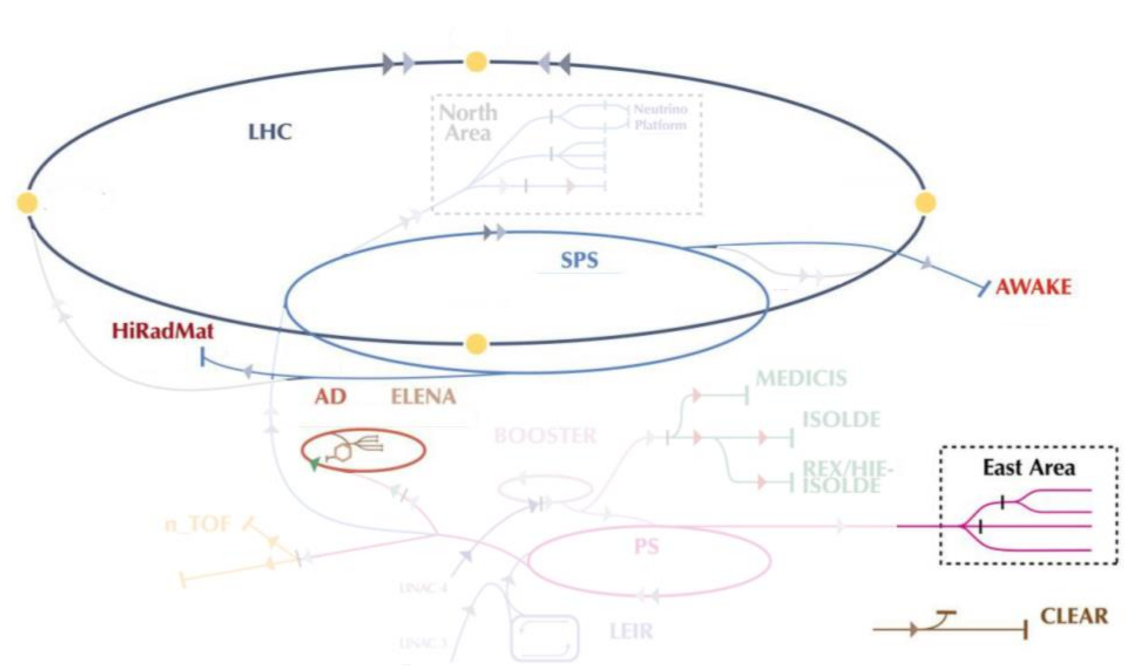
A generic real-time software in C++ for digital camera-based acquisition systems at CERN

A. Topaloudis*, E. Bravin, S. Burger, S. Jackson, S. Mazzoni, E. Poimenidou, E. Senes
Beam Instrumentation, CERN

Abstract

Until recently, most of CERN's beam visualisation systems have been based on increasingly obsolescent analogue cameras. Hence, there is an on-going campaign to replace old or install new digital equivalents. There are many challenges associated with providing a homogenised solution for the data acquisition of the various visualisation systems in an accelerator complex as diverse as CERN's. However, a generic real-time software in C++ has been developed and already installed in several locations to control such systems. This paper describes the software and the additional tools that have also been developed to exploit the acquisition systems, including a Graphical User Interface (GUI) in Java/Swing and web-based fixed displays. Furthermore, it analyses the specific challenges of each use-case and the chosen solutions that resolve issues including any subsequent performance limitations.

Introduction



Digital cameras installations at CERN so far

Analogue cameras are generally considered more **radiation tolerant** than digital cameras. However, they continue to become outdated.

Digital cameras on the other hand have **better image quality** and require **less additional electronics** (e.g. ADC, timing integration, etc.)

There are already **several installations** at CERN featuring digital cameras visualisation systems with **various requirements**.

Location	# cameras	Frame rate (Hz)	Frame size (pixels)	Bandwidth (MB/s)
SBDS	1	35	1450x740	56
AWAKE	27	10	1936x1216	945
	10	N/A		N/A
CLEAR	23	1	2048x1536	108
AD/ELENA	4	N/A	170x170 / 1265x1060	N/A
HiRadMat	6	N/A	440x815	N/A
LHC	1	1	1936x1216	3,5
IRRAD	1	N/A	1028x664	N/A

Challenges

The main challenges in providing a **homogenised acquisition system** is the **variety in the hardware installations** and **in the requirements** of each installation.

