ISBN: 978-3-95450-262-2 ISSN: 2673-5350 doi: 10.18429/JACoW-IBIC2025-MOPMO11

# MACHINE VISION CAMERAS FOR BEAM SPOT ANALYSIS

M. A. Bree*, T. D. Batten, W. Javed
Canadian Light Source, Saskatoon, Canada

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2024). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

*Abstract*

In 2020, the Canadian Light Source (CLS) began a project to replace all beam diagnostic analogue cameras with digital ones. Over time, this project has expanded to include beam analysis capabilities. We present an EPICS-based imaging system that uses inexpensive Machine Vision (MV) cameras. The system computes beam parameters including intensity, centroid, ellipticity, eccentricity, orientation, and more.

## INTRODUCTION

The Canadian Light Source is a third generation synchrotron that supports 22 operational beamlines [1]. When the CLS was commissioned in 2005, analogue video cameras were used for beam diagnostics and analysis at various locations. The limitations of these cameras include:

- no (or minimal) remote control,
- no external triggering (they are free running),
- poor image quality (the analogue images need to be digitized to perform any kind of quantitative analysis).

Advances in digital sensor technology have rendered analogue cameras obsolete. Digital cameras now dominate all areas of imaging, including consumer, military, scientific, and industrial applications. In particular, Machine Vision (MV) cameras are a class of camera tailored for industrial applications [2].

The CLS has invested in commercial solutions, such as those from BeamGage [3], for beam spot analysis at select locations. However, these systems are expensive and require integration into our EPICS [4] control system.

In 2020 a project was initiated at the CLS to add inexpensive MV cameras to our instrumentation toolkit. The goals of this project have expanded to include:

- creation of a scalable infrastructure to support an arbitrary number of MV cameras,
- replacing obsolete analogue cameras and associated infrastructure (cables, CRTs, etc.) with MV ones,
- adding cameras to previously unmonitored locations,
- adding quantitative beam spot analysis capabilities (similar to those provided by BeamGage), and
- replacing expensive commercial systems such as BeamGage.

### *Machine Vision Cameras*

Machine Vision is a general term used to describe cameras designed for industrial and automation applications. MV camera features include:

- small and robust construction; fanless,
- simple cabling requirements,
- external triggering,
- access to raw data (i.e., no lossy compression),
- standardized Application Programming Interface (API) support,
- global shutter across all pixels.

The CLS has chosen the **BFS-PGE-16S2M-CS** (`1440 X 1080`, 78 FPS) from Teledyne FLIR [5] for the majority of the cameras in the MV system. This camera is GigE Power over Ethernet (PoE), which means that a single Ethernet cable to a 1 Gigabit/s switch port that supports PoE provides both data and power. The cameras use monochrome CMOS sensors and ADCs that support 8, 10 or 12 bits per pixel. In addition, the cameras support the **GigE Vision 1.2** and **GenICam** standards [6, 7]. Therefore, with the available vendor-supplied software development kit (SDK), these cameras are supported by EPICS **areaDetector** [8].

## SYSTEM ARCHITECTURE

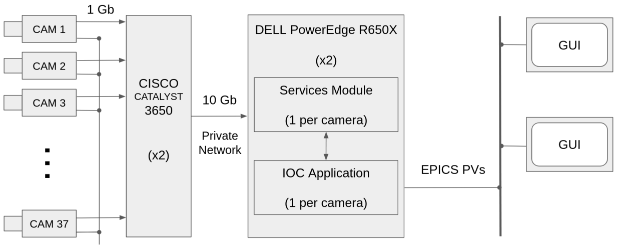


Figure 1: MV camera system architecture.

The system architecture, illustrated in Fig. 1, is very simple. The cameras are connected to the CLS timing system [9] for external triggering. They are connected to 1 Gb ports on two Cisco Catalyst 3650 switches for data and power. The switches are connected via 10 Gb copper links to two Dell EMC PowerEdge R650XS servers running the Debian 12 operating system. The servers are equipped with Intel 82599 10 Gb Ethernet Controller cards, single Xeon 4309Y CPUs, and 32 Gb of memory.

### *Data Rates and Hardware Requirements*

Consider the **BFS-PGE-16S2M-CS** 1440 x 1080 pixel camera. At 12 bits per pixel, 2,332,800 bytes are generated per image (the data is transmitted using 3 bytes per 2 pixels) and the maximum acquisition rate is 43 images per second.

