

# Aravis 及び macOS を使用した GigE カメラによるスクリーンモニタ 画像データ取得解析システムの開発

## DEVELOPMENT OF GigE CAMERA IMAGE DATA ACQUISITION AND ANALYZING SYSTEM FOR SCREEN MONITOR USING Aravis AND macOS

柳田謙一<sup>\*,A,B)</sup>, 清道明男<sup>A,B)</sup>

Kenichi Yanagida<sup>\*,A,B)</sup>, Akio Kiyomichi<sup>A,B)</sup>

<sup>A)</sup> Japan Synchrotron Radiation Research Institute (JASRI)

<sup>B)</sup> National Institutes for Quantum Science and Technology (QST)

### Abstract

A GigE (Gigabit Ethernet) camera image data acquisition and analyzing system was developed using Aravis and macOS to create a customized beam-size-measurement program with least resources. Major elements of the system are GigE cameras, a computer running macOS, gnuplot software, Aravis and GLib libraries. Because Aravis and GLib enabled us to communicate with camera and to obtain image data from camera, we only created image data processing, analyzing, displaying and saving programs. Executable programs were written in C language and compiled, linked in macOS. The programs were initially developed by macOS 10.14 (Mojave) on Intel-based Mac in 2019, but now in 2023 they are also used by macOS 13 (Ventura) on Arm-based Mac.

### 1. はじめに

SPring-8 サイト内、兵庫県立大学高度産業科学技術研究所 NewSUBARU 蓄積リングへ電子ビームを供給する 1GeV 線型加速器が 2020 年度に完成し [1]、それ以降、NewSUBARU 蓄積リングの入射器として運転されている。また、東北大学キャンパス内に建設中の NanoTerasu 蓄積リングへ電子ビームを供給する 3GeV 線型加速器のコミッションングも順調に進んでいる [2,3]。

双方の線型加速器に使用される電子銃は規格化エミッタンスが 10mm-mrad 以下の新型低エミッタンス電子銃である [4]。この新型電子銃の性能評価を行うため、2019 年度に SPring-8 サイト内マシン実験棟と呼ばれる建屋に於いて、NewSUBARU 線型加速器用の電子銃、238MHz 空洞及び 476MHz 空洞を含む電子入射部が一時的に設置され、ビーム試験が行われた。エミッタンス測定にはスクリーンモニタ (SCM) を使用した Q スキャン法を採用した為、カメラからの画像データを取り込んで画像解析する必要があった。

マシン実験棟は SPring-8 加速器コンポーネント制御用の制御 LAN から切り離された、独立 (孤立) した建屋である。SPring-8 では 2018 年頃より Linux 上のオープンソースライブラリである Aravis [5] を使用して、様々なカメラベンダーに対応した Gigabit Ethernet (GigE) カメラ制御システムを構築している [6-9]。しかしながら、当該制御システムは SPring-8 のデータベース (DB) をベースとしたデータ収集系と密に連携しているため、一時的且つ独立した実験環境で使用しようとした場合、ハードウェア準備やセッティングの為にリソースが過大となる。それ故、マシン実験棟に制御 LAN を引き込み、SPring-8DB へ画像データを取り込むと云う確立された既存カメラ制御システムを採用する事は困難と判断し断念した。

そこで、実機で使用する GigE カメラをパーソナルコ

ンピュータ (PC) に直接繋いで、画像データを取り込む手法を計画した。GigE カメラ画像データを直接 PC へ取り込む場合、カメラベンダーが配布するソフトウェアを使用するのが簡易であり、ビューアとして画像確認のみ行うか、API を使用してデータ転送後に PC で画像データ解析を行う事となる。カメラベンダーから配布されるソフトウェアを使用する問題点は、使用可能なカメラが限定されたり、PC のオペレーティングシステム (OS) が限定 (バージョンを含む) されたりする為、画像システム構築の負担が大きな割りに、他への移植性が悪い事等である。

一方で macOS は UNIX 系の OS であり、Aravis も簡易に導入出来る為、macOS 上の Aravis を活用し、リソース負担を最小にした、一時的・独立した実験環境で使用可能な、且つ移植性の高いスクリーンモニタ用の画像データ取得・解析システムを構築した。

