ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-THP09

SMART VIDEO PLUG-IN SYSTEM FOR BEAMLINE OPERATION AT EMBL HAMBURG

E. Galikeev†, U. Ristau, C. Blanchet, D. v. Stetten, V. Palnati, S. Fiedler
all European Molecular Biology Laboratory (EMBL), Hamburg Unit
c/o DESY, Hamburg, Germany

Abstract

Fast data collection, image processing, and analysis of video signals are required by an increasing number of applications at the EMBL beamlines for structural biology at the PETRA III synchrotron in Hamburg, Germany.

Consequently, a new Smart Video Plug-in system has been designed in-house to meet the needs by combining video capture, machine learning, and computer vision with online feedback for motion control. The new system is fully integrated into TINE: data acquisition, and experiment control system.

In this paper, the architecture of the new video system is described and use cases relevant to beamline operations are presented.

INTRODUCTION

Recent projects carried out at EMBL Hamburg synchrotron beamlines for structural biology showed the necessity for a flexible tool that combines modern image processing libraries, video codecs, and motion control support for such experimental workflows like:

- Sample positioning using piezo stage control
- Control of the focus of the sample observation optics
- Image processing (e.g. filtering, denoising)
- Object detection and image recognition using machine learning techniques.
- Fast video recording and compression for offline and online sample motion analysis

The usage of the OpenCV library for image detection and computer vision at synchrotron beamlines was studied by Gofron and Watson [1] who showed that it can be applied for sample detection and centering robotic mounting, and evaluation of x-ray beam properties.

TINE at PETRA III [2] is a cross-platform, scalable distributed control system covering the majority of beamline operations and can be used with various OS like Windows, MAC/OS, and Linux. One of the core concepts of the TINE system is the "TINE Server" which represents a unique entity in the control network. It exports properties as endpoints used by the other TINE servers and clients to communicate with each other.

The new video system uses the full potential of the TINE API to provide means for the building and deployment of a new server as well as utilizing the centralized TINE alarming and archiving functionalities.

ARCHITECTURE

There are several reasons for choosing the Microsoft .NET framework [3] as a platform for integrating 3rd party functionality and user algorithms into the TINE system:

- .NET is a free and open-source, cross-platform software development framework.
- It enables interoperability between native C/C++ to preserve and take advantage of existing investments in unmanaged code.
- There exists wide official API support of the .NET platform among motion control and video solutions manufacturers.
- TINE .NET API allows getting full access to the EMBL beamline control system [4].
- C# is a powerful, modern, object-oriented type-safe programming language for rapid application development.
- Bindings for the NumPy and OpenCV libraries as well as such C# language features like LINQ provide extensive data processing toolsets for the experiment scripts.

The plug-in architecture [5], as shown in Fig.1, was chosen to meet the needs of the flexibility of adding new functionality (e.g. new camera or motion controller API support). The plug-in modules are meant to be separate containers for the additional features or user code that is designed to enhance the main functionality.

The core system provides interfaces to which plugins can connect, creating TINE server instances and routing data exchange between the plug-ins, and offering external communication with clients using standard TINE data transfer mechanisms. Also, the core system provides logging, native 3rd library support, and system configuration functionality.

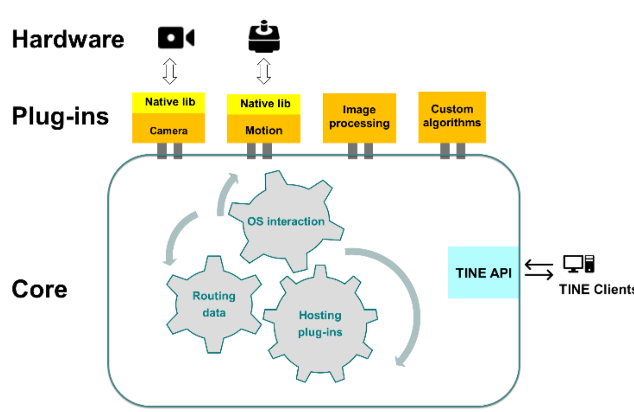

Figure 1: Architecture of the Smart Video System.

†emil.galikeev@embl-hamburg.de

ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-THP09

Microsoft .NET framework supports the dependency injection (DI) software design pattern, which is used for achieving Inversion of Control (IoC) between classes and their dependencies. This pattern allows class A to call methods on an abstraction that class B implements, making it possible for class A to call class B at run time, but for class, B to depend on an interface controlled by class A at compile time (thus, inverting the typical compile-time dependency) [6]. At run time, the flow of program execution remains unchanged, but the introduction of interfaces means that different implementations of these interfaces can easily be plugged (e.g. new video cameras or motion controllers) within one application.

