monitors are accessed through the configuration registers of the accelerator. To facilitate the reading and make it transparent for the user, the PAPI calls [PAPI] are used.

3.8.2 Related state of the art

Modern complex applications are required to be flexible and power-aware, especially in the Cyber Physical Systems (CPS) and Internet of Things (IoT) contexts. Despite Application Specific Integrated Circuits (ASICs) can better fit with the computing and power efficiency demand, they are not flexible enough to meet reactiveness and dynamism requested in the CPS context. The FPGA-based platforms are meant to provide efficiency close to the ASICs, while offering also programmability [T2015]. CGR approaches as those used by MDC-compliant accelerators already demonstrated to be suitable to support adaptivity needs in CPS [PSF+2017, SPP+2017] contexts, providing requested flexibility and dynamism. At the same time, their deployment and usage are made friendly by a complete flow for automatic customization and programmability support. With respect to this context FitOptiVis studies related to CGR accelerators is focusing on different aspects.

- Approximate Computing: It introduces quality as a new design metric to be traded-off with power consumption, performance or resource occupancy, expanding the design space. Combining approximate computing with CGR seem to be a viable and suitable solution to address runtime trade-off management, but there is a *lack of specialized supporting tools for approximate computing*.
- Monitoring: One of the most adopted SW approaches for performance monitoring is based on the Performance Monitoring Counters (PMCs) existing on modern CPUs. PAPI provides a unified method to access these PMCs [PAPI]. Madroñal et al. [MML+2018] presented PAPIFY, a tool that provides a lightweight monitoring infrastructure by means of an event library aimed at generalizing PAPI for embedded heterogeneous architectures. A first tentative to exploit PAPI to monitor HW reconfigurable systems has been done by Suriano et al. [SMR+2018], which present a custom approach to read monitors of a HW architecture based on Dynamic and Partial Reconfiguration. The monitoring approach to be adopted by MDC-compliant accelerators goes in the same direction. PAPI-based accesses are meant to be generic enough to be easily extended to different types of application specific HW substrates, not necessarily CGR based ones.
- Runtime modeling: Since CGR acceleration has to guarantee fast reconfiguration and the correct execution associated to the different configurations of the system,

a runtime strategy to monitor and estimate of the performance metrics is needed. Methods based on abstractions reduce the explosion of complexity, such as problems of scalability. Among these methods, modeling of architectures (MoA) can be used in order to achieve energy and latency estimation, as similarly proposed in [PMD+2017]. MoAs for different estimation of qualities have to be derived to be used at run-time to drive the choices upon trade-off to be taken at.

3.8.3 Commercial positioning

N/A

3.8.4 Extension within FitOptiVis

Defining TRL level for MDC-compliant accelerators it is not feasible. These accelerators are the outcome of an academic tool, with TRL3-4. What we propose is more a methodology to define flexible and re-programmable accelerators rather than one specific single component.
TRL@M0:
N/A
FitOptiVis Extensions:
N/A
TRL@M36:
N/A
In future releases of this document we expect to describe MDC-generated accelerators specifically conceived for FitOptiVis use-cases. In those cases, it will be possible to provide quantitative evaluation of each specific component.

3.8.5 Prospective adoption

MDC-compliant accelerators are meant to be used within the water supply use case. Starting from a high-level dataflow description of the functionalities to be executed, MDC tool generates ad-hoc accelerators with multiple, configurations (if more than one functionality shall be executed on the same accelerator) or working points (if given a certain functionality more profiles, with different trade-offs, shall be enables).
We expect to deliver at M24 the fist MDC-compliant accelerators for the water supply use case, however the exploitation of this type of acceleration/reconfiguration is generic enough to be used in other application scenarios.

3.8.6 Intellectual Property information

As discussed in D3.2, MDC has already been released open source (https://github.com/mdc-suite/mdc), so that the flow for generating coarse grained reconfigurable HW accelerators is already available for both users, as an executable, and developers, as source code. In future releases of this document we expect to describe MDC-generated accelerators specifically conceived for FitOptiVis use-cases. In those cases, it will be possible to provide information about Intellectual Property specific for each developed accelerator. Documentation with examples and tutorials on MDC is available on [RPCT] and [IDEA]

3.8.7 Related documents

[SMR+2015] C. Sau, P. Meloni, L. Raffo, F. Palumbo, E. Bezati, S. Casale-Brunet, M. Mattavelli, “Automated Design Flow for Multi-Functional Dataflow-Based Platforms”, Journal of Signal Processing Systems, 2015.

[PSF+2017] F. Palumbo, C. Sau, T. Fanni, L. Raffo, “Challenging CPS Trade-off Adaptivity with Coarse-Grained Reconfiguration”, Conference on Applications in Electronics Pervading Industry, Environment and Society, 2017.

[SPP+2017] C. Sau, F. Palumbo, M. Pelcat, J. Heulot. E. Nogues, D. Menard, P. Meloni, L. Raffo, "Challenging the Best HEVC Fractional Pixel FPGA Interpolators with Reconfigurable and Multi-frequency Approximate Computing" IEEE Embedded Systems Letters, 201.

[PFS+2017] F. Palumbo, T. Fanni, C. Sau, P. Meloni, “Power-Awarness in Coarse-Grained Reconfigurable Multi-Functional Architectures: a Dataflow Based Strategy”, Journal of Signal Processing Systems, 2017.

[MF19] D. Madroñal and T. Fanni, “Run-time performance monitoring of hardware accelerators”, International Conference on Computing Frontiers, 2019.

[ABG+2016] S. Ahmad, V. Boppana, I. Ganusov, V. Kathail, V. Rajagopalan, R. Wittig, “A 16-nm Multiprocessing System-on-Chip Field-Programmable Gate Array Platform”, IEEE Micro, 2016.

[BSB2013] N.W. Bergmann, S. Shukla, and J. Becker, “QUKU: a dual-layer reconfigurable architecture”, ACM Transactions on Embedded Computing Systems, 2013.

[NMP2019] E. Nogues, D. Menard, M. Pelcat, "Algorithmic-Level Approximate Computing Applied to Energy Efficient HEVC Decoding," IEEE Transactions on Emerging Topics in Computing, 2019.

[PAPI] Performance API. Online: http://icl.utk.edu/papi/

[T2015] S. Trimberger, “Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology”, IEEE, 2015.

[MML+2018] D. Madroñal, A. Morvan, R. Lazcano, R. Salvador, K. Desnos, E. Juarez, C. Sanz, “Automatic Instrumentation of Dataflow Applications Using PAPI”, ACM International Conference on Computing Frontiers, 2018.

[SMR+2018] L. Suriano, D. Madroñal, A. Rodriguez, E. Juarez, C. Sanz, E. de la Torre, “A Unified Hardware/Software Monitoring Method for Reconfigurable Computing Architectures Using PAPI”, International Symposium on Reconfigurable Communication-centric Systems-on-Chip, 2018.

[PMD+2017] M. Pelcat, A. Mercat, K. Desnos, L. Maggiani, Y. Liu, J. Heulot, J. F. Nezan, W. Hamidouche, D. Ménard,S. S. Bhattacharyya, “Reproducible evaluation of system efficiency with a model of architecture: From theory to practice”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems2017.

[RPCT] Reconfigurable Platform Composer Tool project. Online: http://sites.unica.it/rpct/

[IDEA] IDEA Lab YouTube channel. Online: https://www.youtube.com/watch?v=_cyYFJCDR3U&list=PLql1YxTzHalZztJPu7wn0uzAYbr81QTpH

3.9 Low energy dynamic voltage and frequency scaling enabled CPU with accelerators [UTU]

This component will be a full custom ASIC, supporting dynamic voltage and frequency scaling, thus enabling low power operation. Additionally, DVFS capabilities will be automated and extended to include memories (if feasible after study) and accelerators.

3.9.1 Short description

This component will be a full custom IC. Actually three (3) versions are planned over the whole project duration. The first version is currently in design phase, and will be submitted for processing in the fall, in the next suitable batch. The CPUs will be based on RISC-V architecture, guaranteeing good SW support and compilers and so on. The idea is to test DVFS components in the first version, make them autonomous in the second version, and optimize the system in the final version. Additionally application specific accelerators will be implemented to versions 2 and 3.
Type:
HW
Target:
Full Custom IC
Input:
Any data, but in FitOptiVis context we expect video stream (2D and/or 3D)
Output:
Any data, but in FitOptiVis context we expect 3D data and/or content analysis data

3.9.1.1 Updates during the second year of the project

During Y2 the CPU version 1 has been designed, and submitted to the fab for processing. The current estimation is that UTU will receive the chips during April. Version 1 has both frequency and voltage scaling facilities built into the chip (not external) and controlled from the CPU core via software. Also test version of an in-house memory block is included. This will allow future extensions to use DVFS for the memory resources in addition to just the processing resources.
The test setup (PCB and software) is currently being designed (March 2020). These are expected to be ready at the same time as the chips arrive.

3.9.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

The DVFS system allows tuning energy consumption based on the load experienced by the CPU. This tuning will be done during run-time, and later versions will have autonomous tuning capabilities, while the first one will be software controlled.
One of the accelerators planned is a limited precision convolutional filtering engine. It will provide design-time parameters for desired precision and also fixed kernels for filter weights.
Other accelerators will be added, once our partners reach a state of maturity, so that we can implement functions that provide benefits to their use case(s). Such co-operation is planned with Nokia and TUT. Others are possible, but not agreed upon yet.

3.9.1.3 Design-time support

We will provide guidelines and manuals on how to tune the limited precision accelerator core. Also at least a technical note will be provided regarding the DVFS capabilities in the CPU.

3.9.1.4 Run-time support

Software support for using the limited precision core and DVFS capabilities will be provided. The software support kit will be in the form of C/C++ driver kit. This will most likely be very basic in functionality, and does not deserve to be called an API. Basically just wrappers to handle the memory mapped I/O to and from the DVFS unit and the limited precision core.

Run Time Adaptivity:

The CPU can change the configuration of the DCDC converter providing the power supply and the frequency synthesis unit providing the main clock during run-time. In later versions of the chip, this functionality will be made autonomous or semi-autonomous, based on the system load.

Programmability:

As this is a Turing Complete device, it can be programmed to perform any functionality. The RISC-V core is supported by most programming languages commonly used, and the full GNU tool set is available for free. Also commercial solutions for programming the RISC-V core exist, but we will be using GNU based tools for the foreseeable future. No specific API is required, and full C/C++ toolset is available.

Monitoring:

- Monitored items will include, but not be limited to:
- Voltages
- Clock frequency
- System load (CPU idle cycle count)
- Current consumption estimate from the DCDC converter

3.9.2 Related state of the art

Some DVFS capable implementations of the RISC-V core already exist. As a reference, see [GST+2017]. We plan to extend the capabilities, increase automation and include accelerator cores to the DVFS scheme. Also, we plan to investigate including memories to the DVFS domain. Typically memories operate at such a small margin, that they do not function correctly under other conditions than the nominal. If tests show promising results, full custom memory banks that can use varying voltage and frequency will be included in later versions.

3.9.3 Commercial positioning

N/A.

3.9.4 Extension within FitOptiVis

TRL@M0:

TRL9 for the CPU core, RISC-V architecture is widely used.

TRL4 for the DVFS functionality, as it has to be tailored for each IC manufacturer's processes and device operating conditions, but the theory is well known.

TRL2 for the DVFS capable memories, the basis for these has been studied in ALMARVI, but not designed for our current IC process, also some major questions remain, regarding the performance of such memory blocks.

TRL0 for the accelerators, as they are new in design, and have no prior research.

Expected Extensions:

Within FitOptiVis, there are plans to improve DVFS capable CPU at least in the following aspects:

- Optimize DCDC based voltage scaling

- Optimize clock frequency scaling
- Automate proper DVFS operating point selection (In HW)
- Add support for accelerators
- If feasible, add DVFS capable memory structures

TRL@M36:
TRL9 for the CPU core, RISC-V architecture is widely used.
TRL9 for the DVFS functionality will be active in real world device.
TRL4/6 for the DVFS capable memories, which will be simulated and at least a prototype implementation will be included, if feasible on current IC technology.
TRL4/9 for the accelerators, depending on the functionality, which is still open in some details.

3.9.5 Prospective adoption

Usage in FitOptiVis use cases: UC2
Fields of application – Improved execution engine, with very low energy footprint.

3.9.6 Intellectual Property information

The CPU core uses Open Source RISC-V architecture [RISCV], but the hardware blocks for generating voltages and frequencies are dependent on the chip manufacturer's libraries, and thus not available as open source.

3.9.7 Related documents

[RISCV] RISC-V documentation. Online: https://riscv.org/
[GST+2017] M. Gautschi, P.D. Schiavone, A. Traber, I. Loi, A. Pullini, D. Rossi, E. Flamand, F. K. Gürkaynak, L. Benini, "A near-threshold RISC-V core with DSP extensions for scalable IoT Endpoint Devices", IEEE Transactions on Very Large Scale Integration Systems, 2017.

3.10 Multistream video image scaler and compositor [PHL]

The multistream video image scaler and compositor component is the hardware basis for use case 6: multi source streaming composition.

3.10.1 Short description

In the multisource streaming composition use case several Ethernet based video streams are combined into one or more large screen images in a medical Xray application. Although the final goal of this use case is to avoid image scaling as much as possible, it must be supported for legacy image sources (those that cannot adapt their output to the requested format). But in the transitional period when a viewport-layout is changed, the video sources cannot react instantaneously so they still output the 'old' video format for some period of time so scaling is also needed during layout changes. Since the use case mandates very short latency for selectable input streams, the component needs to be optimized for this purpose and will support synchronization between the output device (display) and an input source. As such this component consists of the following sub-components:

- Ethernet inputs and router

- A set of noise compensated LancZos-2 downscalers [LANCZOS] shared by multiple input streams
- Image memory
- Output scaler/compositor(s)
- Ethernet output [ETH]

Input:

- Multiple 10G optical Ethernet [ETH] interfaces.
- Overlay image stream(s) generated by the controlling processor subsystem

Output:

- Multiple 10G optical Ethernet [ETH] interfaces
- May be shared with the input interfaces

Target Technology:

- Main target: Xilinx Ultrascale+ MPSOC ZCU7EV [XILINX]
- Side target: other members of the Utrascale+ MPSOC family and the upcoming Xilinx Versal FPGA's

3.10.1.1 Updates during the second year of the project

The full implementation of this component is not ready yet, but the major building blocks have been verified in simulations. Meanwhile improvements to the design have been made both in the area of enhanced flexibility and reduction of FPGA resource usage. For reference of the down scaler architecture depicted in Figure 3 has been adopted. It is mainly composed by:

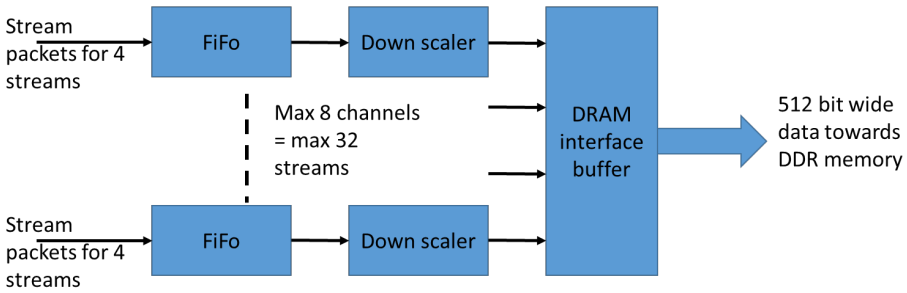


Figure 3 Down scaler reference architecture.

- A single down scaler now supports up to 4 different input streams as long as the total line-width of the input streams does not exceed 4K pixels. We are still working on improvements on the resource usage (notably memory usage) of this scaler. Target is to reduce the memory usage from 10 to 5 memory blocks.
- A special type of Fifo has been developed that can handle multiple streams in the same memory block. Storage allocation for the 4 input streams within this memory block is fully automatic. This Fifo is used at the input of the down scalers and will be extended with monitoring capabilities to measure throughput and package jitter of the incoming streams. This Fifo can be used as a separate component outside the full scaler and compositor component.
- The 32-bit data from the down scalers needs to be stored in DRAM which uses a 512-bit wide interface. We developed a very resource efficient

interface block that can buffer and translate the data for the DDR memory from 8 down scalers at full speed with minimal resource usage.

The up scalers/compositor blocks have been verified in simulation. These simulations also show that the target maximum latency can be met. Due to some changes in the architecture the viewports can now be rendered next to each other without borders in between while kernel folding[1] at the edges of the viewport is still possible.

3.10.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Image input comes in 2 flavours:

- Fixed format (legacy devices)
  - 640x480
  - 1024x768
  - 1280x720 (720p)
  - 1920x1080 (1080p, FHD)
- Variable formats with dynamic adjustment
  - Any format between 640x480 till 1920x1080; this creates the ability to negotiate the format with the images source to optimize on bandwidth and quality.
- Image output:
  - 3480x2160 (4K, 60 Hz)
  - 2560x1440 (1440p, 60 Hz)

3.10.1.3 Design-time support

We will provide guidelines and manuals on how to use this core and the accompanying driver that runs on the ARM-R5 cores of the FGA device.

3.10.1.4 Run-time support

Monitoring:
N/A
RT adaptivity:
N/A
Programmability and Programming API:
There will be a software driver that does all the low level runtime calculations that runs on the ARM-R5 cores of the FPGA including a communication interface library and API description.

3.10.2 Related state of the art

Although Lanczos image scalers already exist for a long time, they are usually implemented on general purpose CPUs or GPUs using standard buffering techniques. Examples in the medical domain: [FLEX] Philips Flexvision and [SLS] Siemens large screen products. This means that power consumption is relatively high and due to the generic buffering the latency is long (1 - 2 complete frame times). Especially for downscaling, the Lanczos algorithm leads to large filter kernels and thus a high computational load so for FPGA implementations it is usually replaced by a small

[1] Without kernel folding the scaler must assume that outside the source image for a viewport the pixels have a certain fixed value which leads to image artefacts at the edges of the viewport. With kernel folding the image is virtually extended with the pixels right at the edge which gives much better results.

[BICUB] bicubic or even just [BILIN] bilineair interpolation kernels leading to less image quality.

3.10.3 Commercial Positioning

This component will be primarily developed and optimized for the targeted use case and as such is planned to be used in a product replacing the current solutions. However, it suits many other applications like traffic monitoring or surveillance systems equally well. Competitor information is omitted by PHL to avoid publication of components details which are considered confidential.

3.10.4 Extension within FitOptiVis

TRL@M1:
TRL1
Expected Extensions:
Within FitOptiVis for this component we will develop a set of FPGA-resource optimized scalers that can do true Lanczos interpolation also for downscaling including the required buffering and synchronization for low latency (< 3 msec) processing.
The up-scalers will support image composition onto 1 or more large screens.
TRL@M1:
TRL6/7

3.10.5 Prospective adoption

Usage in FitOptiVis use cases: UC6.
Fields of application: image scaling and compositing.

3.10.6 Intellectual Property information

The scalers and compositor in this component use a number of techniques that are considered a trade-secret. Because of this, the component cannot be an open source component, but it will be delivered as an encrypted VHDL package for Xilinx FPGA's.

3.10.7 Related documents

[LANCZOS]	Lanczos algorithm. Online: https://en.wikipedia.org/wiki/Lanczos_resampling
[BICUB]	Bicubic interpolation. Online: https://en.wikipedia.org/wiki/Bicubic_interpolation
[BILIN]	Bilinear interpolation. Online: https://en.wikipedia.org/wiki/Bilinear_interpolation
[FLEX]	Philips FlexVision. Online: http://www.usa.philips.com/healthcare/product/HCOPT15/ flexvision-xl-display-screen
[SLS]	Siemens Large Screen products. Online: https://www.siemens-healthineers.com/en-us/angio/options-and-upgrades/components-and-options/artis-large-display
[XILINX]	Xilinx Ultrascale MPSoC. Online: https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html

[ETH] 10G optical Ethernet standards IEEE802.3ae and IEEE802.3aq

3.11 Efficient magnetic field generators [PHL]

This kind of component deals with magnetic field generators adopted in several medical applications.

3.11.1 Short description

The purpose of these components is to convert an electric current into magnetic field of a specific target shape. The efficiency of these components can be expressed in terms of stored energy and dissipation for a defined reference characteristic of the generated field.
The input is therefore electric current (in Amperes) and the output is magnetic field (in Tesla).
For FitOptiVis, two types of magnetic field generators are considered:

- The gradient coil. The target spatial field distribution is a spatially linear increasing magnetic field in a low (< 6kHz) frequency range. The power consumption is in the order of tens of kW. The rapid switching of the magnetic field of a gradient coil is capable of causing stimulation of the peripheral nerves or even cardiac stimulation, as well as the source of high levels of acoustic noise (up to 130 dBA).
- The body coil. The target spatial field distribution is a uniform field at a virtually fixed frequency of tens of MHz. This field is capable of causing significant tissue heating.

3.11.1.1 Updates during the second year of the project

The main updates on the efficient magnetic field generators during the second year of the project are:

- Two prototypes of the gradient coil have been delivered and tested in Y2. The intended performance has been confirmed. An issue related to the manufacturing process has been identified on the first prototype, which has been resolved on the second one. The applied FitOptiVis technology relates to abstraction of the gradient coil properties, so that the system can optimally use this component while also supporting many other variants. The way this is done is part of the PHL IP, and it will be not published, rather it will be implemented in PHL products. Also note that IP is present and has been used, most of it even not belonging to PHL but to the supplier, in designing and building this component.
- The body coil has been designed by addressing and optimizing performance and safety. A prototype has been produced internally, also using components produced by FUT, and it is currently under test. The FitOptiVis abstraction here consists of parameters related to patient-safety, which have been determined using extensive modelling techniques. The body coil will be verified in Y3. As with the gradient coil, much of the design and production methodology involves company-private IP.
- A patient-safety related test has been defined, executed on 22 volunteers and the results have been analysed. The objective was to provide a better model predicting for the occurrence of Peripheral Nerve Stimulation (PNS) and thereby the protection mechanism to prevent PNS. PNS is an unpleasant and potentially painful sensation that can occur in MRI when the magnetic field is switched too quickly. The final intention of the test was to improve the related safety standard,

IEC60601-2-33. However, the results were disappointingly inconclusive, and no standard improvement has been possible. A more traditional experiment is defined in Y3 to determine the protection mechanism.

3.11.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Qualities: N/A, the current to magnetic field ratio is fixed.
Set-Point(s) and Parameters:
The set-points are related to the physiological stress that can be caused by driving the component. These physiological stresses (tissue heating, peripheral nerve stimulation and acoustic noise) are described by a model:

- The input is the current.
- The output is a measure of the physiological stress. E.g., for acoustic noise this is the Sound Pressure Level.
- The models also have parameters.

Each model is evaluated design time, that is, at the time a scan protocol is defined by the user. At that moment the input currents that will be driving the components during the execution of the scan are determined. Once these currents are delivered to the components (the scan is 'executed'), a very limited number of run-time changes are possible because of the extreme sensitivity of the spin state to the magnetic fields.

3.11.1.3 Design-time support

In-house developed design tools, simulation packages like ANSYS, development tools like Matlab.

3.11.1.4 Run-time support

Not applicable.

3.11.2 Related state of the art

In the state-of-the-art no MRI product offers the user the ability to control the energy consumption explicitly.

3.11.3 Commercial positioning

The MRI market is characterized by a few players: Canon (Japan) [CANON], GE Healthcare (US) [GEH], Hitachi (Japan) [HITACHI], Philips (the Netherlands) [PHILIPS], Siemens Healthineers (Germany) [SIEMENS] and United Imaging (China) [UNITED].
Energy consumption of an MRI system is not yet an important part of the product data. COCIR, the European Trade Association representing the medical imaging, radiotherapy, health ICT and electromedical industries, has defined a standard measurement method to characterize the energy consumption of an MRI system. The data is intended for potential customers to compare products. At the time being, publication of this information is still voluntary. Philips references COCIR when publishing the average power consumption during scanning in the Philips MRI product data.
In view of the successful sustainability strategy of Philips, an MRI product that allows the user to control the energy consumption explicitly will fulfill market need.

3.11.4 Extension within FitOptiVis

TRL@M0:
TRL4
Expected Extensions:
The energy consumption and acoustic noise shall be modeled with an accuracy of at least 10% and used in a multi-objective optimization scheme.
TRL@M36:
TRL6

3.11.5 Prospective adoption

The exploitation will be in a Philips MRI product. The developed HW and SW components have specific proprietary interfaces.

3.11.6 Intellectual Property information

PHL does not have plans to release FitOptivis technology related IP, e.g. in the form of publications, patents or marketing communications. The FitOptiVis technology is intended to be used in the product, available as a function to the user. IP generated from the activity is protected using standard PHL processes such NDA and supplier agreements.

3.11.7 Related documents

[CANON] Online: https://global.medical.canon/products/magnetic-resonance
[GEH] Online: https://www.gehealthcare.com/products/magnetic-resonance-imaging
[HITACHI] Online: http://www.hitachi-medical-systems.eu/products-and-services/mri.html
[PHILIPS] Online: https://www.usa.philips.com/healthcare/solutions/magnetic-resonance
[SIEMENS] Online: https://www.siemens-healthineers.com/magnetic-resonance-imaging
[UNITED] Online: https://www.united-imaging.com/en/product/product-index

4 Software components

This section describes software components developed in FitOptiVis related to the use cases of the project.