Location	Trigger	CPU	Power Control	Requirements
SBDS	Internal	VME	BTVI (DC)	Saturation detection Constantly online Continuous light integration
AWAKE	External	Super Micro	RUCKUS (PoE)	Demanding throughput Auto recovery from SEUs Timing integration in SW Fine synchronization Sanity check maintaining the full resolution
CLEAR	External	VME/ Industrial PC	RUCKUS (PoE)	Multi-camera installation
AD/ELENA	External / Manual	VME	GUDE (DC) / RUCKUS (PoE)	Synchronised manual trigger
HiRadMat	External	VME	RUCKUS	Standard operation (Profiling)
LHC	Internal	VME	N/A	Standard operation
IRRAD	External	Industrial PC	N/A	Standard operation

The main **limitations** of the acquisition system is the **bandwidth** required to transfer the images from the camera to the real-time server and from the server to the clients.

Frame Size (pixels)	Frame Size 8bit encoding	Frame Size 12bit encoding	Data size per frame (clients)	Time needed for grabbing
1936x1216	2,3 MB	3,5 MB	4,7 MB	~40 ms

Acquisition Software

Despite the diversity of the requirements, a **modular acquisition system** has been developed.

While there is a variety in the installed hardware (CPUs, network interfaces and switches, etc.), a **common, generic real-time** software in C++ is responsible for coordinating the acquisition process.

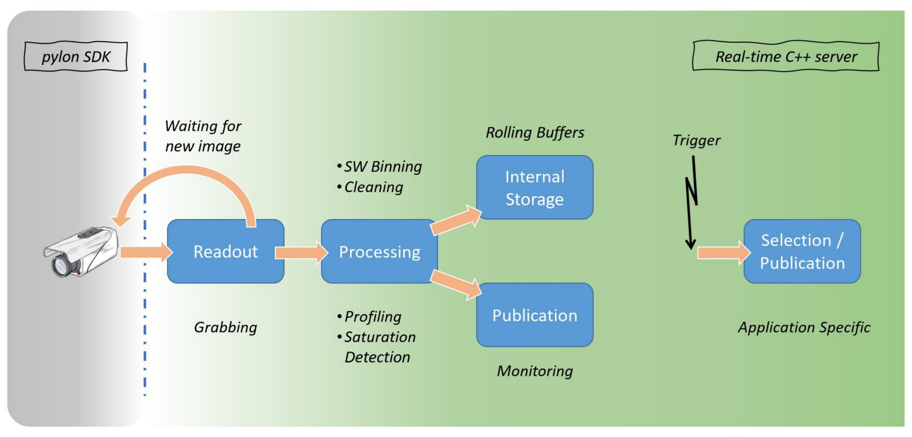
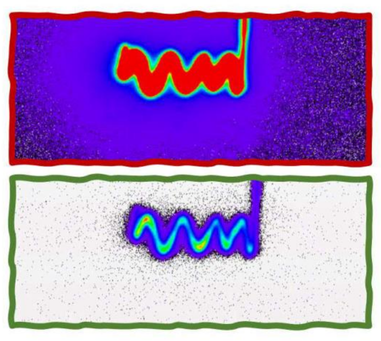


Image acquisition software model



Saturation detection

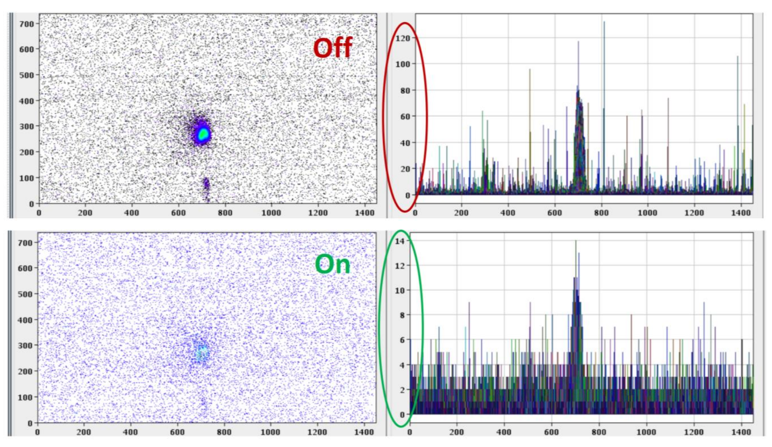
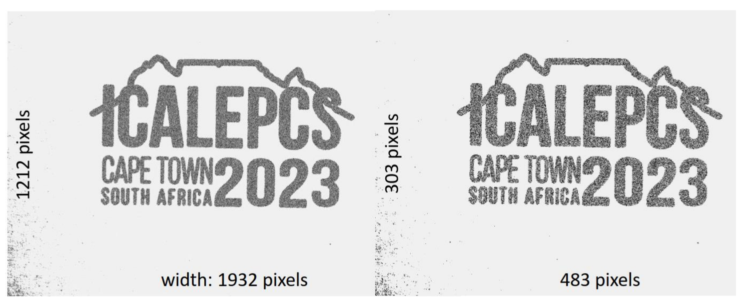