The decoupled architecture allows easy unit testing using mock objects for simulation of the behavior of complex real objects. It is extremely useful in the case of testing experimental scripts without involving the full hardware data acquisition loop.

Each plug-in implements a certain device or library-specific logic.

Video Camera Plug-In

- Receiving raw images captured from specific cameras
- Video streaming into the beamline control environment
- H.264 compression and transfer into the video storage for the offline sample motion analysis.

Image Processing Plug-in

- Processing raw images using OpenCV
- Receiving inputs for image processing algorithms from configuration and user interface
- Providing processed output for different consumers (e.g. lens autofocusing, sample positioning).

Motion Control Plug-In

- Piezo stage control using native libraries of SmarAct, PI, and other vendors
- Stepper and DC motor control via the Common Device Interface (CDI) in TINE.

USE CASES

Sample Exposure Unit at P12 Beamline for Small Angle X-ray Scattering (SAXS) Experiments

The design and installation of the new multi-purpose sample exposure unit (SEU2B) for scanning small-angle scattering experiments at the P12 SAXS beamline was carried out in collaboration with EMBL teams in Hamburg and Grenoble and with ESRF. The SEU was designed to accommodate different types of samples and sample holders such as bones and tissues for scanning SAXS experiments or microfluidic chips. One of the main purposes of SEU is to observe the microscopic sample while it is exposed to an X-ray beam during scanning. The installation (see Fig 2a,b) includes an on-axis sample observation camera (UEye, IDS GmbH, Germany) with a motorized lens system (Precise Eye Fixed Lens System, Navitar, USA) and a piezoelectric stage with motion control (Physik Instrumente, Germany). With the on-axis camera, the area of the sample to be scanned and exposed can be visualized during the SAXS experiments. Images of a microscopic sample can be collected in parallel with scanning SAXS measurements and microfluidic cells can be precisely aligned. Beamline control is available via the graphical user interface BECQUEREL [7], a Linux QT5-based application (see Fig 3). User experiment scripts are executed via the "meta server" which communicates with the Smart Video System via exported properties of the TINE Server.

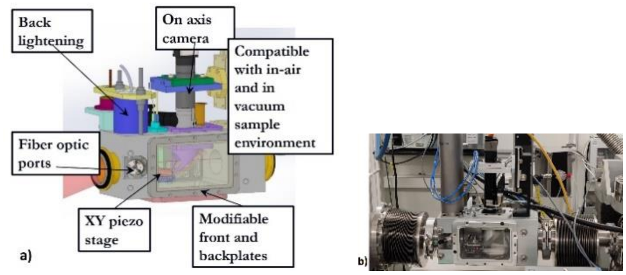

Figure 2: Sample Exposure Unit a) Model, b) Installation at the Beamline.

Video camera, image processing, and motion control plug-ins were added as part of the system configuration, providing manual and automatic focus control, and video streaming with image processing support for the beamline users. Currently, the piezoelectric stage control with 1D and 2D scanning is implemented to run directly through the BECQUEREL suite and the transfer of this functionality to the new system is underway.

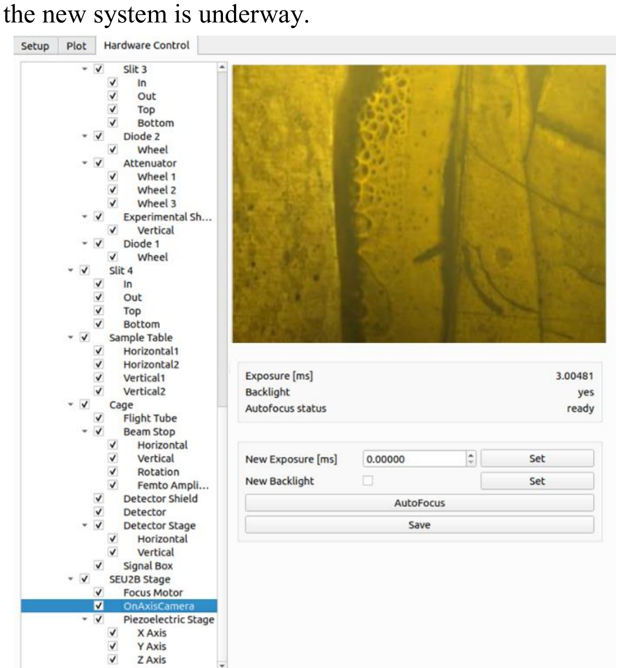

Figure 3: BECQUEREL GUI (SEU2B camera control tab).