4.1 Person tracking software [UGR]

This component is meant to be used to detect the location of people in the proximity of critical infrastructures, for the Smart-Grid UC9. This component has the objective of monitoring people in the environment of sensitive perimeters, anticipating potentially dangerous or unpredictable behaviour and reacting to intruders by generating alarms. In addition, a module has been included in this software that allows the creation and edition of polygons from point to point selection on a bird's eye view map (see Figure 4), so that they can be used as perimeter information for surveillance tasks in sensitive areas. Thus, each perimeter will be given by a list of real world coordinate points in the format (latitude/longitude).

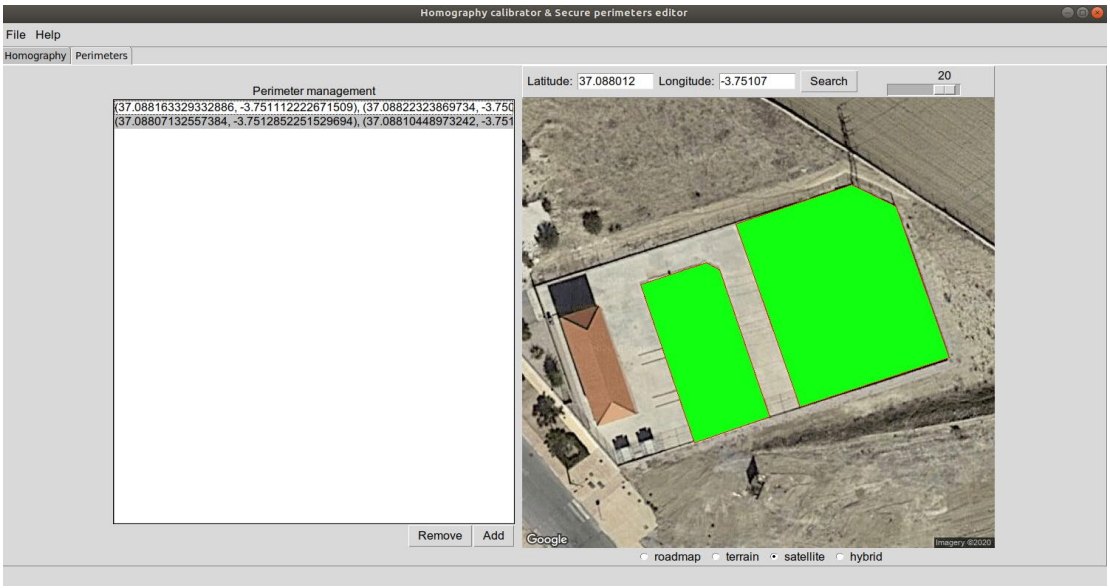


Figure 4 Secure perimeter editor

4.1.1 Short description

The person tracking component entails two main subcomponents: a person detector and a tracker for the detected target. The operation of each of these sub-components is briefly described below:

1) Person detector: the person detector extracts the position of a person moving around the scene. Then, it encapsulates the person in the smallest rectangular region that is called bounding box. This component also provides a characteristic vector of the bounding box.
2) Person tracker: the bounding box and the location determined as the person position are sent to the tracker. The tracker computes the trajectory of the person over all the input sequences while moving around the scene [NGU2016]. Also, the tracker allows the identity of a detected human subject to be maintained in a

consecutive sequence of video frames, using spatial-temporal information and the appearance of previous detections of each of the subjects.

Target:

In order to perform in real-time, the component is developed for a GPU (Graphic Processing Unit), using NVIDIA CUDA C++ for high-performance computation.

Input:

Each of the sub-components has the following entries:

- The person detector input is the set of consecutive frames.
- The tracker takes as input the information from the person detector subcomponent: bounding boxes points (x1, y1, x2, y2) and feature vectors from the image surrounded by these boxes.

Output:

- The person detector outputs will be: the position of the detection, determined by (x1, y1, x2, y2) coordinates in pixels, and a 128 real feature vector for each bounding box.
- The tracker will emit the position of the person (lat., lon.) in latitude/longitude and the unique identifier of each of the human subjects detected, for each of the moments of time in which each chart is analysed and for each camera perspective. An example of the expected result of the corresponding visualization of two detected persons is depicted in Figure 5.

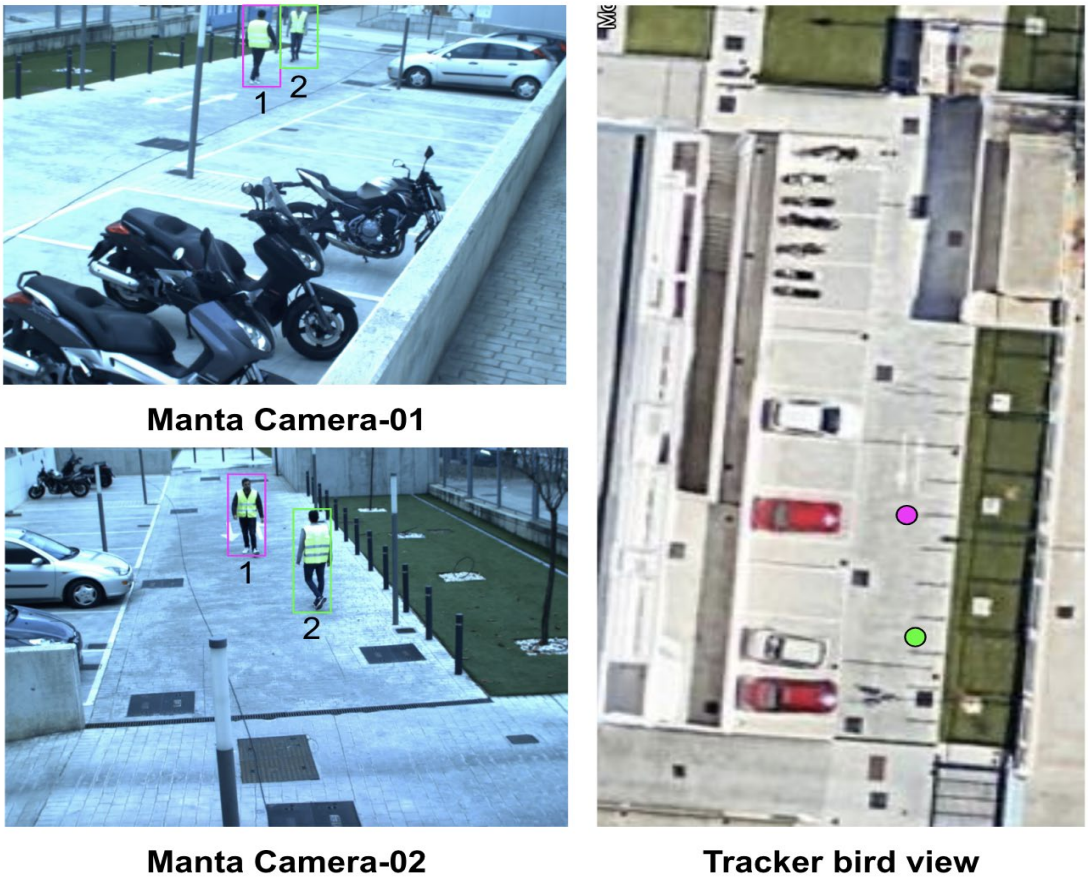


Figure 5 Tracker visualization

4.1.1.1 Updates during the second year of the project

As part of the development of this component, there have been several changes during this second year (M12 to M24), which can be seen in this document. In summary, these changes are presented below:

- The person tracking software is now used exclusively for outdoor environments, since its usage has been limited to electrical substations surveillance. Because of that, now person tracking component is not considered for habit tracking use case (UC3).
- Python becomes the main programming language used, due to its easy handling for debugging and testing tasks and for the amount of packages and libraries it has for artificial intelligence and computer vision tasks, specifically.
- In order to re-identify targets between different frames of the video stream, feature vectors of the detection bounding boxes have begun to be used. These feature vectors are composed of 128 32-bit floats, acquired using a neural network with an autoencoder architecture.
- This year, the possibility of integrating the video-surveillance system into a multi-camera environment is beginning to be included. To this end, different calibration techniques and planar correspondences between cameras have been studied. Finally, homography calibration has been chosen and a software (available in open source) has been developed to facilitate this task (see Figure 6). Having multiple cameras, both independent and overlapped, allows better handling of occlusions, light changes and, in general, offers greater robustness in both person detection and tracking.
- During this year, we have defined the scenarios of interest that trigger the reconfiguration of the different components. Thus, in the absence of subjects on the scene, the system operates in the lowest performance mode, while in the presence of operators, intruders or abnormal circumstances, the quality which the different video surveillance tasks are performed is increased.

4.1.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

With respect to the qualities, we are taking into account accuracy and performance. Accuracy is measured by the Deviation of the location determined for the target, and the PASCAL criterion [SDC+2014] measured by the overlap of the bounding box covering the ground-truth defined by a person. Bear in mind that the component is defined by the tracker, and the person detector is only meant to provide the position of the target to initialize the location. Then, the qualities to be measured concern only the visual tracking operation, not the human detection (although it also affects the final accuracy). In addition, other metrics are used to measure the quality of the re-identification of human subjects in tracking, such as the percentage of tracks Mostly Tracked (MT), as those that have the same tag for at least 80% of their life or the Changes in Identity (SW ID), as the number of times the identity of a track changes during its life cycle [BS2008]. Performance is given by the number of frames tracking locations provided per second.

With respect to the setpoints, some of the potential features to be tuned are given by the following trade-offs: a) number of targets to be processed vs number of frames processed per second; b) accuracy vs number of frames processed per second.

Figure 6 Homography calibration software.

4.1.1.3 Design-time support

In order to develop this component, general design and development tools are used to implement the CPU and GPU code, being Python the programming language mainly used for the design and development of this component.

4.1.1.4 Run-time support

Monitoring:

Three different profile configurations are available that provide different qualities for video analysis (SD, HQ and HD video analysis) at the expense of lower processing performance with runtime switching between configurations.

Programming APIs:

NVIDIA GPU boards (Jetson TX2, Jetson Xavier), and CUDA package Python programming language.

4.1.2 Related state of the art

The process is performed usually as follows: 1) Extract candidate regions for potential human targets, 2) The use of an algorithm of selection of region of interest (ROIs) has the purpose of limiting the search space [RHG+2017]. 3) Compute the center of masses and the dimensions of the smallest bounding box around the target [IMG2017, XSB+2017]. 4) Extract features to build a model of the target, 5) Determine if the target is a human or not, 6) Some of the potential human descriptors listed in the literature are: shape [WR2007]; Local Binary Patterns [HT2010], Histogram of Gradients [OE2009];

motion [NOL2011]; fusion of different features. However, we have chosen to use convolutional neural networks (CNN) on the images directly. These are able to extract information from the pixel content and to learn characteristics automatically without the need, as in the previous image descriptor methods, to carry out explicit programming on the network [TMB+2016].
Second, tracking is the estimation of the trajectory of a target moving around the scene, over consecutive frames [LWW+2018]. A taxonomy is provided next:

A. Matching: these methods compute a representation of the target model using previous frames, e.g. by sampling potential candidates in a window around the position of the previous target location. Then, all possible candidates are compared by matching using normalized cross correlation [LMS+2008], intensity gradient measures in the Lucas-Kanade fashion [SAS2007], or mean-shift [ZP2007]. After that, the new location is determined selecting the candidate with the highest score [CLK2016].
B. Matching with appearance model: previous approach extended with an appearance model strengthened with possible variations of the target [TPM2007].
C. Matching with constraints: sparse optimization [SJ2010].
D. Using discriminative classification: based on foreground-background differentiation, building a classifier for the foreground (target) that is updated continuously [ADF2010].

With respect to the features used to build the models, some examples are: spatio-temporal features, gradient features, color, texture, or the fusion of multiple features. In recent years, with the development and lowering cost of computer graphics systems, an increasing state of the art has begun to develop in terms of tracking by detection methods when dealing these kind of problems [HWW+2018].
Recently, this matching that is done to associate tracks in consecutive moments of time is known as re-identification, and refers to the relationship of each detected human subject over the time of a recording, using for example its last known location and the prediction of its trajectory [ZYW2019].
And finally, some of the issues when performing visual tracking are: change of perspective, noise, changes in the illumination, complex motion, non-rigid targets, occlusions, real-time performance and number of targets (scalability).

4.1.3 Commercial positioning

N/A.

4.1.4 Extension within FitOptiVis

Classic tracking methods achieve an accuracy of more than 80% on average, but when it comes to occlusions, multiple targets or changes in lighting this accuracy drops to about 40% - 50% (understood as the deviation of bounding boxes of less than 5-7 pixels for VGA - SD resolution). We currently maintain the accuracy of classical methods in

normal situations and we reach an accuracy of 65% - 70% when dealing with occlusions, while up to 25 fps is achieved for people tracking.
TRL@M0:
TRL1 - Only the model was defined, no other library/code was available.
FitOptiVis extensions:
At M12, the component is developed and it is able to track up to 3 targets reaching state of the art performance, while achieving around 50% of accuracy. No robust behavior is implemented yet (no occlusion management, no illumination change invariation).
At M24, and in the absence of more in-depth testing, the component already considers detection and tracking achieving real-time performance. Similarly, the integration of this system is being implemented for multi-camera environments.
TRL@M36:TRL4-6 - We expect a tested library for robust outdoor person tracking for, included in the system demonstrators for the Smart Grid UC.

4.1.5 Prospective adoption

Applied in FitOptiVis to the Smart Grid UC. Other fields of application not considered in the project: intelligent transportation systems, pedestrian traffic management, tourist flow estimation, search and rescue applications, automatic counting systems, or human-robot interaction

4.1.6 Intellectual Property information

In parallel to this person tracking component, a tool has been developed to assist in the calibration of one or more cameras by calculating the homographic matrix with respect to the dominant ground plane. Thus, a homographic correspondence is achieved between any position of an image from a camera and the position in real world coordinates. This simple calibration meets the criteria of scalability and adaptability of FitOptVIS. This software is fully implemented, and is available as open source in the following github repository: https://github.com/JuanIsernGhosn/homography-calibrator.

4.1.7 Related documents

[NGU2016] D.T. Nguyen, L. Wanqing, P.O. Ogunbona, "Human detection from images and videos: A survey", Pattern Recognition 2016.

[SDC+2014] A. Smeulders, W.M. Dung, M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, "Visual tracking: An experimental survey." IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014.

[IMG2017] U. Iqbal, A. Milan, J. Gall, "Posetrack: Joint multi-person pose estimation and tracking", IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[XSB+2017] T. Xiao, L. Shuang, W. Bochao, L. Liang, W. Xiaogang, "Joint detection and identification feature learning for person search." IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[WR2007] B. Wu, N. Ram, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors." International Journal of Computer Vision, 2007.

[HT2010] S.U. Hussain, B. Triggs. "Feature sets and dimensionality reduction for visual object detection." British Machine Vision Conference, 2010.

[OE2009] P. Ott, M. Everingham. "Implicit color segmentation features for pedestrian and object detection." IEEE International Conference on Computer Vision, 2009.

[LWW+2018] P. Li, D. Wang, L. Wang, H. Lu, "Deep visual tracking: Review and experimental comparison", Pattern Recognition, 2018.

[NOL2011] D.T. Nguyen, P. Ogunbona, W. Li, "Human detection with contour-based local motion binary patterns." IEEE International Conference on Image Processing, 2011.

[LMS+2008] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, "Learning realistic human actions from movies", IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[SAS2007] P. Scovanner, S. Ali, M. Shah, "A 3-dimensional sift descriptor and its application to action recognition", ACM international conference on Multimedia, 2007.

[ZP2007] G. Zhao, M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions." IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007.

[CLK2016] L. Čehovin, A. Leonardis, M. Kristan, "Visual object tracking performance measures revisited", IEEE Transactions on Image Processing, 2016.

[TPM2007] O. Tuzel, F. Porikli, P. Meer, "Human detection via classification on riemannian manifolds", IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[SJ2010] L. Shao, L. Ji, "A descriptor combining MHI and PCOG for human motion classification", ACM International Conference on Image and Video Retrieval, 2010.

[ADF2010] B. Alexe, T. Deselaers, V. Ferrari, "What is an object?", IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[BS2008] K. Bernardin, R. Stiefelhagen, “Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics”, Journal on Image and Video Processing, 2008.

[TMB+2016] D. Tomè, F. Monti, L. Baroffio, L. Bondi, M. Tagliasacchi, S. Tubaro, “Deep Convolutional Neural Networks for pedestrian detection”, Signal Processing: Image Communication, 2016.

[HWW+2018] T. Hao, Q. Wang, D. Wu, J.S. Sun, “Multiple person tracking based on slow feature analysis”, Multimedia Tools and Applications, 2018.

[ZYW2019] J. Zhang, Y. Yuan, Q. Wang, “Night Person Re-Identification and a Benchmark”, IEEE Access, 2019.

[RHG+2017] S. Ren, K. He, R. Girshick, J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, IEEE Transactions

on Pattern Analysis and Machine Intelligence, 2017. https://doi.org/10.1109/TPAMI.2016.2577031

4.2 Behavior classifier [UGR]

UGR will implement this component in the framework of FitOptiVis to be used in the monitoring of the elderly at home. This component detects indoor actions performed from a video stream. It will be able to recognize potentially risky situations such as falls.

4.2.1 Short description

This component is a library for action classification using machine learning techniques, that recognizes a human action performed over a video sequence that contains a complete action execution. The component is intended to be trained indoor for monitoring the elderly at their own house (Habit Tracking UC). We have trained several neural networks that perform action recognition, but having different trade offs between accuracy and energy consumption. Only one model will do the action inference at a time.

Target:

In order to perform in real-time, the component is developed for a GPU (Graphic Processing Unit), using Python, NVIDIA Cuda C++ for high-performance computation, and Machine Learning tool sets such as TensorFlow.

Input:

The behavior classifier input is a set of consecutive frames also called a video snippet that shows the complete execution of an action.

Output:

The output is a label that describes the recognized human action. The actions are determined depending on the task to be performed: for the Habit Tracking UC potential actions are watching TV, walking, cooking, and potential interesting action to trigger alarms such as accidental fall, fainting, or lying on the floor.

The action classifier is not run constantly, it requires sequences of frames to the action inference. Therefore, we plan to log the actions that have been performed in a concrete period of time, showing which neural model was responsible for doing the inference of the video feed, and the performance in terms of number of frames that the system can analyze per second. It should also show at least the two actions that have the higher confidence. An example of what we expect is shown in Table 6.

Table 6 Snippet of output results for action recognition. Five rows from left to right: Date, performance for prediction in frames per second, ML (Machine Learning) model used for the inference, list of action labels and confidence levels. In this example, the ML model is changed to improve confidence of action labels.

date and hour	performance [FPS]	model	action – confidence	action – confidence
2020-02-24 11:51:26	20.14	rgbi3d-original	stand up - 84.47%	sit – 6.92%
2020-02-24 11:51:29	19.11	rgbi3d-original	blowing nose – 61.56%	eat – 12.58%
2020-02-24 11:51:32	21.76	rgbi3d-original	hit – 17.83%	eat – 17.19%
2020-02-24 11:51:34	23.08	rgbi3d-reduced	eat – 36.78%	lying on bed or sofa – 27.25%

2020-02-24 11:51:37	24.52	rgbi3d-reduced	walk – 51.70%	eat – 6.92%

4.2.1.1 Updates during the second year of the project

During the second year (M12 to M24), some updates and modifications have been made to this component. These are the following:

- We have focused attention on developing a behavior classifier that is capable of identifying indoors actions and not outdoors. Therefore, this component now only concerns UC3.
- We created a custom and heterogeneous dataset from more than five action recognition datasets like Kinetics [KCS+2017], UCF-101 [SZS2012] or HMDB [KJG+20E11]. We collected from them over 17 indoor actions with about 19000 video clips.
- We have trained and compared the performance between several neural network models proposed. We have made use of the Two Stream I3D [CZ2017] architecture, because it enables us to train two networks, one fed with RGB data, and the other with an Optical Flow estimation. These are independent networks, but if we fuse both outputs, we can get a better accuracy in inference.
- In order to reduce the load of the system on the embedded platform (NVidia Jetson TX2 or Xavier) where it will run, we plan to compute remotely the estimation of the Optical Flow when it is necessary to confirm if a critical action like a fall has taken place or not. As said above, performing the inference with a RGB and an Optical Flow neural net provide us better accuracy than when using them separately. To achieve this we will use pocl-remote from TAU. This tool allows us to do calculations on an external GPU over the network in a transparent way through OpenCL. As shown in Figure 7, we can speed up the optical flow computation when using an RTX2080 Ti GPU with pocl-remote in comparison with the GPU and CPU of the Jetson TX2. In this way, we can also reduce the **energy consumption** on the embedded device.

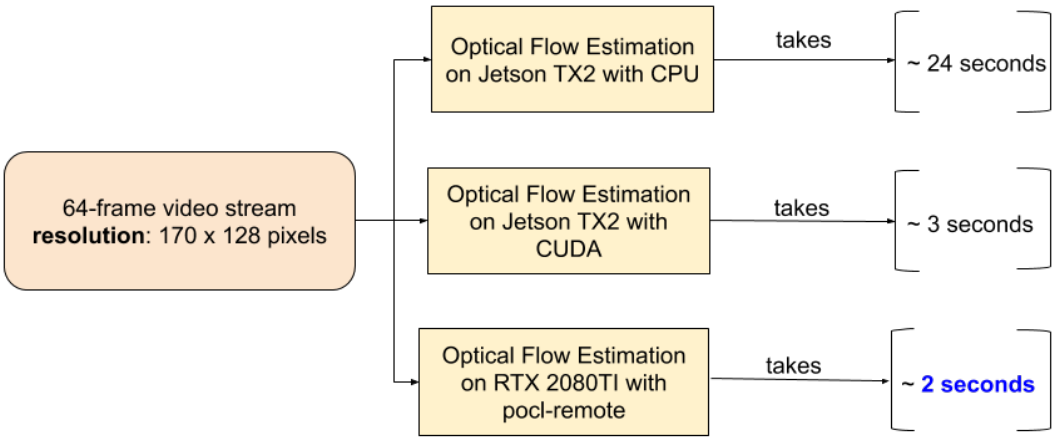


Figure 7 Compare time required to estimate Optical Flow TV-L1

During this year we have designed several scenarios where the system needs to be reconfigured. The reconfigurations aim to obtain acceptable accuracy considering the power consumption. One way to do this is to change the neural model that is detecting actions for the one that fits better in every scenario.

4.2.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

With respect to the qualities, we are taking into account accuracy, performance, and Precision-Recall curves (of F-measure). Accuracy using P-R curves defines the relationship between the precision or fraction of positive labels that are correct, and recall or fraction of real positives were correctly labelled. A combination of precision and recall is given by the F-measure or F-score, basically defined as a harmonic mean. The performance is given by the number of actions labelled per second, and depends on the size of the video input streams and the number of labels or classes initially considered, since they determine the complexity of the network. We also consider the number of frames that the network is capable of processing per second as a performance measure. With respect to the set points, some of the potential features to be tuned are given by the following trade-offs: a) precision vs recall; b) classification accuracy vs performance; c) number of considered classes or labels.

We show in Figure 8 the precision vs recall curve for 4 of our trained models. We can see how the Two Stream model, the one that fuses the results between the RGB and Optical Flow Stream, is the one that has a higher area below the curve. Thus, is the one that offers a better trade off between labelling correctly the video with the corresponding action, and not labelling it in the wrong class.

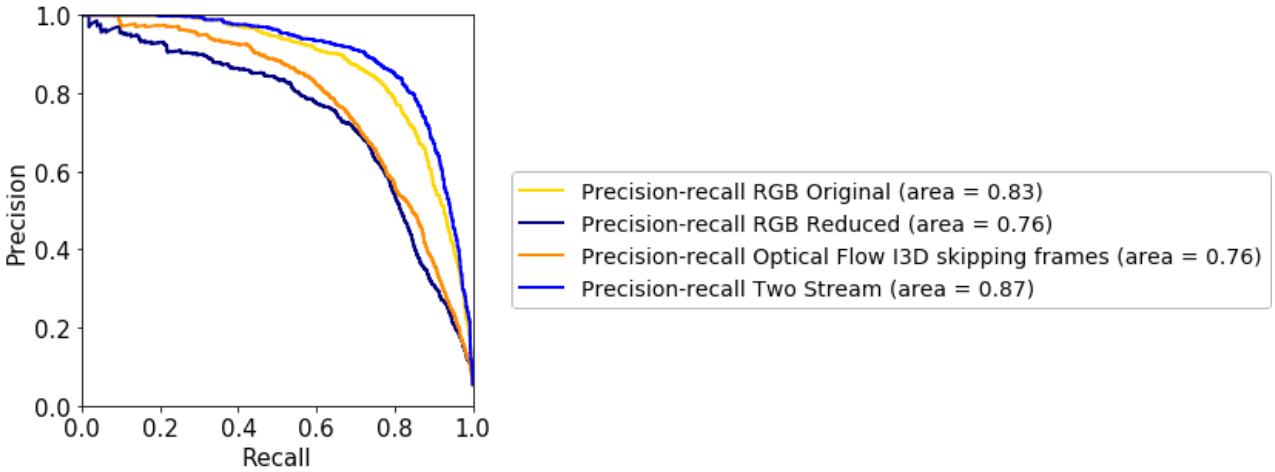


Figure 8 Precision vs Recall curve of neural networks models

Next, in Figure 9, we can appreciate the trade-off of F1-measure with the performance of the neural networks in terms of labeled actions per second. We can see how the more complex model, which is the one that obtains a higher value of F1, but the performance is the worst compared to the other models. With all this information, we should choose which neural model should be analyzing the video in the Habit Tracking System, depending on whether we want the highest possible accuracy, or whether we want to reduce energy consumption.