A system with one 10 Gb switch has been stress tested at approximately 600 Mb/s (i.e., 60% capacity) which corresponds to approximately 260 images[1] per second. Above

* michael.bree@lightsource.ca

[1] 1440 x 1080 pixels, 12 bits per pixel, 2,332,800 bytes per image.

600 Mb/s, we begin to observe lost and retransmitted packets. This is not unexpected as the data is transmitted using the UDP protocol [10] in which packets can be discarded under busy conditions.

The number of switches and hosts needed is a function of system requirements. Currently, the CLS system consists of 37 cameras (with more planned). They are run at 12 bits per pixel and one image per second. In addition, simultaneous images from multiple cameras are rarely required. Therefore, a single host and single 48 port switch could suffice for current CLS requirements. However, two switch-servers pairs are used to simplify cabling requirements.

## EPICS IOC APPLICATION

The EPICS Input/Output Controller (IOC) application architecture is shown Fig. 2. It consists of EPICS base v7.0.9 along with **areaDetector** (AD) and the AD plugins **ADGenICam** and **ADSpinnaker**. The proprietary Spinnaker [2] SDK [11] from Teledyne FLIR provides compiled shared libraries that **ADSpinnaker** uses to communicate with the cameras over a private network. While it was necessary to install the Spinnaker SDK and compile the IOC application, no custom CLS code was required. Any AD-supported camera can be seamlessly integrated into the system.

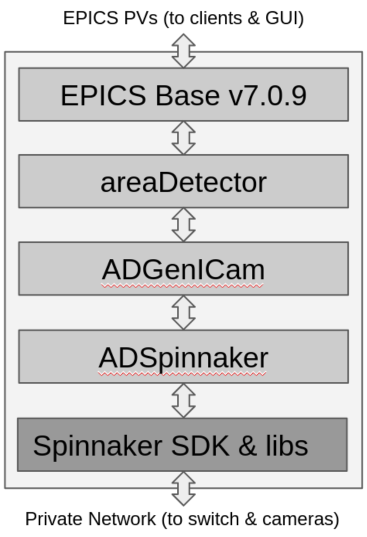


Figure 2: EPICS IOC application architecture.

The IOC application (including SDK) is able to support multiple cameras. However we run one application per camera, which facilitates the ability to add/remove cameras to/from the running system, or restart individual cameras, without impacting other cameras.

## SERVICES MODULE

The Services Module is a pure Python application that serves two primary purposes: beam spot analysis and image saving. These features are described in detail in following sections.

Python was chosen for the Services Module for ease of implementation. It can be deployed, started, and stopped independently of the IOC application. Performance is not an issue because analysis and save are implemented using the optimized `numpy`, `scikit-image` and `scipy` libraries. It uses EPICS for its API—the system Grahical User Interface (GUI) can configure, enable, and disable image analysis and save features via EPICS Process Variables (PVs).

The Services Module uses `pyepics` to monitor PV updates via asynchronous callbacks. PVs are not read directly, which prevents the system from stalling if the IOC application becomes unresponsive or is not running. There is one notable exception to this patten however. The Services Module monitors the `.HASH` field of the waveform PV that contains the image. It then reads the waveform PV directly if and only if it needs the image (i.e., to analyze or save). This avoids the need to monitor and retrieve large PVs if the data is not required. Both the analysis and save features can run without the GUI running.

### *Beam Spot Analysis*

The Services Module uses the Python library `scikit-image` to do beam spot analysis. In particular, `scikit-image.draw` is used to crop the image into circular, rectangular, or elliptical regions of interest (ROIs). The rectangular and elliptical ROIs can be rotated to any arbitrary angle. For the actual analysis of the ROI, `scikit-image.measure` is used. This computes the centroid, orientation (angle), axes (major and minor), ellipticity, $d4\sigma$ (major and minor), beam profile, and other parameters. If a background image is specified, it is subtracted from the image of interest before the analysis is done. The results are published in EPICS PVs for GUI display and archiving purposes.

The EPICS **areaDetector** and its plugins do support ROIs. However, implementing the ROI cropping and analysis outside of **areaDetector** allows the analysis of both live and saved images.