### 2. システム構成

#### 2.1 ハードウェア

##### 2.1.1 パーソナルコンピュータ

使用する PC は macOS が動作する Apple 製の Mac に限られる。CPU は Intel 製若しくは ARM アーキテクチャのものとなる。今回開発したソフトウェアは並行するスレッドをいくつも立ち上げる事が可能なため、CPU やメモリ等のリソースが許す限り、カメラの台数を増やす事が可能である。

複数の GigE カメラからのデータ取得にはカメラ台数分の Ethernet 入出力ポートが必要となる。その場合、PC 標準で備わっている RJ45 コネクタ以外のポートを使用する事となる。具体的には、USB3 若しくは Thunderbolt のポートをアダプタを介して Ethernet 入出力ポートとして使用する事となる。Figure 1 は PC に 4 つの Ethernet 入出力ポートを設け、4 台のカメラから画像データを取得する際の接続構成例である。

IP アドレスはプライベートアドレスの

\* ken@spring8.or.jp

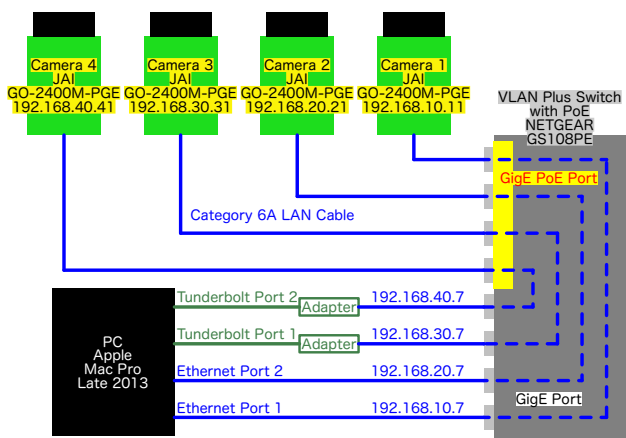


Figure 1: Hardware configuration of a GigE camera image data acquisition and analyzing system which uses four cameras.

192.168.\*\*\*.xxx を用い、設定は macOS システム (環境) 設定のネットワークパネルから手入力で行う (Fig. 2 参照)。注意として PC 側各ポートの IP アドレスは第 1 オクテット及び第 2 オクテットを 192.168 と共通とし、第 3 オクテットを例えば 10, 20, 30 or 40 と云う様に異なる数字にして、サブネットマスクを 255.255.255.0 とする事でポート毎にサブネットを分け、特定のポートと特定のカメラを 1 対 1 で通信させる。



Figure 2: Setting IP addresses with Network panel in System Preferences of macOS.

### 2.1.2 GigE カメラ

GigE カメラはインターフェースとして、Ethernet 入出力ポートを有している必要があるが、Aravis を使うためには GenICam (Generic Interface for Camera) 規格 [10] に準拠している必要がある。今回使用したのは SPring-8、SACLA、NewSUBARU 及び NanoTerasu で使用されている GO-2400M-PGE (JAI 製) である。他のカメラを使用する場合、GenICam 規格に準拠している GigE カメラであれば関数およびコマンドは本プログラムと共通となるが、ピクセル数設定 (GO-2400M-PGE は 1936×1216 ピクセル) を変更する必要がある。

GO-2400M-PGE は入出力ポートとして Power over Ethernet (PoE) 用のポート (RJ45 コネクタ) 及び電源・外部トリガ供給用のポート (丸形コネクタ) を有し

ている。電源はどちらのポートから供給しても良いが、本システムでは SPring-8 と同様に PoE 給電方式を採用した。

### 2.1.3 PoE インジェクターとしてのネットワークスイッチ

カメラ電源として PoE 給電方式を採用した場合、PC (Mac) には PoE 給電対応の Ethernet ポートが存在せず、また一部機種を除き PCIe の Ethernet ボードも実装できない為、PC とカメラの間に PoE インジェクターを設ける必要がある。本システムでは PoE インジェクターとしてネットワークスイッチの GS108PE (NETGEAR) を採用した。

GS108PE は合計 8 ポートを有しており、第 1 ポートから第 4 ポートまでが PoE に対応し、第 5 ポートから第 8 ポートまでが PoE 非対応である。さらに、GS108PE はポートベースの Virtual LAN (VLAN) が付加されたレイヤー 2 スイッチであるため、Fig. 1 の様にポート毎の仮想的 LAN セグメントを 4 つ作り、カメラと PC のポートを 1 対 1 に対応させた。