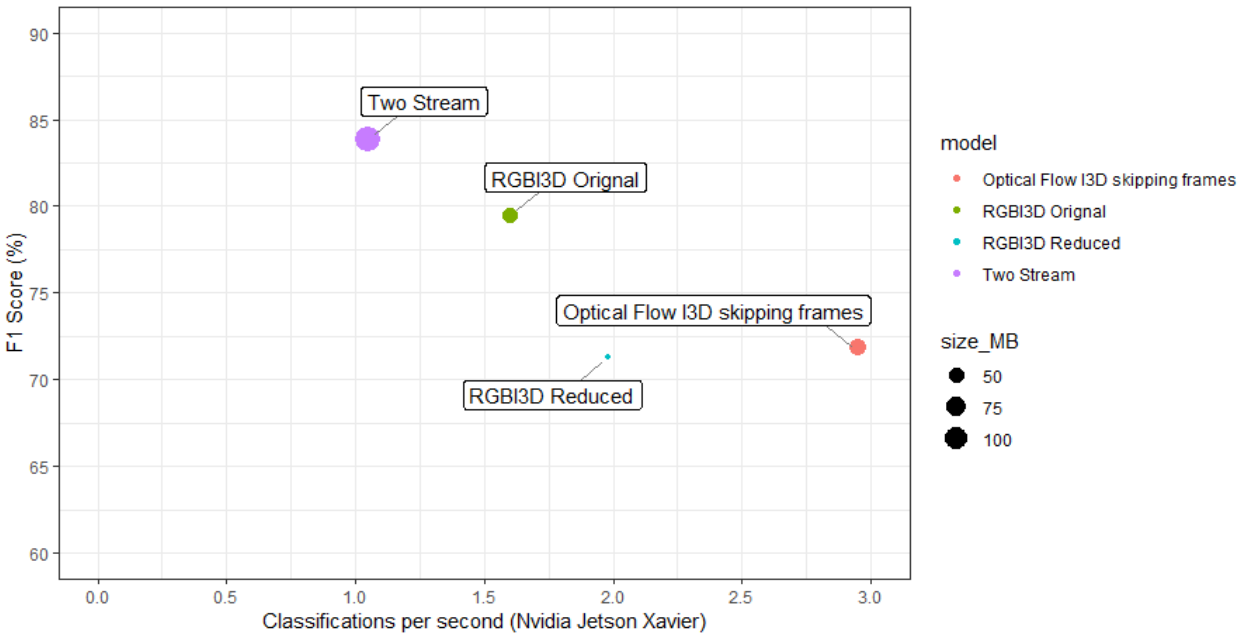


Figure 9 F1-measure vs labeled actions per second in Jetson Xavier

4.2.1.3 Design-time support

In order to develop this component, general design and development tools are used to implement the CPU and GPU code. Python is the programming language mainly used

4.2.1.4 Run-time support

Monitoring:
Indoors (action/micro-action classification of the elders at home). The scenarios features are design-time configurable, but we plan to allow run-time adaptation depending on the scenario.
Programming APIs:
Python, NVIDIA GPU boards, and CUDA package C++ programming language, TensorFlow package for machine learning application

4.2.2 Related state of the art

Human actions are performed pursuing some purposes. Understanding human actions through a vision system and determining its purpose is quite relevant for many applications. The definition for action ranges from a simple hand movement to complex actions composed by multiple simple actions (or microactions), performed over a sequence of consecutive frames that usually last a few seconds [KF2018].
Action recognition entails 1) the action representation that extracts features from the video sequence and 2) the action recognition or classification that is fed with the action vector and infers an action label.
The action representation component requires features that extract discriminative information while ensuring invariance against illumination changes, camera perspective, pose, and speed. Some methods use holistic representations such as Motion History Volumes, shape features [GBS+2007] or silhouette [YS2005]; others work on local representations defining local motion in space-time such as 3D Histograms of Oriented Gradients [KMS2008], histograms of optical flow [LMS+2008], or Motion Boundary Histograms [WKS+2013]. With respect to the action classifiers,

some examples include nearest-neighbor clustering [LP2007], Hidden Markov Models [IF2007], Support Vector Machines [WWX+2012]. However, the most successful approaches are based on deep learning approaches and require large datasets instead of hand-crafter features. In this case, both components are merged into the same networks, and are substituted by the convolutional operation to extract features frame by frame, and the temporal modeling. There a three main approaches: 1) applying 3D convolutions to consecutive frames capturing the temporal dynamics from them [JXY+2013]; 2) modeling temporal dynamics using a stream for optical flow (Flow Net) while the other captures 2D spatial features [FP2016, GRG+2017]; 3) using temporal pooling to capture temporal information and use 2D convolutions for the spatial features, aggregating by using an LSTM model on top of a 2D ConvNet [YHV+2015].

4.2.3 Commercial positioning

N/A.

4.2.4 Extension within FitOptiVis

Current methods for action classification reach accuracies between 40-50% in average, but only when trained in specific datasets with a large number of examples for each class, and with actions that are generic and easily discriminated [WKS+2013].
With respect to the performance, it ranges from 1-2 fps for large datasets with a dozens of classes [WKS+2013]. We expect accuracies of 50% - 60% when dealing with the actions specific to the use cases for which the component is developed. We also expect higher time performance around 25 fps.
TRL@M0:
TRL1 – At M0 only the model was defined, no other library/code was available. Only state of the art pre-trained models are available, but not tailored.
FitOptiVis Extensions:
At M12, the first version of the learning network is setup and partially trained on different datasets. It reaches accuracy close to 50%. With respect to the performance, it is still not running in real time.
At M24, we have trained several neural networks. We are reaching an accuracy over 80% in some models. As for the performance, we are close to get a performance in real time at 25 frames per second.
TRL@M36:
TRL4/6 - At M36, we expect a tested library for action classification for different scenarios, included in the system demonstrators for the Smart Grid and the Habit Tracking UCs.

4.2.5 Prospective adoption

Applied in FitOptiVis to the habit tracking UC.
Other fields of application not considered in the project: human-robot interaction, video retrieval or video scripting, gaming industry, or autonomous driving vehicles.

4.2.6 Intellectual Property information

One of the most important elements of our behaviour classifier component are the neural network models that perform indoor action recognition from a video stream. Once we have developed, designed and trained our final neural network architecture, we will release it as open source on GitHub. The repository can be found in the following link:

https://github.com/DaniDeniz/indoor-action-recognizer. Figure 10 shows an example of the output tags of the videos analyzed with the associated confidence. Eight different actions are shown that the Habitual Tracking System is able to recognize. We can see how there are some classes that the system identifies with more confidence, and others, like the one shown in red (Lying on the sofa), is wrong. The action being performed is walking, but because the system is seeing some sofas in the scene, it outputs the wrong action. This is something we will be working on in the next few months.

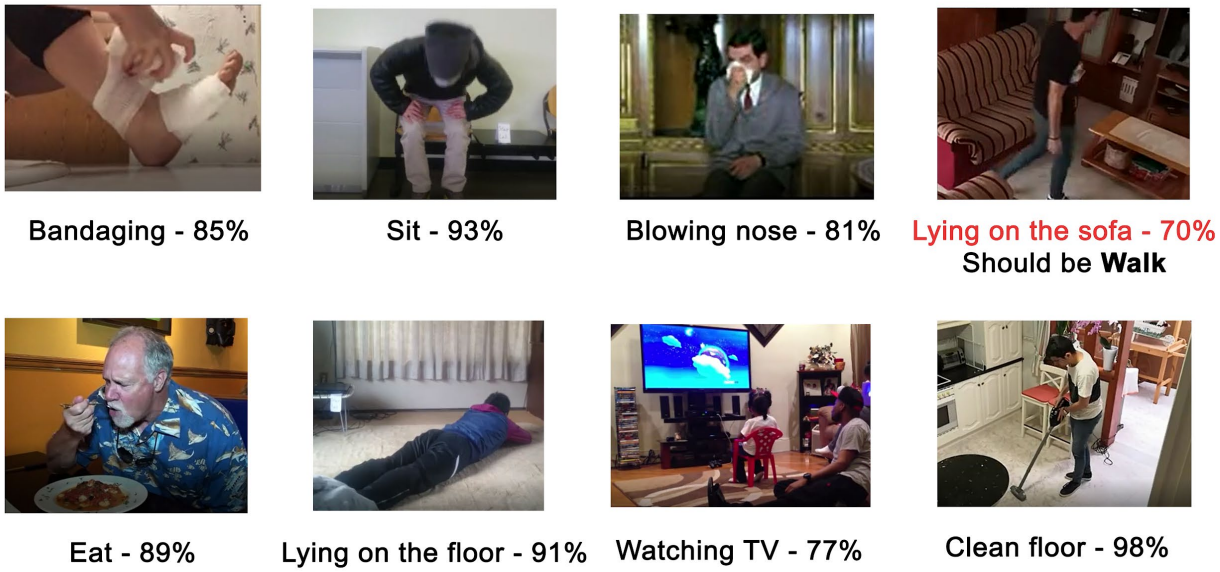


Figure 10 Detecting actions with associated confidence

We expect to write a scientific paper explaining the methodology followed to develop the presented neural network. Once the scientific paper gets accepted, we will upload the mentioned action recognized model to the GitHub repository

4.2.7 Related documents

[KF2018] Y. Kong, Y. Fu, "Human Action Recognition and Prediction: A Survey", Computer Vision and Pattern Recognition, 2018.

[GBS+2007] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, "Actions as space-time shapes", IEEE transactions on pattern analysis and machine intelligence, 2007.

[YS2005] A. Yilmaz, M. Shah, "Actions sketch: A novel action representation", IEEE, 2005.

[KMS2008] A. Klaser, M. Marszałek, C. Schmid, "A spatio-temporal descriptor based on 3d-gradients", British Machine Vision Conference, 2008.

[LMS+2008] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, "Learning realistic human actions from movies", IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[WWX+2012] Z. Wang, J. Wang, J. Xiao, K.H. Lin, T. Huang, "Substructure and boundary modeling for continuous action recognition", IEEE Computer Vision and Pattern Recognition, 2012.

[LP2007] I. Laptev, P. Pérez, "Retrieving actions in movies", IEEE International Conference on Computer Vision, 2007.

[IF2007] N. Ikizler, D. Forsyth, "Searching video for complex activities with finite state models", IEEE Conference on Computer Vision and Pattern Recognition, 2007

[FP2016] C. Feichtenhofer, A. Pinz, A. Zisserman, "Convolutional two-stream network fusion for video action recognition", IEEE computer vision and pattern recognition, 2016.

[GRG+2017] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, B. Russell, "Actionvlad: Learning spatio-temporal aggregation for action classification", IEEE Computer Vision and Pattern Recognition, 2017.

[YHV+2015] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici. "Beyond short snippets: Deep networks for video classification", IEEE conference on computer vision and pattern recognition, 2015.

[WKS+2013] H. Wang, A. Kläser, C. Schmid, C.L. Liu, "Dense trajectories and motion boundary descriptors for action recognition", International journal of computer vision, 2013.

[CZ2017] J. Carreira, A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset”, IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[KCS+2017] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, “The kinetics human action video dataset”, Computer Vision and Pattern Recognition, 2017.

[KJG+20E11] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, “HMDB: a large video database for human motion recognition”, International Conference on Computer Vision, 2011.

[SZS2012] K. Soomro, A.R. Zamir, M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild”, Computer Vision and Pattern Recognition, 2017.

[JXY+2013] S. Ji, W. Xu, M. Yang, K. Yu, "3D convolutional neural networks for human action recognition", IEEE transactions on pattern analysis and machine intelligence, 2013.

4.3 Salmi Care Platform [HURJA]

HURJA’s AR-based (Augmented Reality) Salmi Care Platform is capable of motivating rehabilitation patients to make daily exercises by utilizing AR-based gamification techniques, assisting rehabilitation patients, brain damage patients, elderly people in their daily tasks, and monitoring daily activities and vital signs of patients as well as automatically alerting nurses/relatives in case of emergency. Rehabilitation patients, brain damage patients, and elderly people are wearing AR-glasses (HoloLens II [HOLOLENS]) as User Interface for Salmi Care service. Service can be used via voice commands and/or gestures. Service also enables patients to communicate with nurses/doctors/relatives/peers via video calls that are directly shown on AR-glasses.
AR-glasses as UI makes it very motivating, effective, and comfortable for users, since many different kinds of text-based, graphics-based (image/video), and/or voice-based options can be utilized for giving exercise instructions and other instructions related to daily activities of patients. Boring indoor exercise location can be changed into more comfortable and motivating environment of patients’ choice, such as walking path in the forest, hiking trail around the lake, idyllic countryside, etc. Patients can live several years

longer at home with help of Salmi Care service. The service also provides context-awareness and vital signs monitoring with automatic alerts in case of emergency, thus relieving workload of nurses/relatives. Benefits of Salmi Care service are:

- Significant improvement in rehabilitation rates.
- Huge cost savings through increased efficiency of human resources utilization and significant reduction in hospitalization times.
- Better quality-of-life for patients.
- Extremely motivating, easy/convenient to use, and gives patients/users WOW Effect.
- Same system can be used both by patients and their caregivers/relatives.

4.3.1 Short description

Rehabilitation patients, brain damage patients, and elderly people are wearing AR-glasses (HoloLens II [HOLOLENS]) that assists/reminds/motivates them to do different important daily tasks, such as taking medicines, making daily rehabilitation exercises, time to eat/sleep, how to make tea/coffee, how to brush teeth as well as having nurse/doctor/relative calls directly through AR-glasses, communicating with other peers (i.e., other brain damage patients or elderly people), monitoring users health condition, and automatically alerting nurse/relatives in case of emergency. Wireless indoor positioning will be utilized to enable more accurate information to be shown on AR-glasses. Service will be used through AR-based mobile application running on tablet or smart phone.
Salmi Care Platform component has three subcomponents: Extent Mobile Application, Salmi Care Mobile Application, and Salmi Care Website. A mobile application called Extent can upon request download a JSON packet which consists of a list (descriptions) of wakeup images, objects, entities, and actions. Either the request can come from the Salmi Care Website (if service is used through the browser of PC or Laptop), from the Salmi Care Mobile Application (if service is used through native application running on smart phone or tablet), or directly from the Extent Mobile Application if the “free roam” state has been switched on (requires GPS). End-users have the option to switch the “free roam” state off at any time and when this happens, the Extent Mobile Application downloads new content only upon request from an external source (currently only the Salmi Care Platform related sources are available). The Extent mobile application downloads all required wakeup images, 3D-models, textures, audio files, videos, etc. based on the instructions received via JSON packet. Figure 11 illustrates an overview of HURJA’s Salmi Care Architecture.

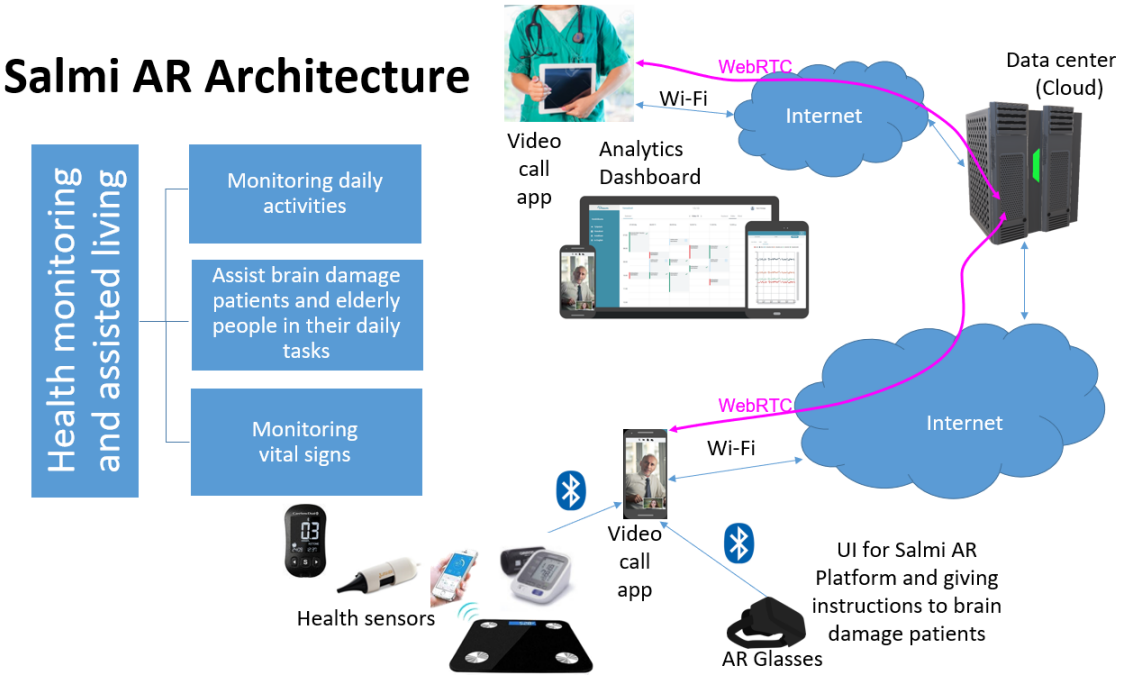


Figure 11 Overview of HURJA's Salmi Care Architecture.

4.3.1.1 Updates during the second year of the project

We started the development work of our Salmi Care Platform at M0 when TRL-level of our system was 2-3. Potential application was then validated in terms of market needs and development of initial version of Salmi Care service Proof-of-Concept started having only assisted living part in Salmi Care service during M0-M12. However, during M13-M24 we managed to get also first version of rehabilitation part implemented into our system, since we negotiated new pilot location at Vetrea Ltd premises in Kuopio, Finland. Both rehabilitation and assisted-living features will be ready by M26 and it will put our system TRL-level at 4-5. Thus, after M26 we can start pilot testing for both assisted-living and rehabilitation use cases. Moreover, context-awareness, vital signs monitoring, and automatic alerts features will be ready by M30 and it will then put our system TRL-level at 5-6. Finally, video call (point-to-point and group calls) features with relatives/nurses/doctors/peers will be ready by M32 and it will put our system TRL-level at 6-7. A complete Salmi Care service will be available at M36, already tested in real world environments with real end-user pilots and thus Salmi Care service will be ready for production use with several different customers when the project ends. It is worth mentioning that during M0-M18 there were huge availability problems for getting high-quality state-of-the-art AR-glasses for our Salmi Care Platform development and thus the development work was a bit restricted during the first half of the project and therefore we needed to utilize earlier versions of AR-glasses (i.e., HoloLens 1 vs HoloLens 2) to start the development work. Furthermore, after M18 COVID-19 pandemics also have had effect on high-quality AR-glass availability, but fortunately we managed to get some new state-of-the-art AR-glass models in order to speed up our development work and catch up most of the lost time.

4.3.1.2 Qualities tuneable set-points and available design-time and run-time parameters

Qualities:

- Near real-time (soft real-time) performance: System performance is monitored/evaluated in terms of frames-per-second and kilobits-per-second. It is worth noting that AR-feature robustness/performance depends highly on the selected AR-glass model. We have made our development work using state-of-the-art HoloLens II AR-glasses to ensure that all possible use cases can be implemented easily. Later on we plan to investigate the use of other (cheaper and less powerful) AR-glass options that may require more optimization of the system code to achieve the level of performance comparable with the high-end, state-of-the-art AR-glasses.
- Optimal energy usage: It is not an easy task to calculate the energy usage for the whole Salmi Care system, since continuous camera feed and required advanced algorithms will present a challenge in terms of optimizing the energy usage of the system as a whole. Thus, we have so far performed only initial measurements on power usage and based on the achieved initial results, we will make adjustments to the implemented algorithms to enable optimal energy usage of Salmi Care system.

Set Points:
Near real-time performance vs Quality vs Energy usage vs Data amount.
Run-time performance monitoring:
Yes
Available model:
N/A
Reconfiguration in the model:
N/A

4.3.1.3 Design-time support

Salmi AR Platform will be developed using Unity 3D, C#, JSON, and PHP. Generic IDEs (Eclipse, Netbeans, etc.) will be used to handle component design, development, and verification.

4.3.1.4 Run-time support

Run-time adaptivity support:
To optimize the run-time performance of the Salmi Care system, all of the packets can be downloaded in advance. All files will be saved locally into end-users mobile device (smart phone or tablet) and those will be shown to end-users based on instructions received via JSON packets as soon as matching wakeup image/object/entity/action is found or when an end-user is within a certain pre-defined distance from the target. Free roam data will be removed on-the-fly from end-users' devices when each session ends. *Extent mobile application* is currently being developed using C# programming language on top of Unity 3D and back-end side is currently being developed using PHP. During our early testing phase, all description packets are in JSON format. The Salmi Care service is intended to be used indoors at Homes or Private Care Homes for Rehabilitation & Assisted Living Services of patients.
Monitoring:
Includes measured performance and energy usage, which can be handled by a generic data model. Relevant metrics to be monitored/evaluated are the near real-time (soft real-time) performance and optimal energy usage. In most of our use cases near real-time performance and low power usage are mandatory requirements and thus we will use our monitoring results to improve our system.

4.3.2 Related state of the art

N/A

4.3.3 Commercial positioning

Many different kinds of assisted living systems/services are common nowadays, but none of them utilize yet advanced AR-based glasses for system smart UI and on top of that provide also fully voice comunication based control of all system features. Thus, our system will be completely novel in this field.
Market positioning for Salmi Care Platform falls into Healthcare & Wellbeing domain being Assisted Living & Rehabilitation type of service for Homes and Private Care Homes rather than being Medical related service with Medical Certified Devices in Hospitals. Global AR-Based Healthcare markets are expected to reach $10.82 Billion by 2025 and AR-based markets are expected to continue to grow at 38% rate annually [BW2019]. Our Revenue targets for Salmi Care service after the end of FitOptiVis project will be 1 M€ for 2022, 3 M€ for 2023, 6 M€ for 2024, and 10 M€ for 2024. There are not currently any big competitors, since AR-Based rehabilitation and assisted living has not been so far possible in the extent our Salmi Care system can provide these services due to the lack of good enough AR-glass models. Microsoft HoloLens II will be the first good enough state-of-the-art AR-glass available to fulfill our demanding requirements for Salmi Care service features.
Our extensive competitor & state-of-the-art analysis revealed that there are no currently available systems on the market that can potentially compete with our Salmi Care service when it will be ready by end of FitOptiVis project mainly due to lack of good enough AR-glass models. Below are the main findings of our competitor & state-of-the-art analysis:

- Victoria University, New Zealand published on 06/2018 the article ”Augmented Reality Could Give Stroke Survivors’ Rehabilitation a Motivational Boost” [SF2018]:
    - According to authors, AR-based hand-held game could help motivate stroke patients through their rehabilitation. In the game rehabilitation patient needs to feed a little bird called Fizzy by throwing blueberries for it to eat. However, the only way to feed the Fizzy is through exercise. Authors believe that Fizzy could be a breakthrough for stroke rehabilitation by fullfilling differing levels of rehabilitation needs. The more Fizzy eats, the more active it gets and thus the exercise becomes gradually a little bit harder. The game has been so far pilot tested in Auckland stroke patients and the results of the pilot were overwhelmingly positive according to authors. The pilot test clearly showed how this kind of games can be very good motivators for rehabilitation patients to do their daily exercises. [SF2018]
- University of Pisa, Center for Rehabilitative Medicine, Italy and Kettering University, USA published on 04/2019 the article ”Review of the Augmented Reality Systems for Shoulder Rehabilitation” [VCT+2019]:
    - According to authors, current studies show the need for new rehabilitation tools for upper extremity, since traditional interventions are less effective than in other body regions. Authors studied on what extent AR-applications are used in shoulder rehabilitation, examining wearable/non-wearable technologies employed, and investigating the evidence supporting the effectiveness of AR. Total of nine different AR systems were identified and analyzed in terms of tracking methods,

visualization technologies, integrated feedback, rehabilitation setting, and clinical evaluation. Results show that all these systems utilize vision-based registration, mainly with wearable marker-based tracking, and spatial displays. None of the systems use head-mounted displays and only one system (11 %) integrates a wearable interface for tactile feedback. Three systems (33 %) provide only visual feedback, 66 % present visual-audio feedback, and only 33 % provide visual-audio feedback, 22 % visual-audio with biofeedback, and 11 % visual-audio with haptic feedback. Moreover, several systems (44 %) are designed primarily for home settings. Three systems (33 %) have been successfully evaluated in clinical trials with more than 10 patients, showing advantages over traditional rehabilitation methods. Further clinical studies are clearly needed to generalize the obtained findings in order to support the improved effectiveness of the AR applications compared with the traditional methods. Authors conclude that clinical studies conducted show clear benefit over traditional rehabilitation methods in terms of usability, enjoyability, and motivation and in terms of improving of exercise performance outcomes.