### *Image Save*

The Services Module implements a “bulk save” feature that can save every acquired image. A total size limit can be specified, and the system configured to stop acquisition or delete oldest images if the size limit is reached. The `numpy.savez` function is used to save files. Along with the image, a Python dictionary of metadata such as acquisition time, camera identifier and settings, and other parameters, is saved.

## GRAPHICAL USER INTERFACE

The Graphical User Interface (GUI) is implemented using the `PyQt` framework. It relies on the Python modules `pyqtgraph` for rendering images and graphics, and `numpy` for image manipulation. Multiple GUIs can be run simultaneously. Like the Services Module, the GUI relies on

[2] Spinnaker is a Teledyne FLIR brand name.

© Content from this work may be used under the terms of the CC BY 4.0 licence (© 2024). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-262-2 ISSN: 2673-5350 doi: 10.18429/JACoW-IBIC2025-MOPMO11

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2024). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

**pyepics** asynchronous monitor callbacks for PV values to ensure it never stalls if any PVs become unresponsive or lost.

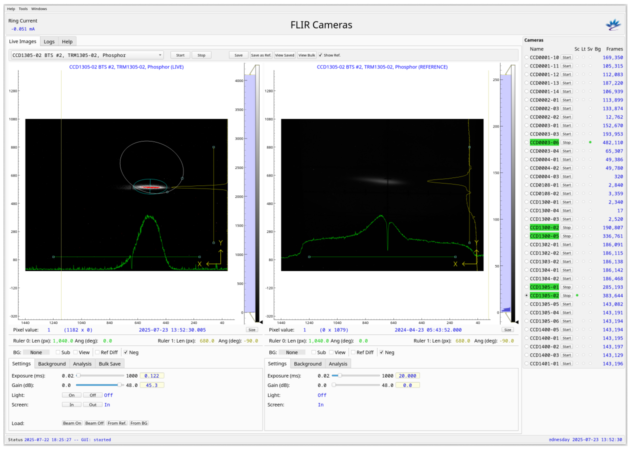


Figure 3: Camera system GUI with live image (left), reference image (center) and system status (right).

The main GUI window is illustrated in Fig. 3. On the left is a live image from the selected camera. The optional right image is a reference for users to visually compare live and expected beam spots. Below are controls for camera settings, background image acquisition, and analysis configuration and results. On the right is a complete system overview including active cameras, analyze, save, screen, light, and background image status.

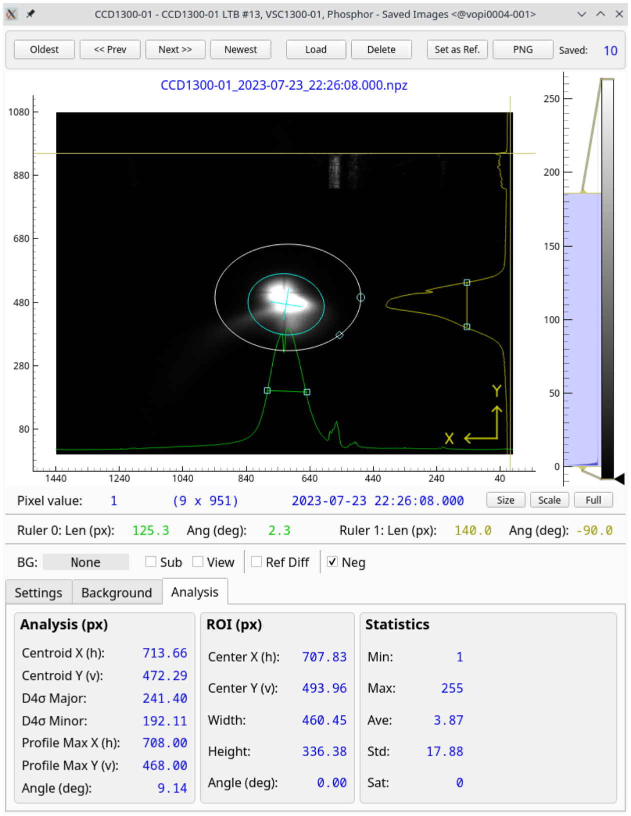
Figure 4 provides a more detailed view, in this case of a saved image. The white ellipse is the configurable ROI. The turquoise ellipse is a visual representation of the beam spot analysis, including the centroid and major and minor axes. The ROI can be adjusted at any time on live or saved images. The beam profile (green and yellow histograms) can be specified to be one of: 1) the sum of all rows and sum of all columns, 2) the row and column at the computed centroid, or 3) the row and column at the cursor position.