### 2.2 PC にインストールしたライブラリ・ソフトウェア

細々としたツールや設定に関する記述は省くが、本システムの動作に必要なライブラリ及びソフトウェアに就いて述べる。

Table 1 は現状で動作を確認した PC、ライブラリ及びソフトウェアのバージョンである。Aravis、GLib [11] 及び gnuplot [12] の PC へのインストールはパッケージマネージャー Homebrew [13] を使用した。Aravis 及び GLib のライブラリに関して、Intel 製と ARM アーキテクチャ Mac では Homebrew によるライブラリの格納先が異なる為、インストール・リンク・実行時に正しいパスを指定する必要がある。

Table 1: Versions of Softwares

PC	macOS	GLib	Aravis	gnuplot
Mac Pro 2019 (Intel)	13.5	2.76.4	0.8.27	5.4.8
Mac mini 2023 (Arm)	13.5	2.76.4	0.8.27	5.4.8

システムの動作に関して、セキュリティソフトウェアその他のプロセスが正常動作を妨げる可能性があるため、上記のインストールに関して、一旦 macOS を PC にクリーンインストールした後、これらのライブラリ・ソフトウェアをインストールすることを勧める。

### 2.2.1 OS

PC の OS として macOS を採用した。本システム開発当初の OS は macOS 10.14 (Mojave) であったが、現在は最新の macOS 13 (Ventura) でも動作している。本システムのプログラムは C 言語で書かれ、macOS 標準のコンパイラでコンパイル及びリンクが行われ、実行形式ファイルが作成される。

動作に必要なライブラリは全て動的ライブラリとなっている。すなわち、コンパイル・リンク時に参照先フォルダ等の情報を保管し、実行時に結合するするタイプである。従って、使用する PC に全てのライブラリ

等をインストールした後、コンパイル・リンクを行い、本プログラムを実行すると云った手続きを踏む。

### 2.2.2 Aravis

GLib を介してカメラとの通信を行うライブラリである。開発当初 Aravis のバージョンは 0.6 であったが、現在は 0.8 である。バージョン 0.8 への変更で使用する関数の引数が変化したため、作成したプログラムの改修（引数の変更）を一度行った。

### 2.2.3 GLib

PC の Ethernet ポートをハンドリングする GIO (Gnome Input/Output) 他を含むライブラリである。

### 2.2.4 gnuplot

取得されたカメラ画像データを PC ディスプレイへ表示させる為に使用する。gnuplot の起動及び操作は作成したプログラムの中でパイプを通し、gnuplot を呼び出して実行させる。gnuplot での出力形式 (Terminal Type) は Qt [14] とした。Qt のライブラリは gnuplot をインストールする際に一緒にインストールされる。Qt での表示は実行形式ファイルである gnuplot Qt が担う。

開発当初、処理速度の考慮して直接 Qt で表示させることを考えたが、gnuplot を試験し 1Hz 程度の画像更新は可能だと判明し、gnuplot を使用した画像データ表示を採用した。若し恒久的なシステムを構築するのであれば、SPRing-8 と同様 [8] に直接 Qt で表示させるのが良いと思われる。

## 2.3 開発したプログラム

作成されたプログラムはターミナルを開き、コマンドライン入力を行なって実行する。Table 2 及び Fig. 3 は作成したプログラム一覧及びプログラムと画像・制御データのやり取りを示すダイアグラムである。以下にそれらの機能に就いて述べる。

Table 2: Developed Programs and Their Functions

<b>disp</b>	Process, display and save image data
<b>stop</b>	Terminate <b>disp</b> program
<b>acqui</b>	Save raw image data
<b>gain</b>	Change gain value in camera
<b>expo</b>	Change exposure time value in camera
<b>init</b>	Initialize parameters in camera
<b>internal_trig</b>	Make trigger mode internal in camera
<b>external_trig</b>	Make trigger mode external in camera
<b>prjh</b>	Display horizontally projected data
<b>prjv</b>	Display vertically projected data
<b>prjh_stop</b>	Terminate <b>prjh</b> program
<b>prjv_stop</b>	Terminate <b>prjv</b> program