- Angelos Karatsidis, Rosie E. Richards, Jason M. Konrath, Josien C. van den Noort, H. Martin Schepers, Giovanni Bellusci, Jaap Harlaar, and Peter H. Veltink have published the article “Validation of Wearable Visual Feedback For Retraining Foot Progression Angle Using Inertial Sensors and an Augmented Reality Headset” [KRK+2018]:
    - Authors propose the setup that is completely wearable and enables gait retraining applications in clinical settings without the need for a complex gait motion analysis laboratory. Wearable biofeedback setup consists of Xsens MVN Analyze software [XSENSa] that receives the MTw sensor data via the Xsens Awinda Station [XSENSb], reconstructs the lower body kinematics, and streams it via UDP. Microsoft HoloLens receives the kinematic input via Wi-Fi, calculates the foot progression angle, and updates the holographic feedback visualization.
- Other relatively new research articles on the topic worth mentioning are “Virtual and Augmented Reality in the Treatment of Phantom Limb Pain: A Literature Review” [DYM+2017], “Usability Assessments for Augmented Reality Motor Rehabilitation Solutions: A Systematic Review” [CSG+2018], “Wrist Rehabilitation System Using Augmented Reality for Hemiplegic Stroke Patient Rehabilitation: A Feasibility Study” [PKK+2019], “Intelligent Assistive Technologies for Dementia – Augmented Reality Assisted Dementia Care” [ZON2019], “Upbeat – Augmented Reality-Guided Dancing for Prosthetic Rehabilitation of Upper Limb Amputees” [MHC+2019], and “Effectiveness of Virtual and Augmented Reality-Enhanced Exercise on Physical Activity, Psychological Outcomes, and Physical Performance – A Systematic Review and Meta-Analysis of Randomized Controlled Trials” [NMH+2019]. More new research papers on the topic can be found directly at:

4.3.4 Extension within FitOptiVis

TRL@M0:
TRL 2-3: Potential application validated in terms of market needs and development of initial version of Salmi AR service Proof-of-Concept started.
Expected Developments/New Features:

Rehabilitation patients can be motivated to make their daily exercises via AR-based gamification techniques and brain damage patients can perform their daily tasks, such as making tea/coffee or brushing teeth, which would not be normally possible without help of assistive technologies. Rehabilitation and Assisted-Living features will be ready by M26 and it will put our system TRL-level at 4-5. Elderly people can live several years longer at home with help of Salmi Care Platform. Our platform provides context-awareness (i.e., what exactly rehabilitation patients, brain damage patients, and elderly people are doing at any given moment) and vital signs monitoring with automatic alerts in case of emergency, thus relieving workload of nurses/relatives. Context-awareness, vital signs monitoring, and automatic alerts features will be ready by M30 and it will put our system TRL-level at 5-6. Video calls (point-to-point and group calls) with relatives/nurses/doctors/peers when needed (and even by automatic/forced manner in case patient or elderly people is not in good enough condition to make calls themselves). Video calls feature will be ready by M32 and it will put our system TRL-level at 6-7.
TRL@M36:
TRL 7: A complete Salmi AR service will be available at M36, tested in real world environments with real end-user pilots and thus Salmi AR service will be ready for production use with several different customers when the project ends.

4.3.5 Prospective adoption

Hurja will contribute to both habit tracking and VR/AR use cases with its Salmi Care Platform. In addition to assisted living, rehabilitation, and remote health-care services for brain damage patients, elderly people, and rehabilitation patients, Salmi AR service can be used for any kind of domain-independent remote consultation services by using its video connection (point-to-point and group calls) and advanced AR/MR/XR features.

4.3.6 Intellectual Property information

No actual IPR protection (patents, etc.) needed yet, but our Salmi Care Platform is protected by the standard software and server based protection techniques, i.e., Data Security is taken care of by protecting the confidentiality and integrity of data in transit and storage; Network Security is taken care of by protection (including access control, availability, and integrity) of network resources, devices, and services (including bandwidth, connectivity, and platforms); and Software Security is taken care of by protecting software (including data in software) against unauthorized access, modification, analysis, or exploitation. At the moment we are using the following open source tools for Salmi Care Platform development: ARCore, .NET Core, and C# Compiler. Other tools (that are not open source) used for Salmi Care Platform development are Unity 3D, WikiTude, Vuforia, JSON, and PHP.

4.3.7 Related documents

[HOLOLENS] HoloLens 2. Online: https://www.microsoft.com/en-us/hololens/hardware

[BW2019] “Global $10.82 Billion Healthcare Augmented Reality and Virtual Reality Market Trend Forecast and Growth Opportunity to 2026” Business Wire, 2019.

[SF2018] “Augmented Reality Could Give Stroke Survivors' Rehabilitation a Motivational Boost”, Stuff Fibre, 2018.

[VCT+2019] R.M. Viglialoro, S. Condino, G. Turini, M. Carbone, V. Ferrari, M. Gesi, ”Review of the Augmented Reality Systems for Shoulder Rehabilitation”, Information, 2019.

[KRK+2018] A. Karatsidis, R.E. Richards, J.M. Konrath, J.C. van den Noort, H. M. Schepers, G. Bellusci, J. Harlaar, P.H. Veltink, “Validation of Wearable Visual Feedback For Retraining Foot Progression Angle Using Inertial Sensors and an Augmented Reality Headset”, Journal of NeuroEngineering and Rehabilitation, 2018.

[XSENSa] Xsens: Xsens MVN Analyze. Xsens 3D Motion Tracking. Online: https://www.xsens.com/products/xsens-mvn-analyze/

[XSENSb] Xsens: Xsens MTw Awinda. Xsens 3D Motion Tracking. Online https://www.xsens.com/products/mtw-awinda/

[DYM+2017] J. Dunn, E. Yeo, P. Moghaddampour, B. Chau, S. Humbert, “Virtual and Augmented Reality in the Treatment of Phantom Limb Pain: A Literature Review”, NeuroRehabilitation, 2017.

[CSG+2018] V.C. Cavalcanti, M.I. de Santana, A.E.F. da Gama, W.F.M. Correia, “Usability Assessments for Augmented Reality Motor Rehabilitation Solutions: A Systematic Review”, International Journal of Computer Games Technology, 2018.

[PKK+2019] H.L. Phan, J.P. Kim, K. Kim, C.H. Hwang, K. Koo, “Wrist Rehabilitation System Using Augmented Reality for Hemiplegic Stroke Patient Rehabilitation: A Feasibility Study”, Journal of Applied Sciences, 2019.

[ZON2019] M.Y. Zhao, S.K. Ong, A.Y.C. Nee “Augmented Reality - Assisted Dementia Care”, Intelligent Assistive Technologies for Dementia, 2019.

[MHC+2019] M. Melero, A. Hou, E. Cheng, A. Tayade, S.C. Lee, M. Unberath, N. Navab, “Upbeat - Augmented Reality-Guided Dancing for Prosthetic Rehabilitation of Upper Limb Amputees”, Journal of Healthcare Engineering, 2019.

[NMH+2019] Y.L. Ng, F. Ma, F.K. Ho, P. Ip, K. Fu, “Effectiveness of Virtual and Augmented Reality-Enhanced Exercise on Physical Activity, Psychological Outcomes, and Physical Performance – A Systematic Review and Meta-Analysis of Randomized Controlled Trials”, Journal of Computers in Human Behavior, 2019.

4.4 Distributed video coding solutions [UNIVAQ]

Current video compression standards, (i.e., MPEGx or H.26x) are characterized by higher computational complexity at the encoder than the decoder. The asymmetry design is tuned to broadcasting or streaming video-on-demand scenarios in which the video is compressed once and then decoded several times. In other scenarios (i.e., wireless video sensors for surveillance) a different distribution of the computational complexity is requested. Furthermore, this section describes the proposed Distributed Video Coding (DVC) component, based on the results of the DISCOVER (DIStributed Coding for Video sERvices) European project.

4.4.1 Short description

This software tool implements an algorithm for Distributed Video Coding (DVC) in order to exploit the correlation among videos recorded by multiple cameras, but also reducing

the bandwidth required by a single video stream, considering CPU cycles and power consumption.

Target:

Target technology will be a function of the chosen video HW components (cameras), which will also identify the Language. MATLAB will be used at the decoder. The test scenario will be configured with an ARM Cortex-A53 processor, while performance will be evaluated by specific test benches.

Input:

The input will be the incoming frame sequences (i.e. videos from cameras) acquired by one or several cameras set on the UC scenario.

Output:

The component output are compressed video frames, split into key frames (i.e., intra encoder H.264 classical coding) and Wyner-Ziv (WZ) frames.

4.4.1.1 Updates during the second year of the project

Given the specific application of DVC to use case 1, second scenario, UNIVAQ is working on the proper configuration of distributed video coding parameters for the case of a single video stream. In particular, a DISCOVER like codec solution has been chosen to be completely implemented, since, after a deep state of the art, it appears to be one the most advanced implementation of distributed video coding. Nevertheless, once the implementation will be complete, UNIVAQ is setting up a performance evaluation procedure in order to establish the properness of distributed video coding solutions for the particular scenario.

It is also to be noticed that two runtime parameters aroused while going through the DVC implementation: correlation model and number of GOPs (group of frames).

4.4.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Quality will be guaranteed in terms of optimal power consumption, increased coding efficiency, reliability through robust transmission.

Tuneable parameters are given by the online adaptation of the correlation model but also by the codec to be exploited depending on the UC requirements.

4.4.1.3 Design-time support

The main selected programming language is C++ and the VStudio environment. Starting from the open-source code available under GPL on git, the adaptation for embedded platform will be released before the end of the project. MATLAB will be used as the reference environment to analyse rate-distortion (RD) results and compare solution with other possible classical video coding methods.

4.4.1.4 Run-time support

The use of DVC component inside the FitOptivis UC is under investigation. To guarantee best effort performance the component will consider an adaptive monitoring and quality manager able to tune DVC parameters following DISCOVER approach. Adaptive parameters for DVC are: correlation model and number of GOPs. The resolution and video rate will be adapted respect to the network requested, the edge capabilities and the cloud computation support. Bit streams between encoder and decoder will be adapted with a feedback loop inside the main component infrastructure.

4.4.2 Related state of the art

There are two main approaches in literature to DVC:
1) Berkley (PRISM) [PR2002]: block based with motion estimation at the decoder.
- Enhancement to PRISM is based on multi-level coset codes as proposed in [MR2004].

2) Standford (pixel domain Wyner–Ziv architecture) [GAR+2005]: frame based, which has gained much popularity because of its comparatively better rate-distortion performance, the most famous version is called DISCOVER.
- Enhancements are focused on modules such as
  - rate-control, [DS2009], [MPP+2007]
  - decoding algorithm, [TTT+2006], [RMP+2007], [ASW2008], [TTT+2005]
  - side information generation, [NBA+2005]
  - correlation noise modelling, [BAP2006].

Based on the analysis performed in [DGT+2010] relevant topics related to DVC are:
- *Coding efficiency*: as already stated, DVC outperforms H264/AVC intra coding method except for scenes characterized by complex motion. Nevertheless, the performance remains significant lower than a full H264/AVC codec. In literature different solutions have been proposed: spatial smoothing, refinement of motion vectors, use of forward and backward motion vectors, subpixel accuracy concept, exploitation of bit plane decoding for motion vectors remaining.
- *Complexity*: several iterations are required at the decoder to converge to a solution. In order to solve this issue, some solutions encompass: a hybrid encoder-decoder, parallel implementation of various channel decoding algorithms, change in the complexity of operations at the viable node.
- *Robust transmission*: the decoding should be performed to be successful even if packet losses occur. Among the solutions available in literature, the encoding of auxiliary frames only for some frames, exploitation of unequal error protection mechanisms.
- Scalability: due to the absence of any closed loop in its design DVC supports codec independent scalability that can be achieved through: development of an enhancement layer, posing the problem as a variant of the WZ side information problem.
- Multiview: compared to monoview the difference is that the SI can be coded not only form the previously decoded frames in the same view, but also from other views. Many solutions have already been proposed like: disparity compensation view prediction (DCVP), Multiview Motion Estimation (MVME), exploitation of a homography model, View Synthesis Prediction (VSP).

In [UBF2010] a comparison between H264/AVC and DVC encoders power consumption is presented with promising results reporting that, for available DVC implementations and considering the power consumption/compression efficiency ratio, when comparing to compression algorithms based on differential frame coding (with zero search radius for ME) the advantages of DVC are not so evident.
A comparison in terms of rate-distortion curves between H264/AVC and DISCOVER (a well known DVC solution) is carried on in [CCO+2013]. The results show that the DVC codec may outperform the traditional intra codec; however, the performance is still far behind that of the inter codec (H.264/AVC SP). Another interesting observation is that the performance highly depends on the characteristics of video sequences. DVC may outperform H.264/AVC No Motion for a sequence where the global motion cannot be addressed well by the simple zero motion vector scheme; however, for a sequence coming from a static camera for surveillance applications, H.264/AVC No Motion

outperforms DVC and H.264/AVC Intra codec. These results show that existing DVC systems can be further improved. The power consumption analysis is also carried on in [CCO+2013] where both processor-based and ASIC-based platforms are analysed. The analysis of the results of the processor-based platform shows that DVC technique can provide a power-efficient solution. Concerning ASIC-based platform, it shows that although the power consumption of a DVC encoder is only 7% of an H.264/AVC Simple Profile encoder, the power efficiency of DVC is still very similar to that of H.264/AVC No Motion codec because of the poorer coding efficiency.
It seems that the DISCOVER codec is one of the best performing DVC schemes reported in literature [DGT+2010], but no open source implementations as well as practical applications on Water Supply Networks (WSNs) are available. Our work in this filed will be the adaptation of DISCOVER implementation to a WSN (edge-oriented) scenario, using embedded platforms and cloud-edge communications. The codec will be integrated and validated with specific test benches and will be introduced inside the FitOptivis UCs (i.e., agriculture UC), only if the requested performance will be reached.

4.4.3 Commercial positioning

N/A

4.4.4 Extension within FitOptiVis

TRL@M0:
TRL4
Expected Extensions:
DVC solutions will be tested within FitOptiVis framework with the main purpose of guaranteeing efficient and reliable information exchange. In particular the analysis of WSNs for video monitoring based on DVC will be carried on in order to reduce requirements about: processing, bandwidth and power consumption.
TRL@M12
TRL4
Expected Extensions:
Since currently there are no practical implementations of DVC solutions for surveillance systems, UNIVAQ aims to carry out an effective DVC based solution for the specific scenario 2 of Use Case 1, in order to evaluate the effectiveness of DVC solutions with respect to traditional codecs where the encoder is much more complex than the decoder. State-of-the-art, alternatives and parallel solution have been considered, analysed and reported as a starting point for the DVC component definition and implementation.
TRL@M24
TRL5
Expected Extensions:
The DSL component interfaces definitions have been completed. The DISCOVER implementation has been considered as a concrete DVC implementation. The code adaptation and the performance evaluation are under investigation. Final prototype will be released in the ongoing months.
TRL@M36:
TRL6
Expected Extensions:
Final DVC component implementation, and possible integration inside FitOptivis UC.

4.4.5 Prospective adoption

Water supply (Use Case 1)

4.4.6 Intellectual Property information

A DISCOVER like source base code has been released under GPL license, so we maintain the specific IP license, while trying to integrate such C++ application on a specific embedded target platform (i.e., ARM Cortex-A53).

4.4.7 Related documents

[PR2002] R. Puri, K. Ramchandran, “PRISM: a new robust video coding architecture based on distributed compression principles.” Conference on Communication, Control and Computing, 2002.

[GAR+2005] B. Girod, A.M. Aaron, S. Rane, D. Rebollo-Monedero, “Distributed video coding”, IEEE, 2005.

[MR2004] A. Majumdar, K. Ramchandran, “PRISM: an error-resilient video coding paradigm for wireless networks”, International Conference on Broadband Networks, 2004.

[DS2009] B. Du, H. Shen, “Encoder rate control for pixel-domain distributed video coding without feedback channel”, International Conference on Multimedia and Ubiquitous Engineering, 2009.

[MPP+2007] M. Morbee, J. Prades-Nebot, A. Pizurica, W. Philips, “Rate allocation algorithm for pixel-domain distributed video coding without feedback channel”, IEEE International Conference on Acoustics, Speech and Signal Processing, 2007.

[TTT+2006] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, “Intra mode decision based on spatio-temporal cues in pixel domain Wyner–Ziv video coding”, IEEE International Conference on Acoustics, Speech and Signal Processing, 2006.

[RMP+2007] A. Roca, M. Morbee, J. Prades-Nebot, E.J. Delp, “A distortion control algorithm for pixel-domain Wyner– ZIV vide coding”, Picture Coding Symposium, 2007.

[ASW2008] A. Avudainayagam, J.M. Shea, D. Wu “Hyper-trellis decoding of pixel-domain Wyner–Ziv video coding”, IEEE Transactions on Circuits and Systems for Video Technology, 2008.

[TTT+2005] A. Trapanese, M. Tagliasacchi, S. Tubaro, C. Brites, F. Pereira, “Embedding a block-based intra mode in frame-based pixel domain Wyner–Ziv video coding”, International Workshop on Very Low Bitrate Video Coding, 2005.

[NBA+2005] L. Natário, C. Brites, J. Ascenso, F. Pereira, “Extrapolating side information for low-delay pixel-domain distributed video coding”, International Workshop on Very Low Bitrate Video Coding, 2005.

[BAP2006] C. Brites, J. Ascenso, F. Pereira, “Studying temporal correlation noise modeling for pixel based Wyner–Ziv video coding”, IEEE International Conference on Image Processing, 2006.

[UBF2010] A. Ukhanova, E. Belyaev S. Forchhammer, "Encoder power consumption comparison of distributed video codec and H.264/AVC in low-complexity mode", International Conference on Software, Telecommunications and Computer Networks, 2010.

[CCO+2013] S.Y. Chien, T.Y. Cheng, S.H. Ou, C.C. Chiu, C.H. Lee, V.S. Somayazulu, Y.K. Chen, "Power Consumption Analysis for Distributed Video Sensors in Machine-to-Machine Networks", IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2013.

[DGT+2010] F. Dufaux, W. Gao, S. Tubaro, A. Vetro, “Distributed Video Coding: Trends and Perspectives”, Journal on Image and Video Processing, 2010.

4.5 Point localization from stereovision [UWB]

Pose estimation and point localization in 3D are important parts of many complex AI and computer vision tasks. Significant and most problematic part of the commonly used techniques is point registration between the pair of stereo images. The presented algorithm can avoid it because of a priori known point correspondence.

4.5.1 Short description

The component is a computation module that computes real-world 3D coordinates of points of interest. For these points are known 2D image coordinates (coordinates in perspective projection image) in 2 images taken from 2 different viewpoints – stereovision. Additionally, a set of reference points with known real-world 3D coordinates as well as coordinates from the 2 perspective projections is supplied.

Input

Function arguments passed in shared memory or over TCP/IP stream, described in Table 7.

Table 7 Function arguments passed in shared memory or over TCP/IP stream.

N_R	Number of reference points (RP)
N_P	Number of points of interest (POI)
RP_{World}	Matrix N_Rx3 with [x,y,z] real-world coordinates of the reference points
RP_{Img1}	Matrix N_Rx2 with [x,y] image coordinates of the reference points in the first projection view (extracted from 1st camera image)
RP_{Img2}	Matrix N_Rx2 with [x,y] image coordinates of the reference points in the second projection view (extracted from 2nd camera image)
POI_{Img1}	Matrix N_Px2 with [x,y] image coordinates of the POIs in the first projection view (extracted from 1st camera image)
POI_{Img2}	Matrix N_Px2 with [x,y] image coordinates of the POIs in the second projection view (extracted from 2nd camera image)

Output

Results passed out in shared memory or TCP/IP stream, described in Table 8.

Table 8 Results passed out in shared memory or TCP/IP stream.

POI_{World}	Matrix N_Px3 with [x,y,z] real-world coordinates of the POIs
R	The location and number of reference points and location of the 2 cameras involve the precision and stability of the computing algorithm (dependability of the system of equations). This result measures the quality of the produced output coordinates.

4.5.1.1 Updates during the second year of the project

UWB has not been working on the component during Y2.

4.5.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

With respect to the module objectives, accuracy, stability and performance measures are taken into account. At first, depending on application preferences (precision vs performance) computing using double-precision floating point numbers (64-bit) or single-precision floats (32-bit) can be used.
As second - the form of used algorithms is not determined yet, but probably iterative algorithms will be used. In this case, conditions on convergence, step deltas or number of iteration cycles can be applied.

4.5.1.3 Design-time support

The first version of the component is to be designed and implemented in Microsoft Visual Studio and compiled with the provided MS C++ compiler or with GNU C++ compiler (for ARM based HW platforms).
Additionally, the used REX block SDK provides framework and base class implementation for the component. Only functional parts of the component have to be implemented - functions Init(), Exec() and Exit() (called upon system initialization, periodically during system run and upon system shutdown).

4.5.1.4 Run-time support

In run-time the module is configured and observed via configuration window accessible through RexDraw (newly re-branded to REXYGEN Studio) - GUI based application that configures and inspects configuration of the real-time control application [REXYGENa], [REXYGENb].
Component set points can be inspected and adjusted here, the results of the last evaluation (POIs coordinates and result reliability) are also available. Optionally, the values obtained from this or any other block in the modular scheme can be used to automatically adjust the module parameters.

4.5.2 Related state of the art

Nowadays most of the 3D localization procedures use key-point detection algorithms executed above stereo image data and followed by point registration algorithms to pair corresponding image points [F1994], [EL2003], [LHM2000]. The subsequent 3D localization algorithms must count with inaccurate registration, as it is the most problematic part of the processing chain here. Using the novel proposed approach of adopting active synchronized LED markers and differential imaging, it is possible to ensure 100% correspondence between point pairs in stereo image and use straight computing algorithms to localize the POIs in 3D.

4.5.3 Commercial positioning

N/A

4.5.4 Extension within FitOptiVis

TRL at M0:
TRL 0.

TRL at M36:
UWB expects to deliver designed and fully implemented component that is tested on real industrial robots in lab as well as in controlled industrial environment (TRL 5/6).

4.5.5 Prospective adoption

The component will be used within the FitOptiVis demo pool as one of the principal components in the robot calibration use case (UC8).
Outside the project scope it can be used in any application where 3D point localization from stereo images is required. A needed prerequisite is successful and 100% accurate point registration, which can be acquired for example using the other proposed component – synchronized LED markers, and differential imaging.

4.5.6 Intellectual Property information

No IP protection is planned at this time. The research results will be published and made available for public. REX intends to extend the component functionality and encapsulate the algorithms into function block of their RexVision computer vision platform and use it in the Robot Calibration Demo (UC8).

4.5.7 Related documents

[REXYGENa] Getting started with REXYGEN and Raspberry Pi. Online: https://www.rexygen.com/doc/ENGLISH/MANUALS/RexygenGettingStarted_RasPi/RexygenGettingStarted_RasPi_ENG.html

[REXYGENb] REXYGEN Studio User Guide. Online: https://www.rexygen.com/doc/ENGLISH/MANUALS/RexygenStudio/RexygenStudio_ENG.html

[F1994] O. Faugeras “Three-Dimensional Computer Vision. A Geometric Viewpoint”, MIT Press Ltd, 1994.

[EL2003] R. Elias, R. Laganière, “Projective Geometry for Three-Dimensional Computer Vision”, Multiconference on Systemics, Cybernetics and Informatics, 2003.

[LHM2000] C.P. Lu, G.D. Hager, E. Mjolsness, “Fast and Globally Convergent Pose Estimation from Video Images”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.

4.6 Video content analysis (VCA) platform [AITEK]

Video content analysis is the capability of automatically analyzing video to detect and determine temporal and spatial events, which can be used to trigger context switch, more-advanced processing in the systems or, in some cases, specific alarms.

4.6.1 Short description

The VCA platform includes a C++ based software modules suite implementing Video Analytics techniques for real-time detection of different events involving the safety and security of the monitored areas focusing on people, vehicles and static objects. These software modules are able to process live images from video-surveillance cameras and to perform post-event analyses of recorded footage and acquire statistical data.
Input:

Images recorded by cameras (including the ones installed on a drone)
Output:
Alarms after each event detection. Still images, video streams and related metadata that show the detected event
Composability:
The platform is composed by the following sub-components

- IP / analog cameras;
- IP connection towards the Network Video Recorder (NVR).
- NVR/HDVR devices hosting video-surveillance and video-analytics SW;
- Other optional components like video management application, decoders and client.[2]

It is worth noticing that VCA software can be executed on the camera, on the NVR or partially on camera partially on NVR, following a hybrid approach. In this last case, the camera takes care of preliminary processing steps, generating metadata processed by the NVR to detect the events.

4.6.1.1 Updates during the second year of the project

During the second year of the project, AITEK has worked on the definition of the video processing algorithms, focusing in particular on the detection of intruders inside restricted area under the context of Water Supply use case. Starting from this general goal, AITEK has identified the specific information that should be detected by analysing acquired images and therefore it has defined and developed algorithms needed to this purpose. Such functionalities are listed below:

- Intrusion detection and classification – according to end user requirements, it is needed not only to detect intruders but also to distinguish between human and animals. In fact, as already reported in WP1 deliverables (D1.1 and D1.2), that information is crucial to implement the correct countermeasures, as currently used system cannot do, generating several false alarms. The video processing solution under development detect any moving targets inside restricted area and classify them.
- Tracking and “movements analysis” – the presence of a human intruder inside a restricted area is not sufficient for an efficient security monitoring. As a matter of fact, the whole restricted area inside the water supply infrastructure consists of different critical areas where the presence of an intruder and/or his movement may represent an even more dangerous situation. So, the video processing solution under development by AITEK will be able also to track and analyse target movements and presence in some specific areas (e.g. to detect if she/he is standing close to a water pump or if she/he is running away/towards it).