Image cleaning



4x4 Image binning

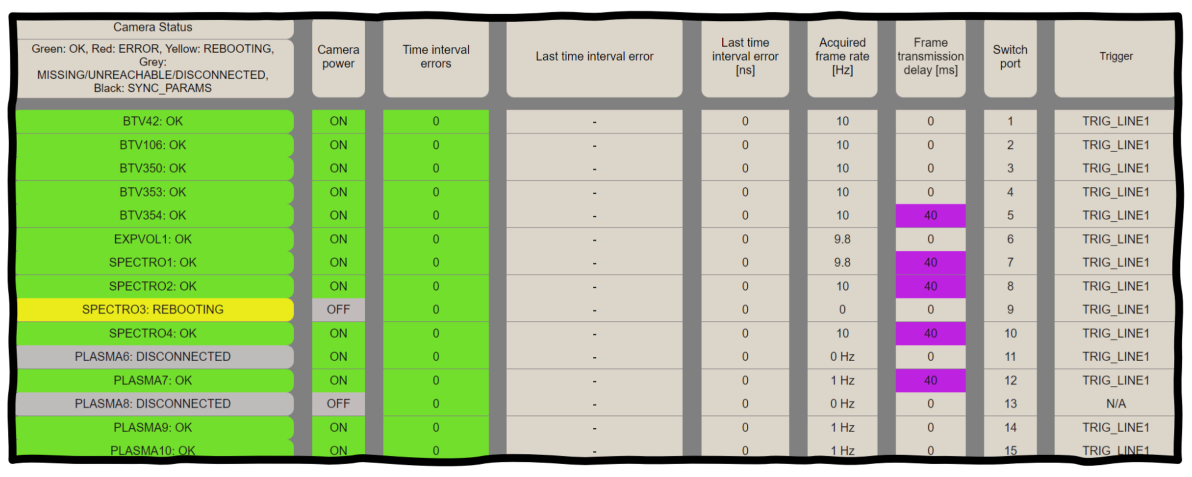
Feature	Remarks
Saturation detection	Best image selection
Network optimization	Demanding throughput
Internal watchdog	Auto recovery from SEUs
SW timing subscriptions	Timing integration in SW
Image timestamping	Fine synchronization
Binning in SW	Sanity checks maintaining the full image resolution
Synchronous SW trigger	When corresponding HW is not available
Image projections	Profiling
Image calibration	Pixel to mm translation
Image cleaning	Background noise removal