### 2.3.1 disp

**disp** は開発したプログラムの本体部分で、カメラから取得したデータを gnuplot が読み出せるような形式へ変換し（含む縮小）、表示 (Display) させたり、残すべきデータをディスク (HDD, SSD または Ramdisk) へ保存するプログラムである。

**disp** はパイプライン経由で gnuplot を起動し、gnuplot へデータを渡して表示させる。データを渡す方法は 2 種類あり、gnuplot のインラインデータ入力機能を使用

してプログラム中で文字列を 1 行ずつ直接 gnuplot へ渡すか、一時的にディスクへ保存して gnuplot にディスクから読み込ませるかのどちらかである。現状では Ramdisk を介してデータを渡す方法が一番速い。

gnuplot での表示画素サイズは Full (1936×1216 ピクセル)、Half (968×608 ピクセル) 及び Quarter (484×304 ピクセル) とした。PC の処理能力によって表示画素数を変える。

**disp** は Aravis の Web サイトに掲載されているサンプルコード [15] をベースに、画像表示に必要なプロセスを付加して作成された。**disp** の画像データ取得スキームは GigE Vision Streaming Protocol (GVSP) [16] に依っている。画像データはトリガに同期してカメラで取得され、PC 側のリングバッファへプッシュされる。我々が行うべき処理（書くべきコード）はリングバッファから画像データを取り出し、取得したデータを加工、表示、解析、保存する事である。

**disp** プログラム動作中、PC とカメラ間の通信はストリーミングプロセスのみとなる。**disp** 動作中（画像データ表示中）にゲイン（感度）や露光時間の変更できないと不便なため、この二つのカメラパラメータに限って **disp** 動作中でも変更可能な仕様とした。具体的にはディスクにパラメータファイル (parameters.txt) を置き、**disp** の周期タスク毎にパラメータファイル中の数値を読み出す事とした。

パラメータファイル 1 及び 2 番目数値は **disp** の停止及びデータ保存に関するパラメータである。1 番目数値が 0 → 1 と変化した場合に **disp** を終了する手続きを進め、2 番目数値が 0 → 1 と変化した場合に画像データ保存を行う。画像データ保存後に **disp** は 2 番目数値を元に戻す (1 → 0) 処理を行う。

パラメータファイル 3 及び 4 番目数値はカメラゲイン及び露光時間に関するパラメータである。3 及び 4 番目数値が **disp** が保持している数値と異なった場合に、パラメータファイル数値をカメラへ転送する処理を行う。尚、カメラ設定値の変更は GVSP とは異なる GigE Vision Control Protocol (GVCP) [16] を通じて行われる。

さらに **disp** は取得した画像データや加工したデータをディスクへ保存する機能を持つ。**disp** は ROI (Region of Interest) が指定する二次元範囲内の発光強度を画面の水平軸方向と垂直軸方向に積算した一次元投影ファイルを作成し、ディスクへ保存する。投影形状のグラフ表示や重心位置、重心位置からの標準偏差計算は **prjh** 及び **prjv** が担う。

### 2.3.2 stop, acqui

**stop** は **disp** を停止 (Stop) させるために作成された。**disp** 立ち上げ時、パラメータファイル 1 番目数値は 0 (=周期タスク) に初期化される。**stop** は **disp** を終了したい時にパラメータファイル 1 番目数値を 1 (=停止) に書き換えるプログラムである。

**acqui** はストリーミングで得られた画像データを取得 (Acquisition) し、ディスクへ保存するために作成された。**acqui** は画像データをディスクへ保存したい時にパラメータファイル 2 番目数値を 1 (=保存実行) に書き換えるプログラムである。