Video processing solution from Aitek implements some of the FitOpTiVis pillar concepts, in particular referring to autonomous adaptivity and resource versus performance optimization.
System adaptivity has been considered in terms of different algorithms used in order to obtain the best compromise between monitoring performance and resource needs (in terms of bandwidth and computational power). In more detail, some video processing algorithms are based on traditional VCA approaches (less performant but less resource demanding) some others are based on neural networks (more performant but more resource demanding). Our proposed solution can autonomously switch from one

[2]These components will be described in the WP6 as they could be used in the demonstrator but they are not developed/improved during FitOptiVis.

processing mode to the other according to some detected events that increase or reduce the dangerousness of the situation.
Currently, in AITEK we are studying a cooperative strategy among the cameras to increment the accuracy of the monitoring system by orienting and zooming the cameras to the position where an intruder has been detected. This strategy is under definition and its eventual deployment will be done during the last year of the project.

4.6.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Quality of the video stream is a fundamental parameter for the VCA platform. It is expressed mainly in terms of frame rate and frame quality. As a matter of fact these parameters has a dramatic impact on the event detection accuracy.
As explained in the next section, the offered load (i.e. the bandwidth needed to transmit the video flow) is another important set-point which has to be considered according to the bandwidth available in the channel.
Considering the Pareto theory, the vector <*frame rate and frame quality*> represents the space in which the Pareto analysis should be done. For each point in this space corresponds an offered load that has to be lower or equal to the available bandwidth (on the network) which represents a boundary for the feasibility region that is an upper limit for frame rate and frame quality.
At the same time, frame rate and frame quality should be enough to guaranty sufficient VCA performance, defining another boundary of the feasibility region. Similar considerations (regarding the Pareto analysis) can be done for the computational resources needed to execute the processing algorithms which may impact the supported frame rate and the frame quality.

4.6.1.3 Design-time support

The VCA platform, or at least some of its components can be used as a test case for the accelerating tools developed by other FitOptiVis partners.

4.6.1.4 Run-time support

Monitoring:
Monitoring of the offered load compared to available bandwidth. This information is used to adapt frame rate and frame quality to maintain offered load lower than available bandwidth, as previously explained.
Runtime Adaptivity:
Run-Time adaptivity is currently under investigation. In particular, the focus is on the definition of mechanisms to adapt the characteristics of the video flows in terms of frame rate and frame quality at runtime. The system periodically monitors the offered load: if it exceeds the available bandwidth, the system triggers a modification of the video stream in order to reduce the offered load. Moreover, we are also studying similar approaches considering the computational resources available and resources required by video processing algorithms.
Programming APIs:
Some programming APIs are supported by the system. In particular it has JSON-REST API (https). It has also an SDK in C. Moreover, similar API JSON-REST (https) are available specifically to configure the Video Content Analysis algorithms.

4.6.2 Related state of the art

There are two different approaches for video processing:

- Traditional algorithms, in which the video flow is analyzed considering a sequence of correlated frames. Currently used approach includes a low level processing that consists in background and foreground separation, a middle level processing that produces as output a list of detected targets and a final processing that generates alarms analyzing the characteristics of such targets.
- Innovative algorithms based on deep learning theory. In this case the analysis can be done frame by frame independently. In practice, events detection can be done analyzing just a single frame. Some examples include Single Shot Detection (SSD) [LAE+2006], Convolutional Neural Network (CNN), Recursive Neural Network (RNN), YOLO [RF2016] are some example of deep learning approaches currently used for image processing. Performance comparison among these algorithms is not an easy job. Nevertheless, a quite good comparative analysis is reported in [H2018]. These new approaches are targeted as possible improvement of the VCA platform within FitOptiVis.

4.6.3 Commercial positioning

The video content analysis platform can be used in several different scenarios, in which video surveillance for safety and/or security purpose is needed.
Some of them regards *i)* critical infrastructures monitoring, *ii)* road and traffic monitoring, *iii)* Industry4.0 etc.
Our goal, as described in the next section, is to improve its capability and the detection performances by using innovative deep learning based approach, while moving the processing as close as possible to the edge. These improvements are crucial to consolidate the position of AITEK in a market dominated by big plyers like Axis (Sweden), Bosch (Germany), Hikvision (China), Dahua Technology (China), VivoteK (Taiwan), Honeywell Security (US), to mention some of them.

4.6.4 Extension within FitOptiVis

TRL@M0
TRL4: Baseline ongoing experimentations in lab in order to assess innovative video processing approaches based on deep learning algorithms.
Expected Extensions
Progresses during FitOptiVis, regard the integration of such new algorithms in our video monitoring solutions. These new processing algorithms may require a quite large amount of computational resources and re-configurability capabilities in case of insufficient resources available.
TRL@M36
TRL6/7: The goal is to integrate such new video processing approaches in our video security system in order to test them in relevant environments (i.e. the water supply system provided by SAT).

4.6.5 Prospective adoption

The proposed component will be used in water supply system demonstrator in UC1. The goal is to define and implement (at prototype level) an advanced security system able to detect intruders in dangerous areas, trying also to infer what he/she is doing, at least in a quite preliminary way. As explained in the previous section, this component will be generic enough to be used in other different applicative domains.

4.6.6 Intellectual Property information

The video processing solution and more precisely the software modules developed during the project will become crucial assets for AITEK. They will be included in AITEK portfolio of products, being integrated into its video surveillance platform which is covered by copyright. Therefore, such project results will be protected in the same way.

4.6.7 Related documents

[LAE+2006] W. Liu, D. Anguelov, D. Erhan, C. Szegedy,S. Reed, C.Y. Fu, A.C. Berg, "SSD: Single Shot MultiBox Detector", Computer Vision and Pattern Recognition, 2006.

[RF2016] J. Redmon, A. Farhadi, "YOLO9000: Better, Faster, Stronger", Computer Vision and Pattern Recognition, 2016.

[H2018] J. Hui, "Object detection: speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3)", 2018. Online: https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359

4.7 Face and behavioral analysis (FBA) platform [UNISS]

The FBA platform aims to cover the entire "traditional" face analysis pipeline, including alignment, normalization and classification of the detected faces. Whenever possible, well established and available techniques will be used; additional modules will be developed and tested in order to improve both face detection and recognition. Finally, the platform will include some modules specially designed to increase the efficiency of recognition and to perform behavioral analysis of the subjects who appear in the video stream.

4.7.1 Short description

This software component, mainly developed in C/C++, will be aimed at analyzing live video sequences characterized by narrow to medium field of view and controlled or partly controlled light conditions. The main purpose of the component will be the detection and the automatic recognition of faces; more in detail

Principal key features and properties

- Efficient and multiple face detection under a large variety of poses;
- Face watch-list in order to verify whether a face is part of a known set of people;
- Basic behavioral analysis based on the temporal track of faces.

Processed inputs and produced outputs

The module will process video frames with arbitrary resolution. Detected faces and specific behaviors will be annotated through metadata (bounding boxes and labels) that will constitute the output of the component.

Composability

The module will involve a number of sub elements (processing cores) useful to extract specific image features. Most of these elements will be organized in form of C/C++ libraries.

4.7.1.1 Updates during the second year of the project

No significant updates of the workplan to report in the second year of the project. The development work is proceeding smoothly with the development of the various modules of the processing pipeline. In particular:

- a first version of the facial detector has been released, based on integral images and including best variants of Viola-Jones;
- a first version of the alignment module has been released, based on coarse eyes detection and model-based refinement;
- a definitive version of the intensity normalization module has been implemented, based on local mean/variance standardization;
- the salient point extraction has been implemented, based on the pyramidal computation of the Hessian matrix;
- the implementation of descriptors and the development of an original comparison among descriptors is ongoing.

With respect to the original plan, and in order to better exploit available OS code, some modules have been developed in JAVA language instead eg. C/C++. No problems related to this decision are expected.

4.7.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Qualities:

The component will manage a number of tuneable features:

- frame rate (per second) and frame quality (image resolution);
- maximum number of tracked faces;
- upper/lower limits for valid face-box dimensions;
- quick change detection (QCD) based sensitivity
- image watch-list.

All the above qualities will be tuneable at run-time, except for the watch-list that will be loaded at start-up.

Set-Point(s) and Parameters:

The component will usually work at full frame rate and full frame resolution. Different set points will correspond to integer sub-sampling in time or space.

Face-box dimensions will be given in percentage with respect to full frame resolution while sensitivity will be expressed in percentage with respect to optimal MAP estimate.

4.7.1.3 Design-time support

Generic IDEs (Eclipse, Netbeans) will be used to handle component design, development and verification.

4.7.1.4 Run-time support

Run-time adaptivity:

Triggers based on the execution time of the component will be used for adaptation. To this purpose, output metadata will include timestamps associated to each processed frame. Timestamps analysis will offer to third party software the possibility to redefine set points and parameters in order to counterbalance low computational resources.

Monitoring:

The component will monitor continuously the time required to complete the processing of a single frame.

Programmability and Programming API:

Programmability support of the component will be offered by C/C++ APIs.

4.7.2 Related state of the art

Face detection has made significant progress in the last decade and a large number of face segmentation algorithms based on different assumptions and applications has been reported in the literature. For example, template matching methods use standard face patterns and derive from correlation a unique index stating the possible presence of a face. These methods are commonly unable to deal with variation in scale, pose and shape but can perform well in controlled applications like airport gates. Bottom-up methods are feature-based approaches that start from basic features (color, contours, salient points) in order to infer the presence of a face. They usually can better manage for variation in scale but still suffer for variation in pose and shape. Holistic (appearance based) methods rely on templates and statistical analysis techniques; in this case, however, templates are learned from example images. Neural networks and Support Vector Machines (SVM) belong to this class so as the Viola-Jones face detector [VJ2004], a well-known technique based on integral images which has made possible face detection in real time.
Face recognition share a number of basic tools with face detection. Feature based methods use features, geometric rules, graph matching and deformable templates. Many approaches have been proposed to localize and describe facial features (SIFT, SURF, GaborJets, LBPs among others) with very interesting results also in case of significant variation of scale and pose [ZR2012]. Holistic methods use the whole image and a set of statistical and mathematical techniques (PCA, ICA, LDA, SVM) useful to better separate classes and improve the robustness to noise and discretization. Recent CNNs and DeepCNNs are holistic approaches based on networks with multiple convolution and pooling layers and complex learning techniques [ZCB+2017]. They proved to perform very well in most practical cases, but they require long learning times and large number of training images. Innovation brought by FitOptiVis has to do with three main aspects:

- design and development of a fast face detection module inspired by Viola Jones but more related to facial features and more accurate for limited rotations of the head
- design and development of a face recognition module based on a new concept of similarity between facial features [YK2018];
- design and development of a basic classifier able to learn from continuous observation and to correctly identify a limited number of behaviours [BG2019].

4.7.3 Commercial positioning

N/A

4.7.4 Extension within FitOptiVis

TRL@M0:
TRL3: A preliminary study and simple experimental proof-of-concept demonstrated the possible development of face detection and recognition algorithms deeply involving facial features. Current accuracy for face detection depends on the reference database used. For complex databases like LFW or IJB-C overall accuracy is under 90%. By fixing standard FAR (0.001) the verification rate is usually under 75%.
Current accuracy for face recognition depends on the reference database used and on the test protocol adopted. For complex databases like LFW or IJB-C and image-restricted protocols, accuracy ranges between 92% and 97% (on pre-detected faces). At the standard FAR (0.001) the verification rate is usually under 55%.

Expected extensions:
Development of a new algorithm for face detection/recognition, exploiting facial features and pose estimation. Full evaluation of the computational complexity of such new algorithm. Integration of
the algorithm in the video monitoring solutions of the company (AITEK) and test in the foreseen UC
TRL@M36:
6/7: Technology integrated in the video security system of the company. Technology demonstrated in a relevant UC.

- Expected accuracy for face detection on complex databases like LFW or IJB-C over 90%.
- Expected verification rate at standard FAR (0.001) over 80%.
- Expected accuracy for face recognition on complex databases like LFW or IJB-C over 95%.
- Expected verification rate at standard FAR (0.001) over 60%.

4.7.5 Prospective adoption

The component will be adopted in Use Case 1: water supply system.
The component is generic enough to be used for a number of potential applications like:

- Soft monitoring of public areas.
- Critical infrastructures monitoring (gates and physical access).
- Attendance registration.
- Smart Factories.

4.7.6 Intellectual Property information

All the development is carried on under GNU GPLv3, thus the component will use open source features (libraries, modules) both developed under GPL/LGPL or weak copylefts. IP will be considered solely at the “component” level.

4.7.7 Related documents

[VJ2004] P. Viola, M.J. Jones, "Robust real-time face detection", International journal of computer vision, 2004.

[ZR2012] X. Zhu, D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild", IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[ZCB+2017] A: Zadeh, Y. Chong Lim, T. Baltrusaitis, L.P. Morency, “Convolutional experts constrained local model for 3d facial landmark detection”, IEEE International Conference on Computer Vision Workshops, 2017.

[YK2018] K. Yordanova, F. Krüger, “Creating and exploring semantic annotation for behaviour analysis”, Sensors, 2018.

[BG2019] M. Bicego, E. Grosso, “On the importance of local and global analysis in the judgment of similarity and dissimilarity of faces”, Image and Vision Computing, 2019.

4.8 Pose estimation engine [HIB]

Systems that are used to track persons using video data captured from cameras often require information that is more detailed that just positioning. Pose estimation engines have developed quite rapidly in the last years using Deep Learning technologies. In FitOptiVis we will provide a component that combines state-of-the-art pose estimation functionality with distributed processing to provide computing flexibility.

4.8.1 Short description

Pose Estimation Engine is a software component that takes video with recordings of persons and generates an estimation of the pose (position of the extremities such as legs, arms and head) of the persons present.

Target:

The software is based on CMU Openpose, mostly coded using C++ and Python. We have also a wrapper so it can be used (with degraded performance) using Javascript from a regular internet browser, which is in principle not used in the FitOptiVis demonstrator.

Input:

Video streams from cameras. As of the moment, for the UC3 system being deployed for FitOptiVis, we are using Foscam FI9900EP Full HD POE cameras that we have been analyzing during the first year.

Output:

A 2D image overlay of the points defining the pose of persons in a frame of video plus an array detailing these points. An example of these can be seen in the Figure 12.

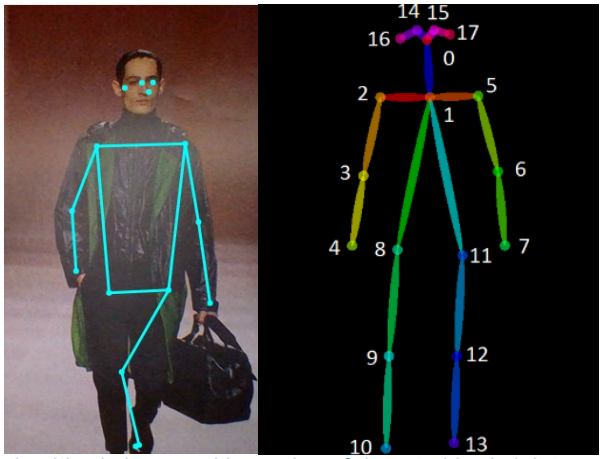


Figure 12 Left: recognized body image with overlay of detected body joints using Openpose. Right: complete model for body recognition in Openpose.

4.8.1.1 Updates during the second year of the project

During the second year of FITOPTIVIS, the Pose Estimation engine has received some optimizations and improvements, mostly to align better with the runtime adaptation work proposed in WP4:

- Implementation of the software in its target hardware (nVidia Jetson Nano).
- Implementation of mechanisms for energy optimization based on monitoring of the battery status and throttling of the hardware.
- Integration with the rest of components of UC3.

During the last part of the project the objective is the evolution of the component to improve the performance, both in recognition features and in energy efficiency.

4.8.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

The quality parameters of the image and detection systems are as follow:

- Frames per second: the number of frames per second that are processed to deliver the results. From 15fps acceptable results can be had but for fine movements up to 60fps are useful. Beyond that value, human movement and pose estimation doesn't get any better for the usual movements of interest (e.g., humans performing actions at home), but could be used for more rapidly moving objects.
- Resolution: the resolution of the frames passed over to the system. We are working from VGA (640x480) to 2K images (1920x1080). The required resolution depends on the spatial resolution achieved from the placement of the camera.
- Number of tracked points: the person model is based on a series of joints for the skeleton (e.g., knee, wrist, elbow) and points for the face in the case of facial recognition. Higher number of points yields finer grained results but require more computational power to be processed in real time.
- Number of tracked persons: The system can detect features for more than one simultaneous person in the field of view. This can be configured and is subject of the same limitations as the complexity of each model.

The adaptivity of the system is described in D4.2 and D4.3 and consists of a set of rules that dynamically change the performance and energy consumption of the hardware (changing the clock frequency and maximum current draw) to maximize the duration of the batteries (using a separate UPS and battery controller board Geekworm NVIDIA Jetson Nano 18650). This enables the system to run on its own energy (simulating a battery powered scenario) and to adapt the performance both to the user needs in each moment and to the remaining battery in the power pack.

4.8.1.3 Design-Time Support

At design time, during the Y1 and Y2 we have just used general IDEs (Eclipse, etc.) and the successive iterations of the QRML modelling language.

4.8.1.4 Run-Time Support

The component is programmed using regular Deep Learning libraries (Tensorflow on Linux/Android targets plus experiments with Tensorflow using a Javascript backend). This uses GPU optimization by default. Current line of work is to enable runtime adaptivity to match real-time changing resources, in particular in terms of energy usage (some components may run on batteries) and used computational power.
Monitoring of the runtime performance is done using the following tools:

- Internal tracing to detect the current frames per second and use of the `tergrastats` command to monitor the energy consumption of the system.
- Communication with the Geekwork UPS daughterboard using the i2c protocol and the GPIO pins in the board.
- Throttling of the nVidia Jetson clock frequencies with the `jetson_clocks` and the `nvpmodel` commands.

4.8.2 Related State of the Art

For detection of persons and their actions in video streams we use the following assets:

- Frameworks: Keras [KERAS], CNTK [CTNK], TensorFlow [TENSOR]
- Models: CNN, LSTM [GBC2016]
- Datasets: Violent Flows datasets [HIK2012], UCF Web Abnormality [MOS2009], UMN Abnormal Crowd datasets, Stanford Question Answering Dataset [RZL+2016]

The ability to detect anomalies in real-time is very valuable, so that appropriate actions can be taken as soon as it is detected to avoid or reduce negative consequences.
Most research works focus on hand engineering features for particular scenes or datasets, but these features need to be manually tuned each time a different scenario is introduced. We propose an automated video surveillance system to accurately determining pose of persons in a scene.
Body position detection state of the art is OpenPose[3] [CHS+2018] (using the Caffe framework and C/C++), which also has a Python wrapper to facilitate its use (PyOpenPose[4]). It is the combination that we currently have in our commercial LIFEonLive surveillance system (see section 4.9.5), with which we get 18 points as maximum in the model. There is also an alternative implementation of Tensorflow models in Javascript (tfjs-posenet [STA+2019]). With this, recognition can be done entirely in-browser.

4.8.3 Commercial Positioning

HI Iberia is currently planning or using pose estimation networks in its commercial and pre-commercial products:

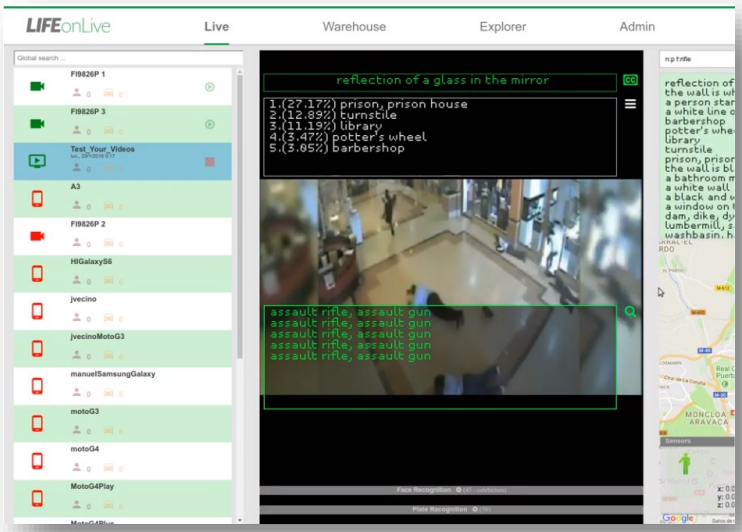


Figure 13 LifeOnLive mobile surveillance screenshot

- LifeOnLive (https://www.youtube.com/watch?v=NFadp8N8KV4) is a mobile surveillance solution that enables security agents wearing bodycams to stream their activities to a data centre automatically (see Figure 13). Besides for purely storing the videos, there are analysis stages for automatically detecting features of interest such as faces, license plates and others. Currently we're exploring

[3] OpenPose Network: https://github.com/CMU-Perceptual-Computing-Lab/openpose
[4] PyOpenPose wrapper: https://github.com/FORTH-ModelBasedTracker/PyOpenPose

the addition of pose estimation to provide more fine grained analysis of attitudes or behaviours of interest for security (e.g., raising the arms, pointing at something, aiming a gun). Competition in this field is strong as mobile surveillance is a rising market driven by an increased requirement for police and security forces' accountability. The current market leader is Axon Enterprises[5] (formerly Taser) who have merged with much of the competition in the field and offer systems that are very well featured (e.g., video processing for detection of activity, secure cloud storage and FullHD streaming. Other alternatives exist, especially systems tailored for the particularities of European markets (e.g., GDPR compliance) such as Zepcam[6] and IWitness[7]. Compared to these systems, LifeOnLive, augmented with the results of FitOptiVis, offers a much lower price point (in the range of the hundreds of euros per device instead of the thousands), done so by leveraging on COTS devices for capture as well as improved edge and cloud distribution.

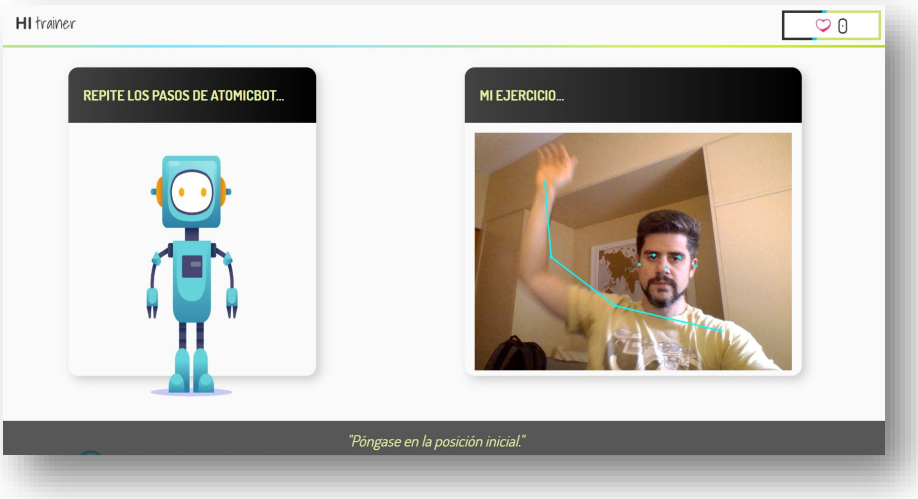


Figure 14 Hi-Trainer screenshot.

- HI Trainer (see Figure 14) is an experimental module for our e-health products including REVITA (http://revita.hi-iberia.es/en/) in which a user (usually a senior citizen or person undergoing rehabilitation or physical therapy) is commanded to follow a series of exercises and then evaluated on their performance. This is a pre-commercial for now and not integrated in commercial releases of REVITA, but technology from FitOptiVis will be used in future evolutions. The market for solutions similar to this product is still nascent in most aspects, with only very narrow niches having significant competition (e.g., sports analytics tracking systems) that have a different set of requirements such as wide angle of vision to analyse whole playing fields. For the applications that are most related to the needs in FitOpTiVis (e.g., tracking the pose of persons in close quarters such as their homes) the situation is less advanced, with many solutions just out of the research phase and not a wide adoption. Among the commercial offerings, Oxehealth[8] offers similar features of what is being researched in FitOpTiVis for Habit Tracking (e.g., cognitive decline, fall detection), although they are recently

[5] https://global.axon.com/ - Axon Enterprises website.
[6] https://zepcam.com - Zepcam website.
[7] https://www.iwitness.nl/index.php - IWitness system.
[8] https://www.oxehealth.com/ - Oxehealth patient monitoring

spun off Oxford University research groups and have a limited number of real-world deployments. However, the efforts in pose estimation in FitOpTiVis can potentially provide more fine tuned results, beyond large full body actions such as falls to more nuanced such as the usage of objects in the home.

4.8.4 Extension within FitOptiVis

TRL@M0:
TRL4/5
Expected Extensions:
The particular features of the pose recognition system that we are pursuing as part of FitOptiVis are the following:

- Adaptivity for changing context of operation (e.g., scaling down performance in low energy scenarios) and distribution of video analytics tasks on various devices.
- Execution in closer to market, more performance-constrained devices (ARM devices, in-camera hardware).