Beamline P14.2 Time-Resolved Crystallography Experiments

Time-resolved experiments at the P14.2(T-REXX) beamline require sample delivery with fast-moving patterned silicon chips (~30 sample positions per second).

ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-THP09

Fast video recording is essential for correct optical sample observation. For this purpose, a new high frame rate (~1000 fps) camera (XStream, IDT, USA) was installed and a beamline video server was developed using the TINE C++ framework. The server includes fast video recording and streaming capabilities with full control from the MXCuBE graphical user interface (see Fig 4). Most of the functionality of this new video server is currently being integrated into the camera plug-in. In addition, a motion control plugin for the SmarAct precision motion control system [8], to be used in the sample alignment part of the experiment scripts, is planned to be developed. Video compression is achieved via H.264 encoder from VideoLAN [9] with the video files automatically transported into the storage.

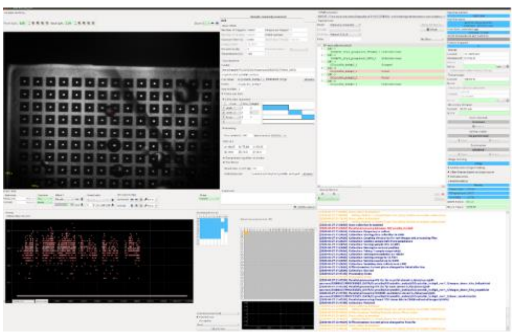

Figure 4: MXCuBE, Beamline Control Software for Time-Resolved Experiments at P14.2 Beamline [10].

SUMMARY AND FURTHER DEVELOPMENTS

The new TINE video system, developed with Microsoft .NET technologies, has shown to be a viable solution for current imaging and motion control tasks at EMBL Hamburg. Currently, the new system is under commissioning and intensive testing. Also, new features are being considered. Based on the feedback from beamline users, we plan to add new functionalities like the detection of sample distribution anomalies in T-REXX and the detection of misalignments of sample pins for the MARVIN robotic sample changer system [11]. Applying GPU solutions to accelerate the performance of resource-hungry image processing and machine learning tasks is being envisaged as well.

ACKNOWLEDGEMENT

The authors would like to thank the EMBL Grenoble instrumentation team for the collaboration in the SEU2B project and Andrey Gruzinov, DESY for his assistance in testing the new product.

REFERENCES

[1] K. Gofron and W. Watson, “Computer Vision at NSLS-II synchrotron beamlines”, *Research Gate*, 2016. doi:10.13140/RG.2.2.24463.79524

[2] TINE website, http://tine.desy.de

[3] Microsoft .NET documentation, https://docs.micro-soft.com/

[4] U. Ristau *et al.* “The EMBL Beamline Control Framework BICFROK”, in *Proc. PCaPAC'14,* Karlsruhe, Germany, WPO 012, pp.60-62.

[5] M. Richards *“Software Architecture Patterns”*, O'Reilly - Media, Inc., 2015.

[6] S. Smith “*Architecting Modern Web Applications with ASP.NET Core and Azure”,* Microsoft Corporation, 2022.

[7] N.R. Hajizadeh *et al.*,“Integrated beamline control and data acquisition for small-angle X-ray scattering at the P12 BioSAXS beamline at PETRA III storage ring DESY”, *J. SynchrotronRadiat.*, pp 906-914, 2018. doi:10.1107/S1600577518005398

[8] SmarAct website, https://www.smaract.com

[9] VideoLAN website, https://code.videolan.org/videolan/x264

[10] M. Oscarsson *et al.,* “MXCuBE2: the dawn of MXCuBE Collaboration”, vol. 26, *J. Synchrotron Radiat.*, pp. 393-405. 2019. doi:10.1107/S1600577519001267

[11] U. Ristau *et al.,* “Marvin Update’ the Robotic Sample Mounting System at the Embl-Hamburg”, in *Proc. PCaPAC'18*, Hsinchu City, Taiwan, Oct. 2018, pp. 163-166. doi:10.18429/JACoW-PCaPAC2018-THP03