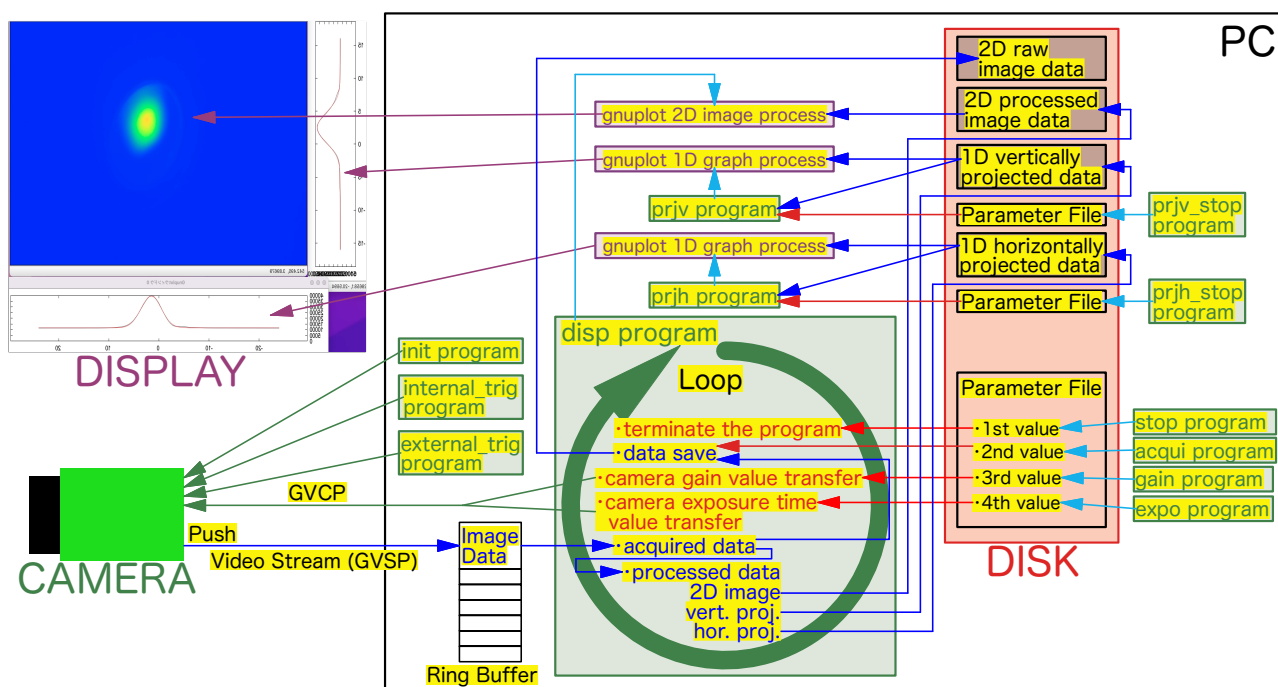


Figure 3: Software diagram.

### 2.3.3 gain %, expo %d

**gain** はカメラのゲイン (Gain) を変更する為に作成された。引数としてカメラが指定するゲイン値が整数値で入力される。GO-2400M-PGE の場合、1~200 の値が設定可能である。**gain** はパラメータファイル3番目数値を希望するゲイン値に書き換えるプログラムである。

**expo** はカメラの露光時間 (Exposure Time) を変更する為に作成された。引数としてカメラが指定する露光時間値が整数値で入力される。GO-2400M-PGE の場合、露光時間の単位は  $1\mu s$  で  $38 \sim 8,000,000$  ( $38\mu s \sim 8s$ ) の値が設定可能である。**expo** はパラメータファイル4番目数値を希望する露光時間値に書き換えるプログラムである。

### 2.3.4 init, internal\_trig, external\_trig

**init** はカメラを初期化する (Initialize) 為に作成された。GO-2400M-PGE の場合、停止状態から電源が入ると、約 50Hz の内部トリガ状態となる。その状態で **disp** を動作させるとプログラムがビジーとなり制御不能になるため、それを回避するための初期化を行うプログラムである。

**internal\_trig** は外部トリガモードから内部トリガ (Internal Trigger) モードへ切り替える際に使用するプログラムである。同時に内部トリガのトリガレートをカメラへ転送する。

**external\_trig** は内部トリガモードから外部トリガ (External Trigger) モードへ切り替える際に使用するプログラムである。

これら3つのプログラムはカメラと直接通信するものであり、**disp** が動作していない、すなわち通信が無負荷の状態で使用される。

### 2.3.5 prjh, prjv

**prjh** は表示画面の水平軸、**prjv** は垂直軸へ投影 (Projection) された一次元輝度分布をディスク中のファイルから読み出して表示し、重心位置及び重心位置からの標準偏差を計算する為のプログラムである。

### 2.3.6 prjh\_stop, prjv\_stop

**prjh\_stop** 及び **prjv\_stop** は **stop** と同様な手法で **prjh** 及び **prjv** を停止させるプログラムである。

## 3. システムの動作

Figure 4 は PC 画面のスクリーンショットである。右上のターミナル (緑) で **disp** プログラム、右下のターミナル (ベージュ) で **prjv** プログラム、左下のターミナル (黒) で **prjh** プログラムを起動している。右上側にデータ受け渡し用の Ramdisk が生成されている。