TRL@M36:
TRL6/7

4.8.5 Prospective adoption

This component is being developed as part of the video processing technology in UC3: Habit Tracking (see architecture Y2 in Figure 15). The target usage is the monitoring of persons in their own homes to detect situations of interest (falls, abnormal patterns of activity) that can be used by family members to check on their relatives' wellbeing and even by doctors remotely as clues of mild cognitive impairments being developed. The adaptation mechanism is presented in D4.2 and D4.3 and enables the system to optimize the use of energy.

4.8.6 Intellectual Property information

The component is being produced as proprietary software by HI Iberia, with the aim to move it closer to market during the medium term following the project. It has been also further integrated with the rest of the remote health components produced in HI Iberia as part of the REVITA platform (see section 4.8.5).

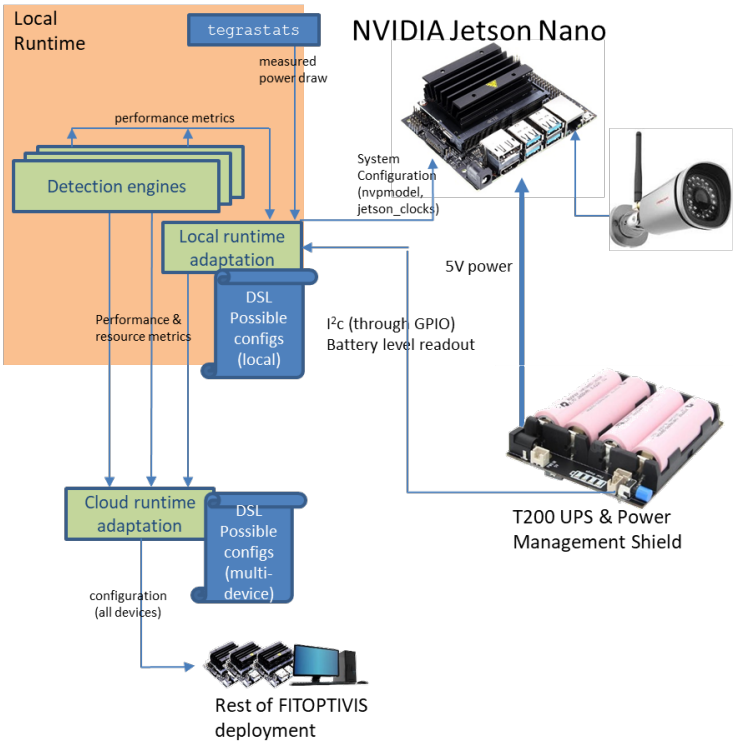


Figure 15 Architecture of distributed pose estimation processing in UC3

4.8.7 Related Documents

[KERAS] Keras Documentation. Online: https://keras.io/

[CNTK] Microsoft/CNTK. Online: https://github.com/Microsoft/CNTK

[TENSOR] TensorFlow. Online https://www.tensorflow.org/xla

[GBC2016] I. Goodfellow, Y. Bengio, A. Courville, "Sequence Modeling: Recurrent and Recursive Nets", Deep Learning Book, MIT press, 2016.

[HIK2012] T. Hassner, Y. Itcher, O. Kliper-Gross, "Violent Flows: Real-Time Detection of Violent Crowd Behavior", IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[MOS2009] R. Mehran, A. Oyama, M. Shah, "Abnormal Crowd Behavior Detection using Social Force Model", IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[RZL+2016] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, "Squad: 100,000+ questions for machine comprehension of text", Conference on Empirical Methods in Natural Language Processing, 2016.

[CHS+2018] Z. Cao, G. Hidalgo, T. Simon, S.E. Wei, Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields", Computer Vision and Pattern Recognition, 2018.

[STA+2019] D. Smilkov, N. Thorat, Y. Assogba, A. Yuan, N. Kreeger, P. Yu, K. Zhang, S. Cai, E. Nielsen, D. Soergel, S. Bileschi, M. Terry, C. Nicholson, S.N. Gupta, S. Sirajuddin, D. Sculley, R. Monga, G. Corrado, F.B. Viégas, M. Wattenberg, "TensorFlow. js: Machine Learning for the Web and Beyond", Machine Learning, 2019.

4.9 Reconfigurable video capture and processing [UC]

This module integrates two types of components: video capture devices and video processing components. The implementations of these components is selected at runtime (dynamic component reconfiguration).

4.9.1 Short description

This module integrates video capture with some video processing algorithms. It supports several configurations and accelerator mapping. At runtime, some part of the module could be re-allocated to a different computational resource. For example, the video capture module could be allocated in HW and supports board camera inputs or in SW and supports USB camera inputs. The component also includes a library of video processing algorithms that provides several functions such as satellite identification. The module integrate four types of components:

- Video capture components. These components support image and depth sensors (RGBD cameras). In the final version, three types of components will be supported: HW [LEOPARD], depth [INTEL] and USB cameras.
- Object detection: These components will detect specific elements (e.g. satellites) using deep learning techniques such as Convolutional Neural Networks. These components will provide the detected object's ID and its location within the image. If required, these modules will also provide the distance of the object from the camera.
- Image compression. In order to encode the image, the module will use space image compression standards (e.g. CCSDS 122.0 [CCSDS]).
- Standard video processing techniques. Finally, some components will implement commonly used image processing algorithms such as image scale, colour conversion, etc.

Type:
SW components that can integrate HW accelerators.
Target:
The SW part is implemented in a multiprocessor cluster and/or GPU. For example, the NVIDIA Jetson Nano platform [NVIDIA] is a typical low-performance target platform for the SW part. The HW accelerators are implemented in a Zynq UltraScale+ platform [XILINXa].
Input:
The video capture components have not dataflow inputs. Other components require an input video sequence.
Output:
Video image and results of the video processing pipeline (e.g. recognized object) are generated.
Available model:
All components are described in an extension of the WP2 SDL (SDSL – Service oriented Design Specific Language). The components are also modelled on UML/MARTE. These models allow simulating and analysing system performances. The module includes a portable C++ implementation of the components. These implementations can be deployed in different computation platforms in order to provide edge computations.
Composability:
The components can be involved in larger components.

4.9.1.1 Updates during the second year of the project

During the second year all the system components (described in Section 4.9.3) have been detailed and developed. The first year module list has been improved and extended with new functionalities and components. Additionally, the RIE (Reconfigurable Implementation of Embedded systems) library has been used to implement the components. Concerning the main modules, the object recognition module that will be used to identify satellites, has been deployed and tested on different edge platforms such as Jetson Nano and Jetson TX2. Different CNN architectures have been also evaluated in order to compare speed and accuracy. The camera component has been improved to support new implementations such as RGBD cameras. We are also evaluating the redesign of the hardware implementation. The original version is based on Xilinx Revision and we have been evaluating to use the new Xilinx framework (Vitis).

4.9.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Qualities:
The qualities depend on the video processing pipeline module. All of them have image resolution and frame rate qualities.
Set Points:
Every point defines, at least, video source input, video processing pipeline and module implementation (HW/SW mapping).

4.9.1.3 Design-time support

DT tools – Xilinx SDSoC and reVISION library [XILINXb]. OpenMP with UC extensions of heterogeneous platform programming. UC library (RIE) for reconfigurable component implementation. UC UML/MARTE modelling framework (S3D). Tensorflow Object Detection API [TENSORa] for training object detection models. TensorRT [TENSORb] for optimizing and inferencing deep learning algorithms.

4.9.1.4 Run-time support

The components have been implemented with a C++ library that provides and manages runtime HW/SW reconfiguration. The library (RIE, Run-time reconfigurable Implementation of Embedded systems) is a WP3 UC development that is evaluated with these components.

4.9.2 Related state of the art

The module uses reconfiguration techniques to modify the component implementation at runtime. This process must preserve the system functionally while produce low-impact on system operation. However, the impact that dynamic reconfiguration produces is difficult to evaluate at design time [MBG+2011]. Additionally, video pipeline also requires that the reconfiguration process has low impact on real-time performances [OVE2016]. A possible solution is to use runtime information (e.g. traces) to reduce the reconfiguration impact [SMJ2018]. For heterogeneous platforms, dynamic reconfiguration techniques improve static configurations with an important speed up (until x7.7 in [FPZ+2019]). The approach of this module goes in this direction: implementing dynamic reconfiguration techniques in which components are allocated to heterogeneous hardware resources in order to improve system performances. The innovation in WP5 is to develop an implementation technique that facilitates dynamic reconfiguration and support WP2 DSL specifications.

Concerning the module components, the image capture element integrates several implementations in the same model. The HW camera implementation is based on the Xilinx reVISION (Responsible and Reconfigurable Vision Systems) code [REV_XIL], although we are evaluating the Vitis-based approach. Other implementations (e.g. depth and RGB camera) are also based on device specific libraries. In this component, the main innovation is the integration of different types of sensors in the same component with different implementations. This component facilitates the camera device uses.
The object recognition components are based on deep learning techniques that combine several types of sensors (e.g. RGB and depth sensors). The goal is to develop deep learning techniques for autonomous vehicles that are able to identify specific objects. The component reconfiguration allows using the most efficient network for a particular system situation. The deep learning models are trained using Tensorflow Object Detection API [TENSOR] and then optimized and deployed with TensorRT [TENSORb]. One of the main problems of space object recognition is the lack of data. State of the art BUAA Satellite Image Dataset (BUAA-SID 1.0) [GZZ+2010] provides data for 3D space object research. This dataset contains images of 20 different satellites. The main innovation of this module is to explore depth-learning techniques for satellite recognition, an area in which there is a limited set of contributions.
Concerning image compression, the module must use an image space standard, therefore the innovation of the compression algorithm is limited. However, the implementation of this standard (CCSDS 122.0) on different computation resources will be one of the component innovations.

4.9.3 Commercial positioning

N/A.

4.9.4 Extension within FitOptiVis

TRL @ M0:
TRL2: Xilinx reVISION interface and OpenCV components.
Expected Extensions:

- Improve reconfiguration capability of video processing algorithms
- Integrate real-time performance monitoring
- Dynamic set point selection
- HW/SW reconfigurable image compression algorithm

TRL @ M36:
TRL5: First complete version. Runtime reconfigurable component that can be integrated in the autonomous vehicle use case.

4.9.5 Prospective adoption

Usage in FitOptiVis use cases: UC10 and UC2 (optionally).
Fields of application – Mainly satellite image processing application.

4.9.6 Intellectual Property information

Some modules could be delivered as IP components or, eventually, open source code. In particular, we are planning to deliver the synthesizable code of the image compression module as an IP or open source component.

4.9.7 Related documents

[LEOPARD]	Leopard Imaging FPGA-oriented camera. Online: https://leopardimaging.com/product-category/xilinx-camera-kit/
[INTEL]	Intel Real Sense. Online: https://www.intelrealsense.com/stereo-depth/
[NVIDIA]	Nvidia Jeson Nano. Online: https://developer.nvidia.com/embedded/jetson-nano-developer-kit
[XILINXa]	Xilinx Ultra Scale. Online: https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html
[XILINXb]	Xilinx reVISION. Online: https://www.xilinx.com/support/documentation/backgrounders/revision-overview.pdf
[MBG+2011]	X. Ma, L. Baresi, C. Ghezzi, V. Panzica La Manna, J. Lu, “Version-consistent dynamic reconfiguration of component-based distributed systems”, ACM SIGSOFT Symposium and European Conference on Foundations of Software Engineering, 2011.
[OVE2016]	R. Oliveira Vasconcelos, I. Vasconcelos, M. Endler, "Dynamic and coordinated software reconfiguration in distributed data stream systems", Journal of Internet Services and Applications, 2016.
[SMJ2018]	W. Song, X. Ma, H.A. Jacobsen, "Instance migration validity for dynamic evolution of data-aware processes", IEEE Transactions on Software Engineering, 2018.
[FPZ+2019]	J. Fumero, M. Papadimitriou, F.S. Zakkak, M. Xekalaki, J. Clarkson, C. Klotselidis, "Dynamic application reconfiguration on heterogeneous hardware", International Conference on Virtual Execution Environments, 2019.
[CCSDS]	Blue Books: Recommended Standards.Online: https://public.ccsds.org/publications/BlueBooks.aspx
[TENSORa]	TensorFlow Object Detection API. Online: https://github.com/tensorflow/models/tree/master/research/object_detection
[TENSORb]	Tensor RT User Guide. Online: https://docs.nvidia.com/deeplearning/sdk/tensorrt-archived/tensorrt_210/pdf/TensorRT-User-Guide.pdf
[GZZ+2010]	M. Gang, J. Zhiguo, L. Zhengyi, Z. Haopeng, Z. Danpei, “Full-Viewpoint 3D Space Object Recognition Based on Kernel Locality Preserving Projections”, Chinese Journal of Aeronautics, 2010.

5 Communication-oriented components

This section describes communication-oriented components developed in FitOptiVis related to the use cases of the project

5.1 Time sensitive networking [7SOLS]

Time Sensitive Networking (TSN) guarantees bounded low latency, low packet delay variation and low packet loss in the context of mixed-criticality applications. TSN, a set of updates of IEEE Ethernet standards, brings backbone for real-time distributed control systems (Surveillance of smart-grid critical infrastructure UC) or edge-cloud distributed

processing and performance monitoring (Habit tracking UC), among other possible applications.

5.1.1 Short description

The TSN component is a mixed software and IP-core-based library to enable Time Sensitive Networking switching and forwarding capabilities in Xilinx FPGA SoCs. The TSN component is composed by two interdependent functional sub-modules:

- the Hybrid Communication Manager subsystem
- the Network Timing subsystem.

The Hybrid Communication Manager (HCM) subsystem provides differentiated or hybrid RT-QoS (bounded latency, guaranteed bandwidth) to user-defined traffics, given by layer 2 (Ethernet), 3 (IP) or 4 (TCP/UDP) protocol header patterns. To this end, mechanisms defined on IEEE 802.3 and IEEE 802.1Q standards are adopted. Regular network sockets on software or AXI4-Stream on programmable logic are used to transfer mixed time-critical traffics to Network User layer applications. Traffic differentiation and prioritization is addressed through VLAN tagging (ID and Prio fields), conforming TSN streams. Traffics are dispatched according to the TSN stream priority, minimizing interferences from lower priority traffics. To this end, a Time-Aware traffic Shaper (TAS) of the output bandwidth is adopted, enabling the desired RT-QoS. This mechanism consists on a strict time-driven cyclic schedule. In order to assure coordination between TAS along the network, this component requires knowledge of the network timing, provided by the Network Timing Component. The TAS is complemented by the frame pre-emption mechanism (IEEE 802.1Qbu) and the interspersing of express traffic, to enhance the isolation between time-critical traffics and less priority ones.
The Network Timing Subsystem (NTS) offers network time synchronization to both the IP-cores present on the programmable logic and the processing system internal clock. To this end, IEEE 802.1AS standard-defined mechanisms, namely generalized Precision Time Protocol (gPTP), are implemented. The NTS collaborate on the Best Master Clock Algorithm, and therefore capable of electing the network time reference. Furthermore, the NTS provides time-aware bridge capability, as it can handle multiple interfaces to receive redundant time synchronization information and to redistribute time synchronization to neighbour peers. Protocol mechanisms and control of the local PTP Hardware Clock (PHC) entity is addressed through the gPTP software module. Sub-microsecond synchronization accuracy is provided thanks to the hardware TimeStamping Unit (TSU) present on every gPTP-capable interface. Network time synchronization is provided by the PHC to programmable logic, whereas a second-leap interrupt line is provided to synchronize internal processing system clock. Finally, a network user API allows configuration and runtime monitoring of the NTS.

5.1.1.1 Updates during the second year of the project

During this year (M12 to M24) the following actions have been addressed regarding TSN component development and verification:

- The different elements related to the Hybrid Communication Manager (HCM) have been object of test and refinement. These actions were focused on the VLAN switching (IEEE 802.1Q) and the traffic shaping and scheduling (IEEE 802.1Qbv). The tests have verified the forwarding capabilities, the isolation between different traffic priorities, as well as determinism and bandwidth guarantee. As a result, the TSN bridge has reached the maturity required for the integration

- The Network Timing Subsystem (NTS) has also been object of testing and refinement. The servo algorithm has been improved to improve the convergence to the time reference, as well as the accuracy. The hardware timestamping mechanism has also been object of minor modifications. Moreover, the interoperability with third party equipment (i.e. National Instruments) has been addressed. After fixing minor message formatting errors, master and slave port roles have been tested successfully.
- The time-aware bridge capability has been implemented on the NTS. The slave and master role capabilities have been modified to compute the residence time, key point to perform the synchronization information forwarding. Besides, the passive role functionality has been implemented. This port role enables redundant synchronization information and enhances reliability in case of link failure.

5.1.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

- *Hybrid communication data manager*

Qualities:

The qualities that can be considered are link layer effective bandwidth, traffic differentiation and prioritization, and deterministic delivery for the most critical traffics. Traffic differentiation and prioritization are provided through VLAN encapsulation, conforming TSN streams. The deterministic delivery can be quantified through the end-to-end latency deviation for the most critical traffics, provided that both talker and listener nodes have knowledge of the network timing. Moreover, the link layer effective bandwidth, defined as the bit rate per second between user applications in absence of traffic interferences can be measured.

Set-Points:

With respect to Set-Points, TSN traffic manager IP is configurable at network deployment, attending to the application and network topology requirements. In particular:

a. To achieve traffic differentiation and prioritization through TSN streams:
   - The traffic types object of differentiated RT-QoS should be defined by a combination of layer 2, layer 3 or layer 4 protocol headers.
   - A specific 2 octet identity should be defined to be mapped on the VLAN ID fields
   - A priority should be specified for each traffic type and will be coded on the VLAN PRIO field.

b. To achieve bounded latency and guaranteed bandwidth for time-critical traffics, a proper scheduling table should be defined for the Time-Aware traffic shaper. The scheduling table is a traffic priority-based arbitration scheme of the output bandwidth. Same priority traffics compete for the same bandwidth resources and are dispatched following a FIFO scheme. The scheduling table determines which priorities are dispatched at a given time. Therefore, there is a trade-off between the guaranteed bandwidth for highest priorities and available bandwidth for the lowest traffics. In other words, deterministic deliveries are performed at the cost of available bandwidth for the rest of traffics.

Time Aware traffic Shapers along the network make use of the time synchronization to cooperate between each other and yield end-to-end RT-QoS.

- *Network timing subsystem*

Qualities:

The qualities of the NTS are the synchronization accuracy, besides capabilities such as the synchronization of the processing system and the Best Master Clock Algorithm (BMCA), including the time-aware bridge capability.
The synchronization accuracy quality can be quantified through the comparison between Pulse-Per-Second signals generated from the PHC at different stations.
The Best Master Clock Algorithm can be evaluated considering each time-aware station configuration and the network topology. The BMCA should detect the time reference, determine the preferred path to the time reference, redundant paths to this time reference and the attached stations requiring retransmission of synchronization event messages. Furthermore, the synchronization of the Processing system can be verified comparing the internal clock and the current PTP Hardware Clock time (PHC) time, which can be accessed through a memory-mapped register.
Set-Points:
According to the IEEE 802.1AS and implementation-specific requirements, some of the tuneable set-points are the following:

a) Protocol messages frequency.
b) GrandMaster eligibility administrative parameters.
c) List of active interfaces.

5.1.1.3 Design-time support

We used DT tools - Xilinx Vivado 2017.3 design suite for IP-core development, and GCC 7.3.0 for IP-core API programming.

5.1.1.4 Run-time support

Run-time adaptivity may be provided at higher layers. TSN provides services (time synchronization, differentiated side-band communication) besides run-time monitoring to assist on run-time configuration.

- *Hybrid communication data manager*

Monitoring:
The delivery of time-critical traffic frames can be precisely monitored by means of hardware timestamping to observe latency and determinism figures.
Programming API
Traffic types object of differentiation and prioritization, defined as a combination of protocol headers. VLAN encapsulation to time-driven cyclic schedule for output bandwidth arbitration.

- *Network timing subsystem*

Monitoring:
This subcomponent provides run-time monitoring of local time synchronization quality (time synchronization jitter) and the protocol functionality: current grandMaster or network time reference, synchronization path to the grandMaster, remote network capability to process the synchronization protocol. These metrics are internally used by the subcomponent to perform switchover of the grandMaster and/or synchronization path.
These metrics, as well as the current synchronization deviation can be retrieved through the user API and sent through a TSN stream to a central server
Programming API.
Time-aware station eligibility as grandMaster (standard-defined attributes). gPTP active interfaces, protocol message periodicity.

5.1.2 Related state of the art

Time Sensitive Networking is an update of IEEE Ethernet Standards [IEEEa], [IEEEb], [IEEEc], [IEEEd] for the implementation of synchronized, distributed control and real-time systems. Currently, TSN is applied on professional audio/video, automotive and industrial distributed systems.
Conventional Ethernet networks (i.e. 1000 Base-X, IEEE 802.3-1999) cannot be considered deterministic because of potential collisions and absence of traffic scheduling. On one hand, 1000 Base-X standard allows the physical medium to be shared by multiple stations (i.e. hub or bus network topologies). The Medium Access Control implements the CSMA/CD mechanism to avoid packet collisions by deferring transfers until the physical media is available. On the other hand, packets are dispatched following a FIFO scheme. To sum up, predictability or determinism for time-critical traffics cannot be guaranteed.
Existing deterministic Ethernet standards alter the 1000 Base-X physical and datalink layers or restrict the network architecture. Alternatives to TSN, such as EtherNet/IP or Ethernet PowerLink consist on application layers over 1000 Base-X and do not require specific hardware. However, network topology is restricted. Other alternatives, such as EtherCAT or PROFInet require specific hardware, thus restricting compatible devices. However, all of them provide optional synchronization support by means of specific IEEE 1588 profiles.
All these alternatives are vendor-locking and are based on protocol enhancements that are incompatible with each other. Conversely, TSN is an IEEE Ethernet standard. Therefore, TSN stations are fully interoperable with legacy Ethernet stations (i.e. 1000-Base-X) and some of the afore mentioned protocols such as Ethernet/IP or PowerLink can take advantage of the RT-QoS provided by TSN. Furthermore, TSN is ready to support Industry 4.0 requirements, such as increased bandwidth and higher information transparency between lower and user application layers.

5.1.3 Commercial positioning

Although TSN is a set of standards currently under development, it has been successfully applied on automotive ADAS, communication networks and industrial automation sectors. Some industrial controller and automotive manufacturers, such as Renesas, Belden, TTech and Analog Devices, among others, have introduced TSN support on their devices.
The TSN solution currently under development on Fitoptivis framework supports ultimate standard updates, such as frame pre-emption capability besides the expertise of Seven Solutions on time synchronization. Our added value, compared with existing solutions is the outstanding level of accuracy, that in our first results is showed to be in the 50 ns range, while current commercial solutions stand for 100 microsecond – millisecond range.

5.1.4 Extension within FitOptiVis

TRL@M0:
TRL3: At this moment the VLAN module and TAS module are being implemented. VLAN tagging/untagging and prioritization attending to the protocol fields specified by the user. Time driven cyclic scheduling of the different traffic priorities per output channel. Seamless redundancy of user specified traffics. Frame pre-emption of user-defined priorities.
TRL@M36:

TRL5: VLAN and TAS module fully implemented. Besides, seamless redundancy and frame pre-emption functionalities will be added to the basic VLAN and TAS functionalities

5.1.5 Prospective adoption

Use Case 5. Habit-tracking
Use Case 9. Surveillance of smart-grid critical infrastructure.
Other fields of application: Industrial IoT, automotive, avionics and any applications requiring networking able to manage low priority and deterministic traffics simultaneously, and/or time synchronization.

5.1.6 Intellectual Property information

This TSN solution addresses deterministic and mixed-critical communication from other nodes and end user applications hosted on the same Xilinx-7000 platform (both in software and hardware). A user manual and End User API's are available to end users. However, it is not planned to release the designs as open source.

5.1.7 Related documents

[IEEEa] IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks," in IEEE Std 802.1AS-2011.

[IEEEb] IEEE Standard for Local and metropolitan area networks-- Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks-- Corrigendum 1: Technical and Editorial Corrections," in IEEE Std 802.1AS-2011/Cor 1-2013 (Corrigendum to IEEE Std 802.1AS-2011).

[IEEEc] IEEE Standard for Local and Metropolitan Area Network--Bridges and Bridged Networks", IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014).

[IEEEd] IEEE Standard for Local and Metropolitan Area Networks--Bridges and Bridged Networks -- Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements," in IEEE Std 802.1Qcc-2018 (Amendment to IEEE Std 802.1Q-2018 as amended by IEEE Std 802.1Qcp-2018).

5.2 HSR implementation in RTU [SCHN]

Highly-available Seamless Redundancy is a redundancy communication protocol that ensure high availability and reduce the network recovery and transmission time in simple fails to "Zero". The communication frames are cloned and sent through two different port to a network with ring topology. Including this technology in Remote Terminal Units (RTU) help to meet the demands of communications time response in control systems used in critical environment.