## IMPLEMENTATION CHALLENGES

The UDP camera traffic can be discarded under busy conditions. If too many packets are lost, the SDK stops communicating with the camera and the IOC application requires a restart. This problem was mitigated by configuring a packet delay wherein the camera inserts delays between transmitted packets so they arrive spread out rather than in bursts, utilizing a 10 GB switch, and utilizing a server class host computer optimized for high levels of I/O.

## CONCLUSION

With tuned camera settings and capable equipment, the MV camera system has been reliable and successfully used to commission the new CLS linac [12]. The use of open source tools, including Python for the analysis/save features, has facilitated fast and reliable implementation. In the immediate future we plan to replace expensive commercial



Saved: 10

Pixel value: 1 (9 x 951) 2023-07-23 22:26:08.000

Ruler 0: Len (px): 125.3 Ang (deg): 2.3 Ruler 1: Len (px): 140.0 Ang (deg): -90.0

BG: None ☐ Sub ☐ View ☐ Ref Diff ☑ Neg

| **Analysis (px)** | | **ROI (px)** | | **Statistics** | |
|---|---|---|---|---|---|
| Centroid X (h): | 713.66 | Center X (h): | 707.83 | Min: | 1 |
| Centroid Y (v): | 472.29 | Center Y (v): | 493.96 | Max: | 255 |
| D4σ Major: | 241.40 | Width: | 460.45 | Ave: | 3.87 |
| D4σ Minor: | 192.11 | Height: | 336.38 | Std: | 17.88 |
| Profile Max X (h): | 708.00 | Angle (deg): | 0.00 | Sat: | 0 |
| Profile Max Y (v): | 468.00 | | | | |
| Angle (deg): | 9.14 | | | | |

Figure 4: Image with elliptical ROI (white), analysis result (turquoise), and X/Y beam profile (green/yellow).

BeamGage cameras with inexpensive MV cameras for beam spot analysis throughout the CLS.

## ACKNOWLEDGMENTS

The authors would like to thank J. M. Vogt, K. McKeith, and A. Hinz from the CLS Beam Instrumentation group for their contributions to this project. Special thanks to Mark Rivers (APS) for help with **areaDetector**.

The work described in this paper was performed at the Canadian Light Source, a national research facility of the University of Saskatchewan, which is supported by the Canada Foundation for Innovation (CFI), the Natural Sciences and Engineering Research Council (NSERC), the Canadian Institutes of Health Research (CIHR), the Government of Saskatchewan, and the University of Saskatchewan.

## REFERENCES

[1] Canadian Light Source Beamlines, `https://www.lightsource.ca/facilities/beamlines/where-to-start.php`

[2] Association for Advancing Automation (A3), `https://www.automate.org`

[3] Ophir Optronics Solutions, `https://www.ophiropt.com`

[4] Experimental Physics and Industrial Control System (EPICS), `https://epics.anl.gov`

[5] Teledyne FLIR, `https://www.flir.ca`

[6] Association for Advancing Automation (A3), “GigE Vision Version 1.2”, Jan. 21, 2010. https://www.automate.org/vision/vision-standards

[7] European Machine Vision Association, “GenICam”, https://www.emva.org/standards-technology/genicam

[8] M. Rivers, “areaDetector”, https://github.com/areaDetector

[9] T. D. Batten, “Timing system at the Canadian Light Source”, presented at IBIC’25, Liverpool, England, Sept. 2025, paper WEPMO15, this conference.

[10] User Datagram Protocol (UDP), https://www.ietf.org/rfc/rfc768.txt

[11] Teledyne FLIR, “Spinnaker SDK Version 4.2.0.88”, https://www.flir.ca

[12] T. D. Batten and M. A. Bree, “Beam instrumentation for the new linear accellerator at the Canadian Light Source”, presented at IBIC’25, Liverpool, England, Sept. 2025, paper MOPMO15, this conference.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2024). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.