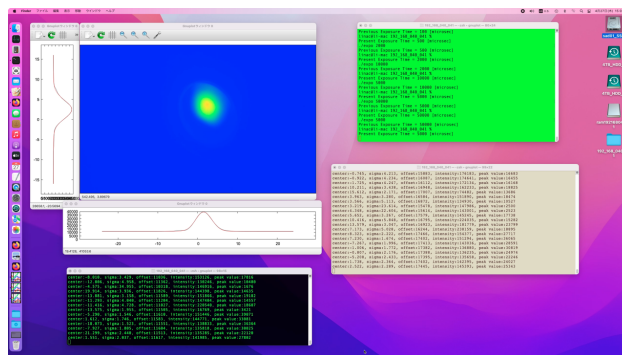


Figure 4: A screenshot of PC display.

**prjh** 及び **prjv** は輝度分布データをディスクから読み

出し、重心位置及び重心位置からの標準偏差等をフィッティング（最小二乗法）で計算する。計算された数値データは各ターミナルにてテキスト出力される。二次元画像データ表示の左側及び下側に gnuplot のグラフ表示があるが、黒線で描かれたグラフはディスクから読みこまれた輝度分布データを示しており、赤線で描かれたグラフはフィッティングの結果を表示したものである。

Figure 5 は 1 台の PC で 4 台のカメラから画像データを取り込む試験を行った際の写真である。右上は 4 つの Ethernet 入出力ポートを取り付けられた PC (Mac Pro Late 2013)、左上はテーブルに置かれた 4 台の GigE カメラ (GO-2400M-PGE)、左下は PoE インジェクタとして使用されるネットワークスイッチ (GS108PE)、右下が 4 つの画像データ表示中の PC 画面である。10 年前の PC (4CPU・8 スレッド) ではあるが、4 つの Half (968×608 ピクセル) 画像データ表示を行っても 1Hz 程度の画像更新は問題なく行えている（但し、フィッティング等のプロセスは立ち上げてない）。

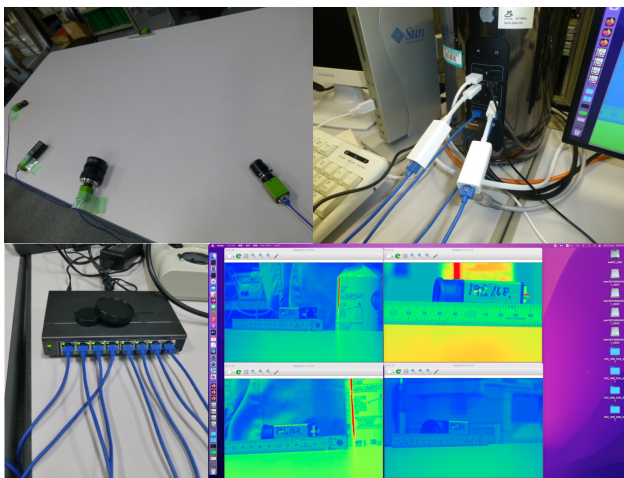


Figure 5: Examination of image data acquisition from four cameras.

また、ARM アーキテクチャ CPU を搭載する PC (Mac mini 2023) に於いても試験を行い、問題無く動作することが確認された。

#### 4. まとめ

GigE カメラから画像データを取り込んで、解析するシステムを macOS 上で構築した。Aravis は GigE カメラから画像データ取得を行う関数を提供するのみであり、PC 画面表示やデータ解析その他の処理は各自がプログラムを作成して行う事となる。

Apple 製 PC 及び macOS は毎年アップデートされ、システム・ライブラリ構成が大きく変化する場合がある。そのような場合でもタイムラグがあるが、最新の PC 及び macOS に対応した新バージョンのソフトウェア (Aravis 等) がリリースされるので、その部分 (新バージョン等) の修正を適宜行えば持続的に本システムを使い続けられる。