5.2.1 Short description

This is supposed to be the communication component for the Surveillance of Smart grid critical infrastructure use case, being responsible for ensuring the exchange of

information between different RTUs present in the smart grid network that allow HSR in the communications with other elements of the computing platform or implemented systems. The aim of this component is to allow communications in a more reliable way by using ring redundancy and enabling zero-time recovery against single point of failures. It involves both hardware and software elements.
Regarding hardware, it's necessary to select the physical interfaces through which the communication will take place (RJ45 Ethernet interface, Fiber optic, etc.). The RTU component includes the AM33xx Sitara microprocessors [TIa] from the ARM Cortex-A8 family, and with their PRUs components will be possible to develop the HSR functionalities thanks to their specific features (see Figure 16).

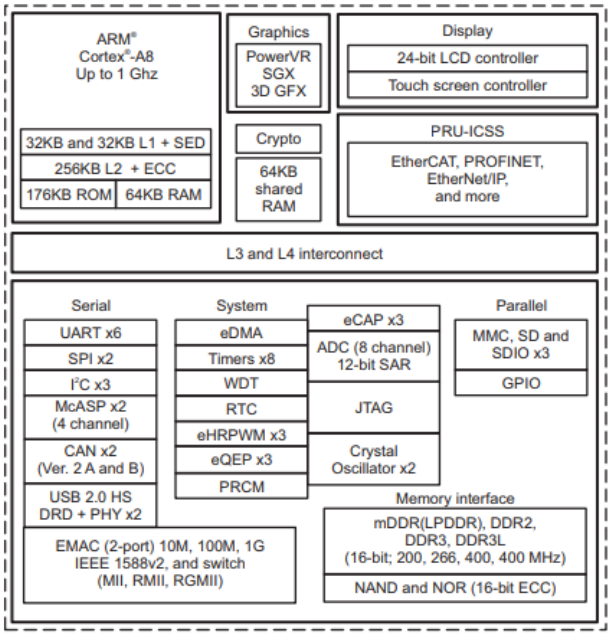


Figure 16 AM335x Functional Block Diagram.

As per the software, it's necessary to select which protocols are used to achieve the information exchange over the physical interfaces already defined.
The system is designed to transmit all information through two communication paths in a ring topology, and to discard duplicated frames. Should there be any problem in one of the communication paths, only one frame would be delivered, and the system would be able to work without any problem and no delays or packets loss due to single point failures. In contrast, other ring redundancy methods would block one path, and would only use the primary one, dropping some packets until recovered from the fault.

Input (inputs to be received by the HSR component)
Incoming communication with smart grid frames as per HSR communication specification conforming a ring.
Output (outputs to be produced by the HSR component)
Outgoing communication with smart grid frames as per HSR communication specification conforming a ring.
Communication (signals or data packets to be exchanged with the HSR component, i.e., inputs and outputs)
Status/commands of binary signals (ON/OFF). Value of analogue registers or set-points. Any other calculated value through PLC configuration (transformer overload, number of switching operations, etc.)

5.2.1.1 Updates during the second year of the project

Based on HSR hardware need analysis, different modifications are being made and tested in the control board of an existing RTU. This RTU contains AM33xx Sitara microprocessors. So, the following changes are needed to aim the HSR requirements:

- Ethernet 1 and Ethernet 2 RGMII interfaces replacement for PRU_MII interfaces.
- Ethernet 1 and Ethernet 2 Gbit 88E1512 physical layer replacement for Texas Instrument.
- The GPMC microprocessor responsible of memory peripherals has been modified.

A validation of this first version has been carried out.
Software modifications have been made in the OS to adjust Linux to the new feature. And a development of HSR functionality is underway.

5.2.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Qualities:

- Quality of Service, QoS:
  - Redundancy communication.
  - Zero time recovery for simple fails in a link or switch.

Set-Points:

- Physical interface in the ring configuration
- Ethernet RJ45 port
- Optical fiber

Available model:

- HSR standard IEC 62439-3 [IECa]. High-availability Seamless Redundancy is based on the Ethernet or Fiber Optic technology. This network protocol provides seamless recovery against single failure of a bridge in the network.

5.2.1.3 Design-time support

- Specific internal development tools for Remote Terminal Units.
- Arago Texas Instrument and Yocto Project to generate Linux Kernel [TIb].
- Concept HDL and Pin Muxing Texas Instrument for schematic design [TIc].
- Allegro for PCB routing [CADENCE].

5.2.1.4 Run-time support

Synchronization provided by the TSN network.

5.2.2 Related state of the art

For critical infrastructure and high-performance networks, the interruption in the communications, even for minimal times, are unacceptable. Ethernet is a technology with broadcast functionality but it could happen that there was only one active link between the source and the destination. So, to have redundancy we would need additional links to compensate and a redundancy control protocol to administrate the links.
A great number of redundancy control protocol exists. A few of them are defined as standards, such as:

- Media Redundancy Protocol MRP (IEC 62439-2) [IECb]
- Parallel Redundancy Protocol PRP (IEC 62439-3) [IECa]

- High-availability Seamless Redundancy HSR (IEC 62439-3) [IECa]
- Cross-network Redundancy Protocol CRP (IEC 62439-4) [IECc]
- Beacon Redundancy Protocol BRP (IEC 62439-5) [IECd]
- Distributed Redundancy Protocol DRP (IEC 62439-6) [IECe]
- Ring-based Redundancy Protocol RRP (IEC 62439-7) [IECf]

The selection of redundancy standards for different smart grid applications would depend on the requirements of those applications.

5.2.3 Commercial positioning

Direct competitors such as Siemens and ABB do not yet have a product offer clearly defined of RTU devices that include HSR technology. Instead they have included this technology on products like industrial switches:

- AFS660 Switch, ABB. [ABB]
- SCALANCE X-200RNA. [SIEMENS]

5.2.4 Extension within FitOptiVis

TRL at M0:
TRL2: There is a technology concept formulation about how HSR should be implemented in a RTU and how it can help to improve the reliability of control systems.
TRL at M36:
TRL6: At the end of the project is expected to have implemented HSR in a RTU, integrated in the surveillance of smart-grid critical infrastructure use case

5.2.5 Prospective adoption

The HSR technology will be used in surveillance of smart-grid critical infrastructure use case. The HSR brings reality to the use case making it closer to the real environments used in the smart-grid.
Furthermore, the most recent equipment that are being installed in some parts of the smart-grid try to reinforce availability of the communications due to the criticality of the data that are exchanged, using redundancy protocols and structures on different occasions. With this concept, HSR gives the use case a high availability for the critical data exchanged between the Surveillance system and the RTUs that manage a part of the smart-grid. In addition, a TSN network will be included between the RTUs and the Surveillance system to be able to assign priorities on the network traffic depending on its criticality.

5.2.6 Intellectual Property information

N/A

5.2.7 Related documents

[TIa] Texas Instrument, AM335x Sitara TM Processors Datasheet. Online: http://www.ti.com/lit/ds/symlink/am3357.pdf

[IECa] IEC, IEC 62439-3:2016. Industrial communication networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR), 2016.

[TIb] Texas Instruments, “What is Arago? Learning more about TI support of the Yocto Project,” Online: https://training.ti.com/what-arago-learning-more-about-ti-support-yocto-project-0

[TIc] Texas Instruments, “Pin mux tool,” Online: http://www.ti.com/tool/PINMUXTOOL#descriptionArea

[CADENCE] Cadence, Allegro PCB Designer, Online: https://www.cadence.com/en_US/home/tools/pcb-design-and-analysis/pcb-layout/allegro-pcb-designer.html

[IECb] IEC, IEC 62439-2:2016. Industrial communication networks - High availability automation networks - Part 2: Media Redundancy Protocol (MRP), 2016.

[IECc] IEC, IEC 62439-4:2010. Industrial communication networks - High availability automation networks - Part 4: Cross-network Redundancy Protocol (CRP), 2010.

[IECd] IEC, IEC 62439-5:2016 Withdrawn. Industrial communication networks - High availability automation networks - Part 5: Beacon Redundancy Protocol (BRP), 2016.

[IECe] IEC, IEC 62439-6:2010. Industrial communication networks - High availability automation networks - Part 6: Distributed Redundancy Protocol (DRP), 2010.

[IECf] IEC, IEC 62439-7:2011. Industrial communication networks - High availability automation networks - Part 7: Ring-based Redundancy Protocol (RRP), 2011.

[ABB] ABB, AFS660 Switch, Online: https://new.abb.com/communication-networks/communication-networks/products/ethernet-communications-af/afs660-switch

[SIEMENS] SIEMENS, “Seamless communication thanks to the PRP and HSR protocols,” Online: https://new.siemens.com/global/en/products/automation/industrial-communication/seamless-redundancy.html

5.3 Fletcher big data communication interface [TUD]

Fletcher is an FPGA acceleration framework targeting data-intensive applications.

5.3.1 Short description

FPGA accelerators are becoming commonplace accelerator solutions. Interfacing efficiently with FPGA accelerators, however, still remains a challenge. This is especially true for applications written in high-level languages that want to make use of FPGA accelerators. Such applications are often developed on top of language-specific interpreters, virtual machines, or compilers. These language specific tools are often designed for ease-of-use/abstraction, control-heavy or computational workloads, but not for data-intensive workloads and application interoperability with accelerators. Therefore, the data containers are often sub-optimal with respect to integration with FPGA accelerators, and require a significant serialization step in preparing the data to be workable by FPGA. The Apache Arrow project defines a smarter, data-centric container format for 11 different languages. For this format, Fletcher can generate highly

efficient hardware structures that interface with the Arrow data containers [PSB+2019]. This allows FPGA accelerators to efficiently integrate with high-level languages, unlocking end-to-end accelerator-application interface throughput limited by system bandwidth only.

Target:
FPGA
Input:
Typically big data collections from storage, memory or network
Output:
Typically reduced or filtered data

5.3.1.1 Updates during the second year of the project

During the second year of the project, TUD continued to develop the Fletcher [FLETCHER] framework to make it industrially ready for deployment in the field. A number of updates have been included in the framework to reduce the complexity of the code and to increase the efficiency of the produced hardware designs. As a result, it was possible to use Fletcher to create demonstrators to show off its capabilities. For example, Fletcher was used to create a demo that parses the full text of Wikipedia in compressed format, streams it through to an FPGA accelerator, decompresses it and performs regular expression matching to filter specific Wikipedia pages. Thanks to Fletcher, the integration of such a complex hardware system took only 1 week of time. The demo was shown in the EFECS technology event in Helsinki and in the DATE2020 conference.

5.3.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Design time:
Fletcher automatically creates HW FPGA interfaces that allow high throughput streaming of data types defined in SW. This means that Fletcher should be able to generate at design-time any streaming data types required by the application. This includes the following qualities

- Numeric data (integers, floats, etc.)
- Alphanumeric data (characters, strings, etc.)
- Lists of data primitives (lists of integers, strings, etc.)
- Complex data structures (lists of lists, etc.)

At the same time, Fletcher generates interfaces that ensure a tradeoff between HW utilization versus data throughput by sizing data buffers and bus widths such that it prevents data stalls as data is streamed through HW. This means that there are a number of qualities that can be used as parameters for interfaces created by Fletcher. These parameters, and their specific pre-defined set-points are as follows

- Amount of HW resources available on FPGA (specifically in terms of BRAMs)
- Data throughput requirements of the application

In order to allow for a tradeoff between HW utilization and throughput, Fletcher uses the following interface parameters

- Buffer sizes in Kbytes
- Bit width: 2, 4, 8, 16, …, 2^n bits

5.3.1.3 Design-time support

Vendor agnostic component library in VHDL

- Structure generation tool (Fletchgen) - generates vendor agnostic components.
- Fletcher designs

- Simulation verified in
  - GHDL (open source, free)
  - Xilinx XSIM
  - Mentor Graphics QuestaSim
- Synthesis verified in
  - Xilinx Vivado
  - Intel Quartus

5.3.1.4 Run-time support

Programmability and Programming API:

- Software:
  - C++ run-time library
  - Python run-time library
  - Easily extendable to 11 other languages through Apache Arrow.
- Hardware:
  - Amazon EC2 F1 platform
  - OpenPOWER CAPI SNAP FPGA accelerator cards
  - Xilinx Alveo platform

5.3.2 Related state of the art

There are no systems at the moment that allow automatic generation of data communication interfaces on FPGA hardware.

5.3.3 Commercial positioning

The Fletcher data communication interface generation system represents an important component for big data applications that can benefit from integrating FPGA hardware accelerators to improve the performance of their kernels. Other data communication interfaces available are today, the most important of which are the Java Native Interface and ByteBuffers [PHA2017] do not interface big data applications efficiently with FPGA accelerators. This is especially true for applications written in high-level languages such as Java and Python that want to make use of FPGA accelerators. In addition, hardware development tools such as Vivado HLS are too generic to be effective in designing HW interfaces for big data applications. Therefore, there are a number of possible commercial opportunities where Fletcher can capture a commercial market share. One commercial opportunity is to use Fletcher as a technology component to integrate into existing hardware accelerator solutions. This allows Fletcher to be commercialized as a service solution for other solutions in the field. A second commercial opportunity is to provide dedicated big data FPGA accelerated solutions for various computationally intensive kernels.
Currently, Fletcher is a publicly available tool chain on GitHub that other commercial organizations can use. We provide a service-based business model to support the needs of commercial organizations to deploy Fletcher in their solutions.

5.3.4 Extension within FitOptiVis

TRL@M0:
The TRL level at the beginning of the project is at technology development stage (TRL4).
TRL@M36:

During the project, we will extend Fletcher by creating a demonstrator for its big data communication capabilities (TRL6). Then this demonstrator will be used to create systems to use Fletcher (TRL7). By the end of the project (M36), these systems will be tested and made operational in practice (TRL9).

5.3.5 Prospective adoption

The Fletcher big data communication interface will be discussed with various FitOptiVis project partners in order to identify its potential for accelerating the use cases developed in the project. At the moment, TUD is working closely with Philips to use Fletcher for their FPGA-based medical imaging data processing solution.

5.3.6 Intellectual Property information

The Fletcher framework is publicly available, is free to use, and is fully open source and accessible on GitHub. The project can be accessed at https://github.com/abs-tudelft/fletcher. The project is getting much attention by the users, receiving 111 starts, was forked 17 times, and on the watchlist of 16 partners. It was even mentioned in an article on the popular technology magazine ZDNet (https://www.zdnet.com/article/apache-arrow-the-little-data-accelerator-that-could/).

5.3.7 Related documents

[FLETCHER] Fletcher repository. Online: https://github.com/abs-tudelft/fletcher

[PSB+2019] J. Peltenburg, J. van Straten, M. Brobbel, H.P. Hofstee, Z. Al-Ars, “Supporting Columnar In-memory Formats on FPGA: The Hardware Design of Fletcher for Apache Arrow”, International Symposium on Applied Reconfigurable Computing, 2019.

[PHA2017] J. Peltenburg, A. Hesam, Z. Al-Ars, “Pushing Big Data into Accelerators: Can the JVM Saturate Our Hardware?”, International Conference on High Performance Computing, 2017.

5.4 Multi-sensor gateway [ABI]

The multi-sensor gateway, through an embedded Linux distribution will ensure a secure, safe and cooperative embedded system integrated in a system-of-systems environment.

5.4.1 Short description

The multi-sensor gateway will allow the runtime configurable multi-context support by modifying data priority transmission upon the scenario requirements. The gateway will be the orchestration actor of the heterogeneous network of sensors and actuators (valves, temperature, alarms, liquid and gas sensors), drones and will be completely integrated in a distributed video processing pipeline.

Input:

The gateway will process inputs from smart cameras (a subset of Video Content Analysis and Face Recognition) and drone.

Output:

The gateway will provide output to smart cameras, PLC and other actuators and sensors.

The basic architecture of system with interfaces is depicted in Figure 17.

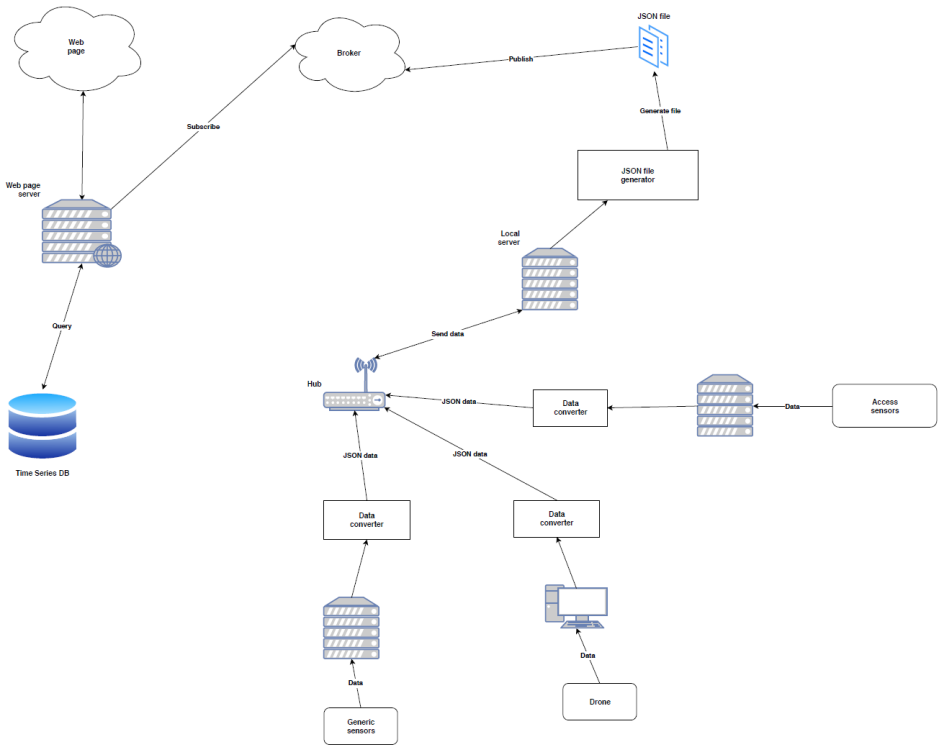


Figure 17 Multi-Sensor gateway architecture.

In order to monitor sensor data and set up the configuration of the communication system, a first release of a dedicated dashboard has been developed.

5.4.1.1 Updates during the second year of the project

During the second year of the project different activities have been performed:

1. Integration of the 'image sensors' (cameras) from different sources. This integration task has completed the acquisition from the different sensors.
2. Development of the module for the configuration of the gateway.
3. Implementation of the different network stacks.
4. A first release of the dashboard in order to monitor the sensor data and configure the communication system.

5.4.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

There are no specific predefined set-points for the gateway.

5.4.1.3 Design-time support

The multisensory gateway is not meant to be used to handle components design, analysis, verification and deployments.

5.4.1.4 Run-time support

At run time, the multi-sensor gateway will eventually orchestrate the re-configuration of the network. Upon reception of a specific trigger or event, the gateway will send to each

sensor a sequence of commands to adjust the system configuration for the specific case (for example sampling frequencies of data image, switch on a different set of cameras etc). It is important to note that the gateway is not meant to manage the video streams, but data already processed by something between the camera and the gateway, for example: detected unidentified person, detected employee, detected suspicious activities.
The dashboard is meant to be the entry point in which to configure the reactions of the gateway according to the inputs, for example: a camera detects a movement and the gateway requires the camera and the other sensors to increase sampling frequency / frame rate / speed etc. and/ or launch an alarm, send email, call the police, etc.

5.4.2 Related state of the art

There is no investigation of the state-of-the-art done for this component.

5.4.3 Commercial positioning

A technological trend that turns out to be very interesting for this type of system is the emerging one in the scientific community that is trying to define and implement the so-called "Internet of Objects" (Internet Of the Things IOT). In fact, as demonstrated by R. Metcalfe at the dawn of the birth of the Internet, the functions that a system can provide are proportional to the power of the number of interconnections of the system and ultimately to its degree of interaction with other systems or, in the most modern sense and inclusive with people or environments or natural resources.
The multi-sensor gateway intends to move remaining adherent to the communication and interface standards through Open API in order to guarantee the "growth" even outside of the gateway itself.
Even if with a completely different model of business (the solution will be released with GNU licence with an extensive use of open API for special applications) in the list below some potential competitors are reported:

1. Mindtree Multiprotocol IoT Gateway [MINDTREE]
2. Zigbee Alliance Programmable and Multi-protocol IoT Gateway [ZIGBEE]
3. Swycs Multi-protocol IoT Gateway [SWYCS]
4. FogHorn Lightning Edge Intelligence [FOGHORN]
5. Bosch IoT Gateway Software [BOSCH]

Other companies offer more similar solutions for more integrated platforms, such as:

1. ADLink product line includes edge gateways based on Intel Quark/QM SoC architectures and on Vortex Edge platform and Wind River Linux; the proposed IoT solutions feature SEMA Cloud platform integration.
2. Dell developed IoT solutions ranging from edge controllers, gateways and cloud platforms. The hardware gateways are based on Intel Atom architecture and support Linux Ubuntu snappy, Wind River Linux and Windows IoT Enterprise operating systems.
3. HPE (Hewlett Packard Enterprise) produces an IoT gateway based on Intel Xeon/Core architecture; the gateway feature modular design and support Linux Ubuntu Snappy, Wind River Linux, Windows 10 IoT.
4. Eurotech provides hardware and software IoT solutions under the name of Everyware IoT, an integrated platform for the Internet of Things which provides IoT gateways, an open edge framework and a modular cloud infrastructure. Hardware gateways are Intel Atom/NXP I.MX 8M -based and include certified Industrial, Automotive, Railway products.

It is important to note that all the reported potential competitors are well developed products ready for the market, that are not suitable for special applications as the Water Supply use case using the drones, and other application where ABI is going to propose these customised solutions.
Also, most of state-of-the-art solutions are based on their own edge application which is specifically dependent to a particular hardware, thus lacking in ease of customization, applicability and in open source aspects.

5.4.4 Extension within FitOptiVis

TRL@M0:
The multi-sensor gateway will be completely developed in the FitOptiVis project, some proof of concept with the basic function has demonstrated the potential of the proposed gateway and the dashboard. At M0 of the project, the TRL level of the component is 4.
TRL@M36:
At M36, we expect to test the final prototype in a realistic environment (water supply demonstrator) at TRL 6-7.

5.4.5 Prospective adoption

The multi-sensor gateway will support the scenario needs in the Water Supply demonstrator while guaranteeing the communication aspects with a customizing of the multi-sensor gateway.

5.4.6 Intellectual Property information

The proposed component will be released as a Linux distribution; for this reason the licence will be GNU General Public License.

5.4.7 Related documents

[MINDTREE] Mindtree, Multiprotocol IoT Gateway. Online: https://www.mindtree.com/services/engineering-rd/wireless-lighting-solution/multiprotocol-iot-gateway

[ZIGBEE] ZigBee, Programmable and multi-protocol IoT Gateway. Online: https://zigbeealliance.org/zigbee_products/programmable-and-multi-protocol-iot-gateway/

[SWYCS] Swycs, “The first and only multi-protocol IoT gateway”. Online: https://www.swycs.com/files/upload/05/27959-productsheet-en-2.pdf

[FOGHORN] FogHorn, Lighting IoT Edge Computing. Online: https://www.foghorn.io/lightning-iot-edge-computing/

[BOSCH] Bosch, Bosch IoT Suite – Gateway Software. Online: https://developer.bosch-iot-suite.com/service/gateway-software/

6 Miscellaneous (Sensor/actuators/composed)

This section describes components other than HW, SW or communication.

6.1 Wireless LED SYNC light [UWB, REX]

Point localization and registration (in case of 3D vision) are crucial tasks of many CV applications. The described module represents highly robust active motion capture markers suitable even for demanding industrial applications.

6.1.1 Short description

The wirelessly synchronized LED markers belongs to the actuator class of components. It is a distributed compound HW+FW solution consisting of a set of LED markers (MCU-based devices) bound to the central FPGA/SoC/MCU/CPU based platform over wireless link. The marker's flash synchronized with camera shutter allows to take 2 images of the scene - one with all the markers off and second one with the selected marker on. Differential image of these 2 captures allows fast, exact and robust identification and localization of markers in the image.

Input:

- LED timing definition table
- Definition table for camera capture timing
- Camera strobe pin specification

Output:

- Flashing of spatially scattered LED markers according to the definition table
- Camera STROBE pin driven according to camera capture timing table and in synchronization with the LED markers flashes

6.1.1.1 Updates during the second year of the project

The component development started during Y1 from zero and continued over Y2 of the project.

The active LED markers are based on EFR32MG SoC (System on Chip) from Silicon Labs. This chip family integrates single 32-bit ARM core with single or dual RF transceiver (see Figure 18). The first transceiver is always working in the 2.4 GHz band. The second, if present, is working in sub-GHz ISM band (tuneable 433 – 975 MHz). Due to the antenna size the single-transceiver 2.4 GHz version has been selected for the active LED markers.

