ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-THP18

STATUS, RECENT DEVELOPMENTS AND PERSPECTIVE OF AVINE VIDEO SYSTEM

Stefan Weisse*, David Melkumyan, DESY, Zeuthen, Germany
Philip Duval, DESY, Hamburg, Germany

Abstract

DESY's TINE-powered Video System, originally released in 2002, was last presented in 2011 at ICALEPCS [1], at this time not yet known under the name Advanced Video and Imaging Network Environment (AVINE). AVINE provides a framework and toolkit for operators, physicists and technicians related to, but not limited to, Ethernet-based imaging at accelerator facilities. Over the past decade, the major emphasis was put on extended support, incorporating user requests, migrating to the latest Windows and Linux operating systems and the latest Java Virtual Machine, all while replacing legacy GigE Vision APIs in order to support past, current and future camera hardware. In this contribution, the current status, layout, recent developments and perspective of AVINE is described. The focus will be on experience migrating to future-oriented (still under vendor support) GigE Vision APIs, the recently upgraded image (sequence) file format, and first experiences on Windows 11.

OVERVIEW

The Video System explained further on originates at the Photo Injector Test Facility at DESY in Zeuthen (PITZ) [2]. Started in 2002 as a test facility at DESY for research and development on laser-driven electron sources for Free Electron Lasers (FEL) and linear colliders, PITZ has been extended over the last years in order to study different applications of photo injectors, e.g. for the generation of THz SASE light [3]. A significant milestone, first emission of THz light from the SASE FEL, has been reached in August 2022 [4].

Despite its to some extent different requirements, AVINE has been exported over the years to accelerator setups on DESY campus in Hamburg, namely PETRA III (e.g. machine diagnostics [5], user beamlines and so-called beam paths E-Weg and L-Weg), REGAE, SALOME and User Beamlines of Petra III at EMBL Hamburg. In addition, small installations have been realized over the years as well, e.g. monitoring and positioning of an electron beam at an electron welding machine in mechanics workshop and standalone AVINE video installations on notebooks for live video, data taking and analysis, which can also be used offline/off-site.

Ethernet-based industrial vision cameras which are conforming to standards GigE Vision and GenICam can easily be connected and used. In general, image acquisition is done with a repetition rate of 1 to 20 Hz, either initiated by external trigger signal (TTL pulse) or by an internal clock of the camera. Most cameras are greyscale (non-colour) with 8 to 12 bits per pixel. AVINE is integrated into the TINE control system [6]. Network transport of video image stream is done via TINE, either via TCP or multicast. To use available network bandwidth more effectively, JPEG compression can be applied on network transport. As a trade-off, greyscale pixel bit depth is limited to 8 bits when using JPEG.

Image size varies from hundreds of kilo- to several megapixels. Colour images (RGB 24 bits) are supported, however client GUIs and applications for measurements and analysis are mainly designed for greyscale data.

The Universal Slow Control System (USC) is designed to control various types of cameras used in the AVINE Video System, providing a common approach to configuring, displaying and controlling camera slow control settings required for PITZ operation, e.g. exposure time and analogue video signal gain. The USC system consists of a USC client, a USC server, and a USC Server Configurator implemented in Java. The USC Client is a TINE-based GUI client application that includes the functionality required by an operator to monitor and control the settings of all available cameras in the Video System. The USC Server is a TINE-based server that provides slow control of various types of cameras using various communication protocols (e.g. RS-232 for analogue cameras, TINE-based for controlling Gigabit Ethernet cameras via AVINE SGP servers). The USC Server Configurator (USCSC) is a GUI application for easy configuration of the USC Server, allowing server maintenance personnel to use predefined templates, modify them, or create a new template from scratch. It also checks server configurations and displays appropriate errors and troubleshooting tips to avoid unexpected USC server behaviour at runtime.

A versatile easy-to-use albeit aged Windows client application called Video Client 3 can be used for live view, data taking, measurements and on/off-line analysis.

With regards to platform independence, a JAVA component called TINE ACOP Video bean is available, which acts as a basis to create client-side GUIs with special functionality based on the experimental setup. The Video bean is also integrated into the TINE Instant Client, a basic Java GUI to browse, read and write TINE properties.