#### 参考文献

- [1] T. Inagaki *et al.*, “Commissioning of The New 1.0 GeV Electron Linac For NewSUBARU Storage Ring”, Proc. of the 18th Annual Meeting of Particle Accel. Soc. of Japan, Online, Aug. 2021, pp. 102-105. [https://www.pasj.jp/web\\_publish/pasj2021/proceedings/PDF/TUOB/TUOB03.pdf](https://www.pasj.jp/web_publish/pasj2021/proceedings/PDF/TUOB/TUOB03.pdf)
- [2] T. Asaka *et al.*, “Installation and Beam Commissioning of 3 GeV Linear Accelerator in NanoTerasu”, Proc. of the 20th Annual Meeting of Particle Accel. Soc. of Japan, Funabashi, Aug. 2023, WEOA7, this meeting.
- [3] T. Asaka *et al.*, “Beam Performance of Injector System in NanoTerasu”, Proc. of the 20th Annual Meeting of Particle Accel. Soc. of Japan, Funabashi, Aug. 2023, THP49, this meeting.
- [4] T. Asaka *et al.*, “Low-Emittance Radio-Frequency Electron Gun Using a Gridded Thermionic Cathode”, Phys. Rev. Accel. Beams **23**, 063401, June 2020. <https://journals.aps.org/prab/abstract/10.1103/PhysRevAccelBeams.23.063401>
- [5] Aravis - A Vision Library for GenICam Based Cameras, <https://github.com/AravisProject/aravis>
- [6] A. Kiyomichi and T. Fukui, “Development of GigE Vision Camera Control for Upgrading Screen Monitor of SPring-8/SACLA”, Proc. of the 16th Annual Meeting of Particle Accel. Soc. of Japan, Kyoto, July 2019, pp. 873-876. [https://www.pasj.jp/web\\_publish/pasj2019/proceedings/PDF/FRPH/FRPH005.pdf](https://www.pasj.jp/web_publish/pasj2019/proceedings/PDF/FRPH/FRPH005.pdf)
- [7] A. Kiyomichi *et al.*, “Application of GigE Vision Camera to New Screen Monitor Control System for Beam Transport from SACLA to SPring-8”, Proc. of the 17th Annual Meeting of Particle Accel. Soc. of Japan, Online, Sept. 2020, pp. 731-734. [https://www.pasj.jp/web\\_publish/pasj2020/proceedings/PDF/FRPP/FRPP24.pdf](https://www.pasj.jp/web_publish/pasj2020/proceedings/PDF/FRPP/FRPP24.pdf)
- [8] A. Kiyomichi *et al.*, “Application of GigE Vision Camera Control System for Screen Monitors of SACLA-BL1 (SCSS+) and NewSUBARU Injector Linac”, Proc. of the 19th Annual Meeting of Particle Accel. Soc. of Japan, Online, Oct. 2022, pp. 535-539. [https://www.pasj.jp/web\\_publish/pasj2022/proceedings/PDF/WEP0/WEP012.pdf](https://www.pasj.jp/web_publish/pasj2022/proceedings/PDF/WEP0/WEP012.pdf)
- [9] A. Kiyomichi *et al.*, “Upgrade of GigE Vision Camera Control System and Applied to Screen Monitors of SACLA and NanoTerasu”, Proc. of the 20th Annual Meeting of Particle Accel. Soc. of Japan, Funabashi, Aug. 2023, WEP09, this meeting.
- [10] The Generic Interface for Cameras standard, <https://www.emva.org/standards-technology/genicam>
- [11] GLib Is a General-Purpose, Portable Utility Library, <https://docs.gtk.org/glib>
- [12] Gnuplot Is a Portable Command-Line Driven Graphing Utility for Linux, OS/2, MS Windows, OSX, VMS, and Many Other Platforms, <http://www.gnuplot.info>
- [13] The Missing Package Manager for macOS (or Linux), <https://brew.sh/index>
- [14] The Future of Digital Experiences, <https://www.qt.io/>
- [15] Aravis/arvexample.c, <https://github.com/AravisProject/aravis/blob/main/tests/arvexample.c>
- [16] Ethernet Protocol (GV), <https://aravisproject.github.io/docs/aravis-0.8/aravis-gv.html>