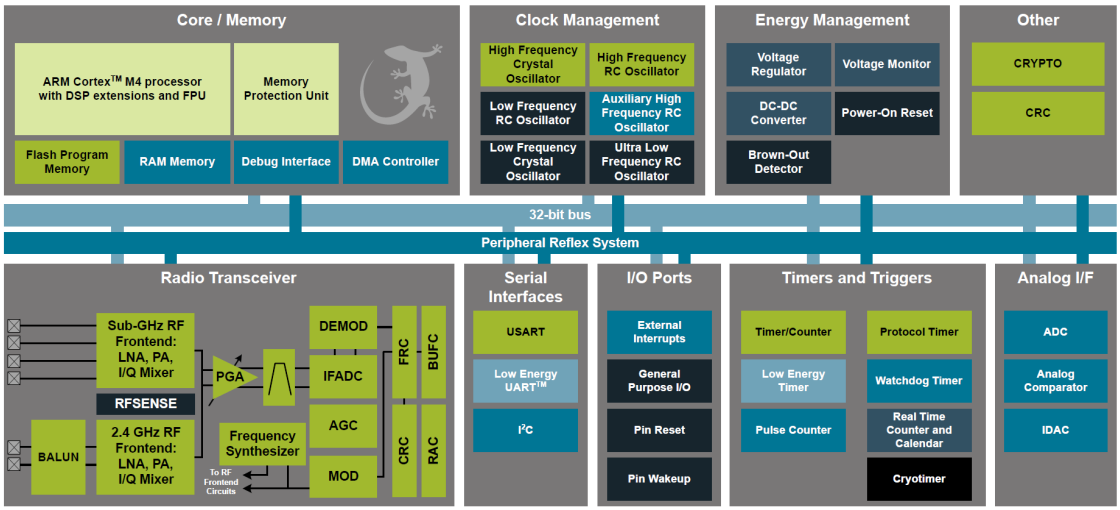


Figure 18 Structure of the Silicon Labs EFR32 chips family

In the reported period UWB has designed the marker schematics and PCB (see Figure 19). The first version of the marker housing has been engineered as well (see Figure 20).

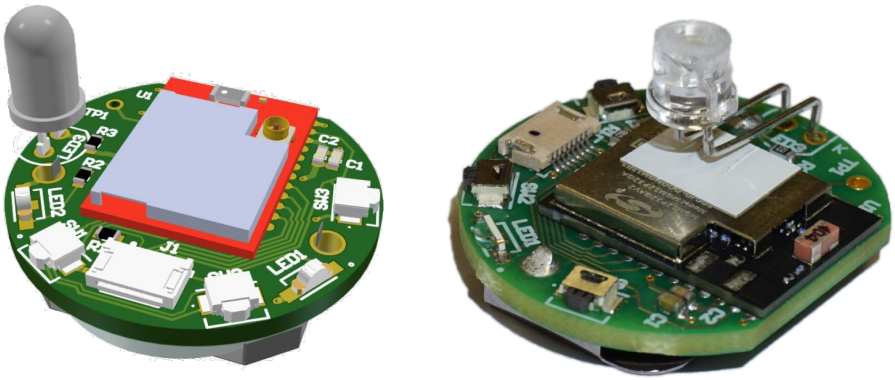


Figure 19 Active LED marker – 3D model and final PCB

Figure 20 Final LED marker in chasis

Next step was firmware development. The LED marker's firmware builds on results of the TAČR Delta 4.0 project "Reliable Time-Sensitive Networks in Distributed Cyber-

Physical Systems for Real-Time Control Industry 4.0 Applications" (No. TF04000048). The fast and synchronous communication protocol RETIS developed there allows precise time synchronization between remote devices and fast, synchronous and jitter free data exchange between the wireless nodes. Having stable time base we have implemented compound HW/SW scheduler that takes care of LED timing (flashing) according to the definition table.
The component is currently basically finished (only small refinements are required to polish the final solution) and ready to interface image processing pipeline.

6.1.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

Timing precision throughout the whole distributed system (RF LED markers + central controller) is expected to be below 10 µs. It is more than sufficient with respect to the ordinary exposition times (~5ms).
The component's working space depends on RF range of the wireless circuitry, which goes in hand with RF TX power and antenna construction. It will be dimensioned for circa 10 m range.
The component does not require any design-time parameters. As run-time parameters serve input arguments listed in the previous paragraph.

6.1.1.3 Design-time support

The wireless LED markers are based on Silicon Labs EFR32 SoC. The GNU C++ compiler together with Gecko SDK (C/C++ library for EFR32 MCUs) is used to implement the markers' functionality.
The controller-side part of the component is to be designed and implemented in Microsoft Visual Studio and compiled with the provided MS C++ compiler or with GNU C++ compiler (for ARM based HW platforms).
Additionally, the used REX block SDK provides framework and base class implementation for the component. Only functional parts of the component have to be implemented - functions Init(), Exec() and Exit() (called upon system initialization, periodically during system run and upon system shutdown).

6.1.1.4 Run-time support

In run-time the module is configured and observed via configuration window accessible through RexDraw (newly re-branded to REXYGEN Studio) - GUI based application that configures and inspects configuration of the real-time control application [REXa], [REXb].
Component configuration (inputs) can be inspected and adjusted here. Component status information is available at the same place. It includes RF packet reception details (signal strength, signal quality) and an aggregated flag indicating synchronization loss (no signal for given period).

6.1.2 Related state of the art

Currently there is no system similar to the proposed component available.
Core part of the component is wireless time synchronization. Products allowing wireless time synchronization with the stated accuracy are rare as well. The typical NTP accuracy is about 1 ms on LAN [NTP] (we speak about wired Ethernet, Wi-Fi is even worse) and thus far away from the requested precision. Better performance advertise proprietary solutions from several vendors - for example EchoRing from R3 Communications (tens of µs) [ECHORING] or V-MON from Inertia Technology (100 ns) [INERTIA]. However

these solutions are not focused on image processing. They target on process control and vibration monitoring and time synchronization is only minor part of their functionality. An alternative for the intended use of the component can be motion capture system from Vicon [VICON]. It is based on passive motion capture markers; therefore point registration has to be performed. There exist also motion capture systems that use active LED markers. Here, however, either the LED color identifies the marker (Biosense Medical) [BIOSENSE] or the LEDs are lighted on sequentially (Qualisys [QUALSYS]), no wireless synchronization with the camera shutter is performed. Also the price excludes the above listed motion capture systems from common use.

6.1.3 Commercial positioning

The state of the art solutions described in the previous paragraph can be considered as competing products as well and thus set commercial positioning of the described module. The working principles, however, are so different that it is hard to speak about typical competition. Instead, the working principle and finally the right solution should be selected according to the application requirements.
UWB plans to use the component in its object localization and tracking system. It should serve as data source for various predictors, estimators and filters as well as input for visual-feedback driven control applications.
Additionally, the component will be included as a separate block in the RexVision functional block library within the REX real-time control system and extend the functionality of the platform.

6.1.4 Extension within FitOptiVis

TRL at M0:
TRL0
TRL at M36:
UWB expects to deliver designed and fully implemented component that is tested in lab as well as in controlled industrial environment. TRL 5/6

6.1.5 Prospective adoption

The component will be used within the FitOptiVis demo pool as one of the principal components in the robot calibration use case (UC8).
Outside the project scope it can be used in any application where object tracking or localization is required. It can be used both for 2D as well as 3D localization tasks. A big advantage of this method is high robustness and possibility to exclude point registration procedures from the processing chain.

6.1.6 Intellectual Property information

Currently no IP protection is planned.

6.1.7 Related documents

[REXa] REX Controls, “Getting started with REXYGEN and Raspberry Pi“, Online: https://www.rexygen.com/doc/ENGLISH/MANUALS/RexygenGettingStarted_RasPi/RexygenGettingStarted_RasPi_ENG.html

[REXb] REX Controls, "REXYGEN Studio User Guide", Online: https://www.rexygen.com/doc/ENGLISH/MANUALS/RexygenStudio/RexygenStudio_ENG.html
[NTP] Network Time Protocol. Online: https://en.wikipedia.org/wiki/Network_Time_Protocol
[ECHORING] EchoRing. Online: http://www.echoring.com
[INERTIA] Inertia Technology. V-MON 4000. Online: http://inertia-technology.com/product/vibration-monitoring-vmon-4000/
[VICON] Vicon Visualization Tool. Online: https://www.vicon.com/
[BIOSENSE] Biosense Medical, Active Marker Motion Capture. Online: https://biosensemedical.com/active-marker-motion-capture/
[QUALSYS] Qualisys, Short range active marker. Online: https://www.qualisys.com/hardware/accessories/active-markers/short-range-active-marker/

6.2 Edge Capturer [ITI]

The Edge Capturer is a component that implements the edge computing concept in the Industrial Inspection use case. It is composed by a set of cameras connected to a set of single board computers. Each camera communicates directly with one of the boards. The objective of this component is to perform as much computation as possible reducing the workload to the following parts of the system.

6.2.1 Short description

The Edge Capturer component is an abstraction that encapsulates an arbitrary number of cameras, the low power execution boards attached to each one of those cameras, and the software running on those boards, including a work dispatcher.
Managing all these subcomponents as a single one facilitates its management in the system architecture. In this way, the Edge Capturer is treated as a sole component that is in charge of capturing, pre-processing, segmenting, and transferring the images that the other elements of a system will process. The architecture of the Edge Capturer is depicted in Figure 21.

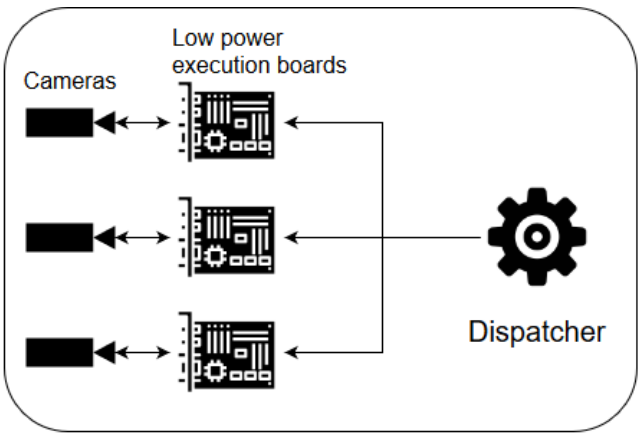


Figure 21 Edge Capturer architecture

The pipeline of the captured images is as follows. The Edge Capturer receives a request for a new capture, it then triggers the cameras so all of them take an image at the same

time (given some jitter). Each camera transfers the captured image to the attached execution board. These execution boards are the first computational layer in the system and are installed as close to the cameras as possible. Each board pre-processes the image (undistortion, debayering, and so on) and then segments it to find eg. the Region Of Interest (ROI), which is the smallest rectangular region that contains the object of interest in the image, and to separate background and object of interest. If at this point a board detects an incorrect capture, it signals the Master and this component should perform the appropriate action. The most common incorrect capture happens when the entirety of the object is not inside the field of view of the camera. Once the image is pre-processed and segmented, each execution board transfers the image to the agent who requested the capture.
To reduce costs and exploit the distributed qualities of the Edge Capturer, this component uses low cost electronic boards (e.g. Raspberry Pi or Expressobin) which are installed close to the cameras.

6.2.1.1 Updates during the second year of the project

During the second year of the project, the comparison between different development boards continued and after choosing the Nvidia Jetson Nano as the edge board to work with, ITI developed the C++ Edge Capturer software adding CUDA and CPU capabilities for image segmentation. As a result of this, the Edge Capturer was connected with GIGE cameras and was able to send the segmented ROI images via ZMQ. After that, the full execution pipeline of the prototype was implemented.
Also, in this second year, a full prototype with the 16 Jetson Nano board was ensembled. In addition, to simulate the throwing mechanism of a production inspection device, a basic hardware was designed and mounted at the top of the prototype to move different objects pulled by a dinamyxel.
Finally, ITI worked on creating a custom monitoring system based on a Telegraf agent to store data in a local database (InfluxDB). Also, the Fivis Telegraf Plugin (FTP) connector was created to send monitoring data to a remote Fivis server.

6.2.1.2 Qualities, tuneable set-points and available design-time and run-time parameters

With respect to qualities, we employ three metrics: throughput, latency, and bandwidth usage.

- Throughput is measured counting the number of captures that are processed by the component in a unit of time.
- Latency is the time that the system takes to process a new capture.
- Finally, the last metric is bandwidth usage, this is the quantity of Mbits per second that the communications from the Edge Capturer to the rest of the system use.

There are several parameters that can be adjusted for the Edge Capturer during design and run-time. In design-time, the main parameter is the number of cameras, however this is something that it is not planned to be investigated within the scope of the FitOptiVis project. Regarding runtime support, there are three parameters which can alter the beforementioned qualities:

- Image resolution will be set according to the camera specifications.
- Image encoding will vary the computation required and the image size.

Segmentation type. could be changed between a simpler or a more complete one. This will be set depending on the performance of the single board computer. Traditionally, cameras are connected to a single computing node, in this way, even when that node is not responsible for processing the images captured by the cameras, all the images

have to travel through it, increasing bandwidth usage. With the dispatcher, the images are sent to the exact node that will continue processing the images. Moreover, since the Edge Capturer has to deal with a wide variety of objects and materials, the segmentation algorithm is highly adaptable.
Further, if several segmentation attempts for a single part fail, the Edge Capturer can try different segmentation settings (which are predefined by the operator) to obtain a valid image. Finally, the output can also be adapted to the task at hand. If the colour and texture qualities of the object are irrelevant for the task, the output of the Edge Capturer can be reduced to the silhouette of the object in a binary image. This reduces even further the bandwidth usage.

6.2.1.3 Design-time support

In order to develop this component, general design and development tools are used to implement the code.

6.2.1.4 Run-time support

We plan to monitor performance at run time. The mentioned parameters (image resolution, encoding and segmentation type) will impact in the obtained monitoring data and thus enable the run-time support. Regarding this, the following metrics will be monitored:

- Throughput
- Average bandwidth usage
- Correct / incorrect capture ratio
- Average processing time per capture.

To optimize the hardware capabilities of the execution boards as much as possible, the pre-process and segmentation algorithms are implemented in C using the following APIs: OpenCV, OpenMP, ZMQ, Aravis and CUDA.
In addition to that, the system will provide adaptivity at run-time, depending on the objects characteristics, the Edge Capturer is capable of employing several predefined segmentation configurations. Moreover, the Edge Capturer can also detect incorrect captures (e.g., if and object was not positioned inside the field of view of the cameras) and report it to the Master component.

6.2.2 Related state of the art

Currently, in industrial computer vision solutions, the pre-processing and segmentation of an image is usually performed in a computational node that is not close to the camera. Moreover, this computational node is the same that performs the rest of the image processing, for example, analysis of the image contents and classification of the detected object.
As explained by Prajapati and Sanjay [PV2011], low level image processing (as segmentation) can be divided into several computation resources to reduce computation time. Moreover, as shown in [WA2005], distributed image processing can reduce energy consumption. Based on these ideas, we propose the Edge Capturer component, which aims at addressing these two challenges (computation time and energy consumption) by means of image processing distribution on low-power execution boards.
As pointed out by Shi et al. [SCZ+2016], in a system where several images taken by a number of devices have to travel to a single node, an edge-computing approach can reduce latency and bandwidth usage, while increasing throughput. Even though a wide number of applications of edge computing on image processing are focused on mobile

devices [MYZ+2017], we believe that an edge computing approach can also benefit industrial applications. Therefore, Edge Capturer takes advantage of edge computing by locating the execution boards close to the cameras. In this way, the first computation layer is as close to the sensor as possible.

6.2.3 Commercial positioning

The Edge Capturer is part of an industrial inspectionN/ device that can be used to detect defects in small pieces produced by a factory (e.g. springs). The objective of this use case is to demonstrate that it is possible to obtain a considerable increase of productivity. There are some comparable competitors being the most remarkable the following:A

- Avizo Inspect offers a 3D analysis and inspection software for dimensional metrology, based on digital object captures performed with a computed tomography acquisition device.
- VisionMaster provides a solution based on structured light scanning to automatically recognize and measure solder.
-

ATOS is an optical 3D scanner offering measurement of small to medium-sized components such as ceramic cores, cast and plastic parts.

6.2.4 Extension within FitOptiVis

Edge capturer will be completely developed and tested within the project. It is expected to take advantage of the proposed edge architecture.
These are the indicators expected to be achieved:

- Current bandwidth usage is 28.7Mbytes per capture (using 16 cameras and one central node, no dispatcher), it is expected less than 9Mbytes per capture.
- Current segmentation time of capture made up by 16 images (there are 16 images in the UC where the Edge Capturer will be integrated) is 735 milliseconds, it is expected that the Edge Capturer with each board segmenting an image in parallel will take less than 200 milliseconds.
- In qualitative terms, we expect that the Edge Capturer with its adaptive segmentation capabilities generates less incorrect captures due to segmentation problems.

In terms of TRL, these are the levels at the beginning and end of the project:
TRL at M0
TRL2: At M0 only the potential application of the Edge Capturer is validated based on the current capturing method employed in Zero Gravity (one product where Edge Capturer can be embedded). However, the specific execution boards are not decided and their software is not developed.
TRL at M36
TRL4/5: At M36, we expect a tested component that can quickly deliver pre-processed and segmented images to the worker agents, synchronize all the cameras and boards, and deal with incorrect captures.

6.2.5 Prospective adoption

In the context of FitOptiVis, the edge capturer is integrated into the 3D industrial inspection system use case.
The Edge Capturer can be applied to any system that has several capture devices (cameras) that can benefit from obtaining those images already pre-processed and

segmented. For example, robot calibration and autonomous driving vehicles would be two candidate application areas where it can fit.
Furthermore, the development is a candidate technology to be embedded into the Zero Gravity 3D industrial inspection system (https://www.zerogravity3d.com/).

6.2.6 Intellectual Property information

The Edge Capturer has two different parts to be protected. The first one is the hardware design which is already protected by patents that were granted before FitOptiVis project started. Currently, ITI is considering to extend this protection through a new patent to include the new Edge Computing oriented design.
The second component in terms of intellectual property is the software which is currently confidential and will remain in the same state.

6.2.7 Related documents

[PV2011] H.B. Prajapati, S.K. Vij, "Analytical study of parallel and distributed image processing", International Conference on Image Information Processing, 2011.

[WA2005] H. Wu, A. Abouzeid, “Energy efficient distributed image compression in resource-constrained multihop wireless networks”, Computer Communications, 2005.

[SCZ+2016] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, "Edge Computing: Vision and Challenges", IEEE Internet of Things Journal, 2016.

[MYZ+2017] Y. Mao, C. You, J. Zhang, K. Huang, K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective", IEEE Communications Surveys & Tutorials, 2017.

7 Analysis and conclusions

This report presented the components developed by the FitOptiVis partners in WP5. All 22 partners of WP5 contributed a total of 26 of their respective component developments. The components were classified into 4 different categories as summarized in the list below.

Sec 3. Processing and acceleration components:11
3.1. TUT: Customized parallel soft-cores
3.2. UTIA: Optimized HW/SW cores
3.3. UniCA: NEURAghe
3.4. UnivAQ: AIPHS
3.5. TASE: Image collection interface
3.6. TASE: Space image processing chain
3.7. CAMEA: Licence plate detector
3.8. UNISS, UNICA: MDC-generated coarse grained reconfigurable HW accelerators
3.9. UTU: Low energy dynamic voltage and frequency scaling
3.10. PHL: Multistream video image scaler and compositor
3.11. PHL: Efficient magnetic field generators

Sec 4. Software components: 9
4.1. UGR: Behavior classifier
4.2. UGR: Person tracking software
4.3. HURJA: Salmi AR Platform
4.4. UnivAQ: Distributed Video Coding Solutions
4.5. UWB, REX: Point localization from stereovision
4.6. Aitek: Video Content Analysis (VCA) platform
4.7. UNISS: Face and behavioural analysis (FBA) platform
4.8. HIB: Pose Estimation Engine
4.9. UC: Reconfigurable video capture and processing

Sec 5. Communication-oriented components: 4
5.1. 7SOLS: Time Sensitive Networking
5.2. SCHN: Communication Element for Smart Grid Surveillance
5.3. TUD: Fletcher big data communication interface
5.4. ABI: Multi-sensor gateway

Sec 6. Miscellaneous (Sensor/actuators/composed-composed): 2
6.1. UWB, REX: Wireless LED SYNC light
6.2. ITI: Edge Capturer

This deliverable shows that the components developed in FitOptiVis cover all the use cases targeted by the project:
UC1 - Water supply

- UC1 – Water supply
- UC2 – 3D inspection
- UC3 – Virtual reality
- UC4 – Habit tracking
- UC5 – Road traffic surveillance
- UC6 – Multi-source streaming composition
- UC7 – Sustainable MRI

- UC8 – Robot calibration
- UC9 – Surveillance of smart-grid critical infrastructures
- UC10 – Autonomous exploration

The development of the considered components already overlaps and interacts with other technical WP activities, such as WP2 (for the modelling part), WP3 (leveraging on tools and methodologies) and WP4 (dealing with run-time features). WP6 activities are instead interlaced with WP5 ones considering the last development steps of components, that is integrations in PoCs and demonstrators, already started for some components and to be done soon for others.
Table 9 lists the use case coverage of the components discussed in this report. The table shows that for each project use case, there is at least one component developed in the project. Some components, on the other hand, cover more than one use case in the project, indicating project collaboration.

Table 9 FitOptiVis components versus use cases mapping.

UC	HW IPs	SW Ips	Comm. IPs	Others	Total
UC1	3.3 UNICA 3.4 UNIVAQ 3.8 UNISS, UNICA	4.4 UNIVAQ 4.6 AITEK 4.7 UNISS	5.4 ABI		8
UC2	3.9 UTU			6.2 ITI	2
UC3	3.1 TUT				2
UC4		4.1 UGR 4.2 UGR 4.3 HURJA 4.8 HIB	5.1 7SOLS		5
UC5	3.7 CAMEA				1
UC6	3.10 PHL				1
UC7	3.11 PHL				1
UC8	3.2 UTIA	4.5 UWB		6.1 UWB	3
UC9		4.1 UGR 4.2 UGR	5.1 7SOLS 5.2 SCHN		4
UC10	3.2 UTIA 3.5 TASE 3.6 TASE	4.9 UC			5

Table 10 provides an overview of the Intellectual Property protection and availability of the different components. Even if the project is only at the end of the second year, several partners already decided how to protect components, with licenses or keeping them proprietary and confidential. 6 out of 26 components can be already, at least partially, accessed through open repositories (some of them offer evaluation code).

Table 10 Components Intellectual Property main information.

Category	IP	License	Repository
HW	3.1	TBD	TBD
	3.2	evaluation	http://sp.utia.cz/index.php?ids=projects/fitoptivis
	3.3	TBD	https://github.com/neuraghe/NEURAghe
	3.4	MIT	TBD
	3.5 [a]	TBD	TBD
	3.6 [a]	TBD	TBD
	3.7	TBD	TBD

	3.8 [b]	TBD	TBD
	3.9 [c]	LGPL	https://riscv.org/
	3.10	proprietary	N/A
	3.11	TBD	TBD
SW	4.1	TBD	https://github.com/JuanIsernGhosn/homography-calibrator.
	4.2	TBD	https://github.com/DaniDeniz/indoor-action-recognizer
	4.3 [d]	TBD	TBD
	4.4	GPL	TBD
	4.5	TBD	TBD
	4.6	proprietary	N/A
	4.7	GPL/LGPL	TBD
	4.8	proprietary	N/A
	4.9	TBD	TBD
Comm	5.1	proprietary	N/A
	5.2	N/A	N/A
	5.3	TBD	https://github.com/abs-tudelft/fletcher
	5.4	GPL	TBD
Others	6.1	TBD	TBD
	6.2 [e]	N/A	N/A

[a] components will be released as open source even if some of them involve commercial Xilinx sub-components
[b] defines a class of components instead of specific components, that will be described in future releases of this deliverable
[c] CPU core is open source, while voltage and frequency generators are not open
[d] the component adopts different tools, most of them open source
[e] part of the component is already protected by patent, while the other will remain confidential

Lastly, Table 11 shows an overview of the expected evolution for all the components involved in FitOptiVis, in terms of Technology Readiness Level (TRL) at the beginning (@M0) and at the end (@M36) of the project. From the table it is possible to appreciate the variety of the development state for the different components: some of them were only preliminarily studied and envisioned (e.g. component 4.5) at the beginning of the project, while others were yet validated in laboratory or relevant environments (e.g. component 4.8). Also the target development state to be achieved at the end of FitOptiVis differs, from components ready to be commercialized (e.g. component 5.3) to components still under validation (e.g. component 3.3). Moreover, from Table 11 it is clear that all the components at the end of FitOptiVis will be at least at TRL4, meaning that each of them will be at least validated in laboratory.

Table 11 Technology Readiness Level at M0 and at M36 of the different components.

Category	IP	TRL@M0	TRL@M36
HW	3.1	2/3	5/7
	3.2	4	6
	3.3	3	4
	3.4	N/A	N/A
	3.5	1	6

	3.6	1	4
	3.7	3/4	6
	3.8	N/A	N/A
	3.9	0/4	4/9
	3.10	1	6/7
	3.11	4	6
SW	4.1	1	4/6
	4.2	1	4/6
	4.3	2/3	7
	4.4	4	6
	4.5	0	5/6
	4.6	4	6/7
	4.7	3	6/7
	4.8	4/5	6/7
	4.9	2	5
Comm	5.1	3	5
	5.2	2	6
	5.3	4	6/9
	5.4	4	6/7
Others	6.1	0	5/6
	6.2	2	4/5

The variety of starting and target TRL for the different components is also related to the overall project outcomes, expected up to TRL6, and especially to the demonstrators which are expected to provide very different achievements (from TRL4 to TRL7). In summary, the development state of the FitOptiVis components at the end of the second year of the project is in line with the plans. Some components are still under refinement, while others have already started integration to deliver demonstrators, resulting either in standalone PoC or as part of the overall UC ones. So that, next steps for WP5 are focused on components refinement and integration in PoCs and demonstrators. Considering the current overall state of the work, WP5 is on track to deliver the intended demonstrations by the end of the project.