LAYOUT

The general layout of AVINE is shown in Fig. 1. AVINE is following a component (object-oriented) based approach. The idea is to hide implementation details and only expose what is necessary, so that changes in implementation are mainly kept inside a single component. For example, migration to a different API to interface cameras is only reflected in the front-end server (so-called Small Grabber Part (SGP)).

* stefan.weisse@desy.de

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-THP18

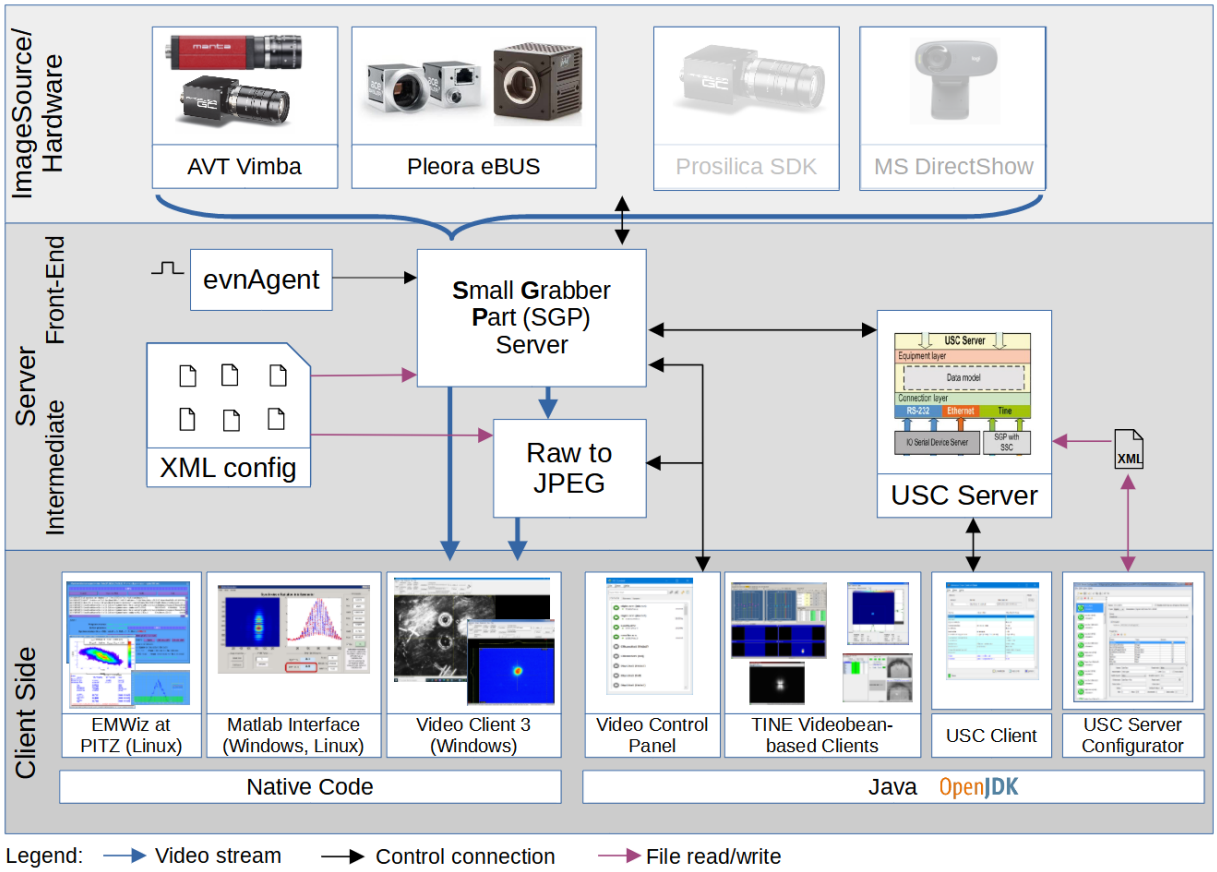

Figure 1: Simplified Layout of AVINE Components and their Interaction.