Tools

An additional **C++ software** for **controlling** the **power** of the cameras

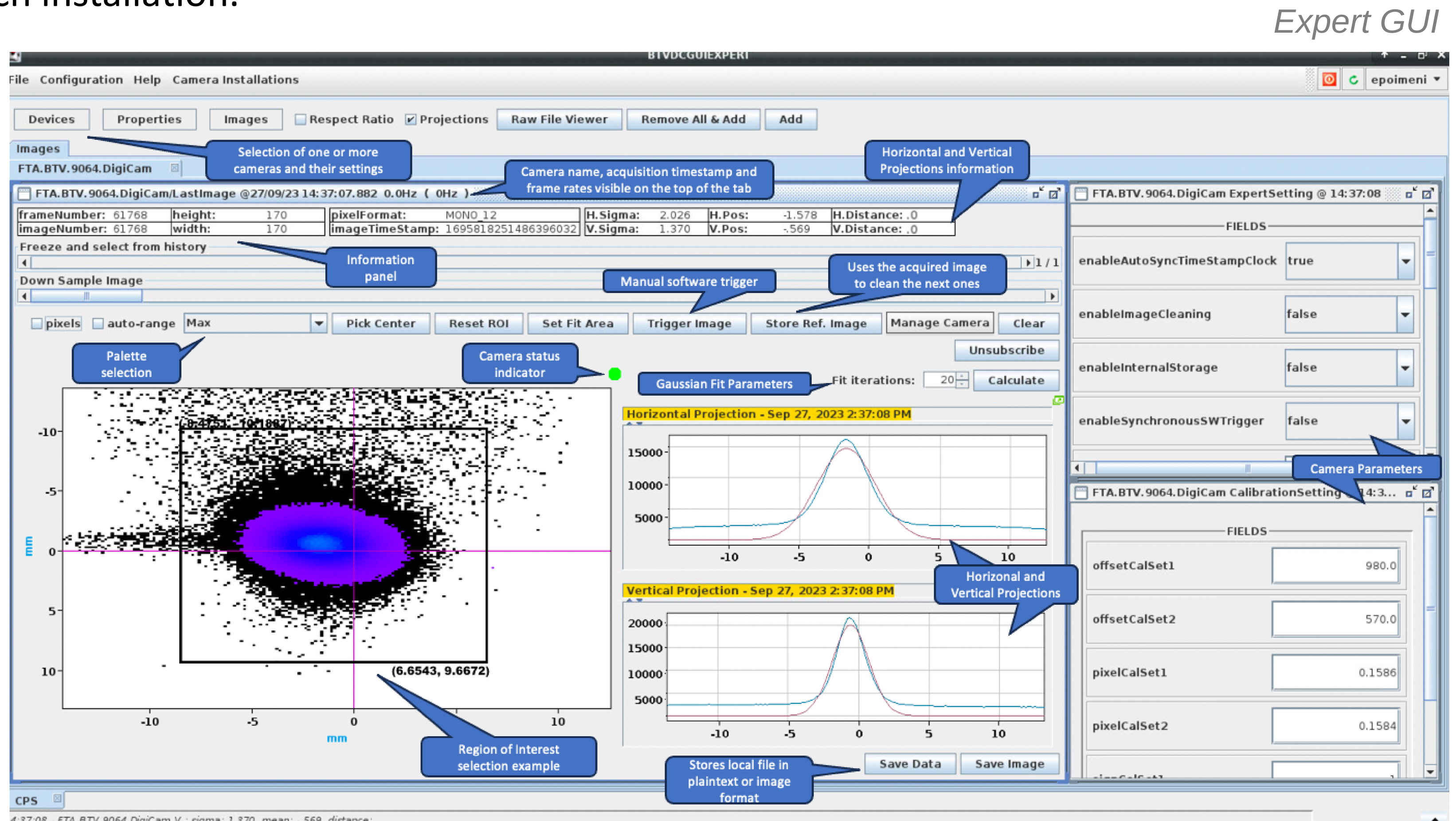
A **GUI** in **Java/Swing facilitating** the use of the system

Web fixed displays to **monitor the status** of each installation.



Fixed status display

Expert GUI



Conclusion

There is an **on-going campaign** to replace increasingly obsolescent analogue cameras used in most of CERN's beam visualisation systems with new digital equivalents. There are many challenges associated with providing a **homogenised solution** for the data acquisition of such systems including the **variety in hardware** and **in the requirements** for each installation.

Despite the diversity of the specifications, a **generic, modular real-time** software has been developed in C++ and installed in several locations to control such systems. The software accommodates the features of the legacy acquisition systems as well as the necessary additional ones including a **synchronous software trigger** and **saturation detection**. It **integrates** the CERN central **timing in software** when the corresponding hardware is not available and maintains an **internal watchdog** to recover from SEUs. Furthermore, it supports **network optimisation** for the most challenging installations as well as **precise image timestamping** for synchronisation.

Lastly, additional tools have been developed to exploit the acquisition systems including additional C++ software for controlling the power of the cameras, a GUI in Java/Swing facilitating the use of the system and web fixed displays to monitor the status of each installation.

*athanasios.topaloudis@cern.ch