All other components continue to work as before, without need for modification. Server-side components running on the same computer can be interconnected efficiently using shared memory. From server to client side, the TINE protocol is utilized, using either TCP or UDP multicast. A dedicated well-defined image type is used to transport video images. Under increasing bandwidth demands and in general busier networks, multicasting video images via UDP proved to have stability issues. In the future, it is foreseen to change the default transport of video images to TCP where UDP Multicast is still used.

Server-side components running on the same computer can be interconnected efficiently using shared memory. From server to client side, the TINE protocol is utilized, using either TCP or UDP multicast. A dedicated well-defined image type is used to transport video images. Under increasing bandwidth demands and in general busier networks, multicasting video images via UDP proved to have stability issues. In the future, it is foreseen to change the default transport of video images to TCP where UDP Multicast is still used.

STATUS

All current production-level server-side and client-side installations are stable and running on monthly security patched Windows 10 (Server, LTSC, Enterprise, Pro) computers. On Linux (EMWiz, Matlab Interface), Red Hat Enterprise Linux based Scientific Linux 7 (SL7) is used. Ubuntu, being considered during implementation and testing, should work as well.

RECENT DEVELOPMENTS

As of April 2020, the JAI SDK and Control Tool, used to interface cameras of companies JAI and Basler, is no longer supported. In addition, PvAPI, widely used to interface Prosilica product line of cameras by vendor Allied Vision Technologies (AVT), has been superseded by AVT Vimba SDK over the past 10 years.

As a successor of JAI SDK for supporting modern JAI cameras (CMOS sensor), JAI recommends eBUS SDK by vendor Pleora. Test have shown that Pleora eBUS is flexible in terms of which cameras can be operated with it, so it was decided to be used as a successor for already existing AVINE camera installations where JAI SDK was used.

Currently, v6.1 of eBUS SDK is used. In contrast to previous APIs like JAI SDK and Prosilica PvAPI, eBUS SDK requires a paid license. A subscription license is required for developer support. A runtime license is in general required, too (one-time payment, bound to MAC address or USB dongle). This complicates development and maintenance. It would not have been chosen if there had been an easier option available. For some JAI cameras, no runtime license is required, because the license is part of the camera

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-THP18

firmware. Video servers that use Pleora eBUS as interface to cameras are already running in production.

To have a second API option for the many Prosilica cameras installed available, Vimba SDK is going to be used as a successor of Prosilica PvAPI. No extra license requirements, the lower cost and regular updates from the manufacturer's website (open access) eases development and use. Proof of concept tests using Vimba 4.2 and 6.0 have been finished successfully. Productive installation is foreseen, the Vimba-based video server needed for that is to be implemented in the coming months.

It must be noted that experience while developing and using eBUS SDK and Vimba SDK showed that the discontinued APIs were superior in terms of ease-of-use, complexity and flawlessness. This is not a good trend. Moreover, accessing Pleora eBUS SDK requires an account and login credentials. During a year of development subscription, no updates to the software were provided.

File Format IMC2

In order to store raw loss-less images, image sequences and background images, proprietary yet well-defined file formats had been designed in the years 2002 to 2005 and are still used today. For uncompressed storage, the file formats IMM (Image Multiple) and BKG (BacKGround image) support greyscale images with very limited metainformation to image: width, height, and a so-called scale factor (pixel to millimetre ratio for both width and height). In successive years, the compression of images was implemented to save disk space in file formats IMC (Images Compressed) and BKC (BacKground Compressed). In 2008 there was an attempt to supersede these file formats by using standardized file-formats PNG (widely used) and MNG, but in the end this attempt could not be completed.

In the meantime, detailed metainformation (e.g. scale factors, time stamp, event number, camera port name (location of camera) ...) are distributed with each image in a so-called image header when transferred via control system protocols TINE and DOOCS over the network at DESY.

Following a request to be able to store individual scale factors for x (width) and y (height), it was decided to implement a new file format for this. A feasible approach from the point of view of using widely adopted standards would be to put PNG files into a ZIP-file-like container, which could easily be viewed by standard tools. But having tight time constraints while aiming to keep the new file format as simple as possible and maintaining full open source policy, this idea cannot be implemented in a reasonable amount of time.

As an alternative, a new proprietary hence well-documented file format (IMage Compressed 2 (IMC2)) was designed, documented and implemented. A sketch of IMC2 file structure is shown in Fig. 2.

The new file format is going to store the detailed metainformation contained in the image header as human readable text (key=value) side-by-side with the compressed (zlib [7]) binary image data.

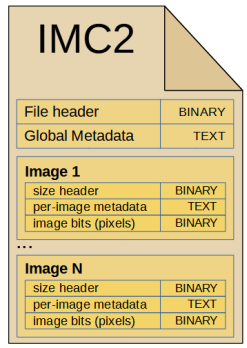

Figure 2: IMC2 File Format.

A release candidate of the file format has been designed and documented. The AVINE C library, available for 64-bit Windows and Linux, has been upgraded to provide functions to save and load the new file format. Based on the C library fundament, further integration into Matlab (mex files) and existing multipurpose application Video Client 3, is subject to happen in the next months.

Windows 11

As end of life of Windows 10 is only few years away, firsts tests of AVINE software have been done on Windows 11 (21H2, patch level 2022-08 and 2022-09). At DESY, a Windows 11 rollout is currently in the early stages, which limits the effectiveness and possibility of qualified tests. Nevertheless, tests of AVINE client and server software and of camera SDKs like eBUS and Vimba on Windows 11 look promising. No changes were necessary, one could simply continue to use existing software. But as time available for testing was short, it cannot be excluded that something was overlooked.

OUTLOOK

It is foreseen to continue supporting already existing camera installations of different camera vendors (AVT, Basler, JAI), e.g. at PITZ, PETRA III, REGAE and SALOME, possibly continuing support also at EMBL Hamburg. In the next months, priority will be the integration of new file format into Matlab and Video Client 3. Afterwards, implementation of Vimba SDK-based video server needs to be completed with the aim of discontinuing the use of PvAPI in the next years. Once ready, Allied Vision CMOS-based cameras Manta G-158B and G-319B, foreseen as replacement of aged Prosilica CCD cameras, can be tested under real accelerator conditions. Having only one productive use case at the moment, the video server which uses Microsoft DirectShow to get the video stream from hardware (Webcam) may be discontinued in future.

As the TINE control system protocol, used throughout AVINE for communication over the network, is planned to be abandoned in the next years, migration to DOOCS control system [8] will be attempted.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-THP18

REFERENCES

[1] S. Weisse, D. Melkumyan, and P. Duval, "Status, Recent Developments and Perspective of TINE-powered Video System, Release 3", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, paper MOPMS033, pp. 405-408.

[2] F. Stephan, C.H. Boulware, M. Krasilnikov, J. Baehr, *et al.*, "Detailed characterization of electron sources yielding first demonstration of European X-ray Free-Electron Laser beam quality", *Phys. Rev. ST Accel. Beams*, vol. 13, p. 020704, 2010. doi:10.1103/PhysRevSTAB.13.020704

[3] P. Boonpornprasert, G. Georgiev, G. Koss, M. Krasilnikov, *et al.*, "Extension of the PITZ Facility for a Proof-of-Principle Experiment on THz SASE FEL", in *Proc. FEL'19*, Hamburg, Germany, Aug. 2019, pp. 38-40. doi:10.18429/JACoW-FEL2019-TUP001

[4] M. Krasilnikov *et al.*, "First Lasing of the THz SASE FEL at PITZ", presented at the FEL2022, Trieste, Italy, Aug. 2022, paper MOA08, to be published.

[5] A. I. Novokshonov, A. P. Potylitsyn, and G. Kube, "Two-Dimensional Synchrotron Radiation Interferometry at Petra III", in *Proc. IPAC'17*, Copenhagen, Denmark, May 2017, pp. 177-179. doi:10.18429/JACoW-IPAC2017-MOPAB042

[6] TINE (Three-fold Integrated Networking Environment), http://tine.desy.de

[7] zlib Compression Library, https://zlib.nett

[8] DOOCS (The Distributed Object-Oriented Control System Framework), http://doocs.desy.de

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI