ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-TUCPR06

FAST INTERACTIVE PYTHON-BASED ANALYSIS OF STREAMED IMAGES*

A. Sukhanov†, W. Fu, J. P. Jamilkowski, R. H. Olsen
Brookhaven National Laboratory, Upton, USA

Abstract

This paper reports on development of a general purpose image analysis application, tailored for beam profile monitor cameras of RHIC Collider-Accelerator complex. ImageViewer is a pure Python application, based on pyqtgraph and scipy packages. It accepts image streams from camera servers (RHIC or EPICS), local cameras or from file system. The standard analysis includes recognition of connected objects; for each object the parameters of a fitted ellipsoid (position, axes and tilt angle) are calculated using 2nd-order image moments, the parameters are corrected using 1D or 2D gaussian fit. Many other features are supported: saving, image rotation, region of interest, projections, subtraction of a reference image, multi-frame averaging, color mapping, contrast control, pixel to millimeter calibration. Playback feature allows for fast browsing and clean-up of saved images. User add-ons can be added dynamically as included modules. The existing add-ons provide turn-key solutions for moving-slit and multi-slit beam emittance measurements, hollow-beam characterization and others.

Each camera of the RHIC complex is equipped with a server (graphic-less) version of this application, providing the same analysis and publishing calculated parameters to RHIC Controls Architecture.

INTRODUCTION

The imageViewer is a general purpose image analysis application, written in pure Python, it is based on pyqtgraph [1] and scipy [2] packages. The pyqtgraph is a fast Python-based visualization package that makes use of Qt GraphicsView for primitive graphics (lines, shapes, antialiasing) and is capable of rendering 1024x1024 video at 50 fps.

Everything is controlled from a single window as shown on Fig. 1, the window is divided into adjustable panes:

- Image pane,
- Control pane with parameter tree of GUI elements (buttons, checkboxes, spinboxes, sliders, etc.),
- Message pane,
- Console pane, a Python interactive console where users can enter python expressions/statements and access local objects for debugging.

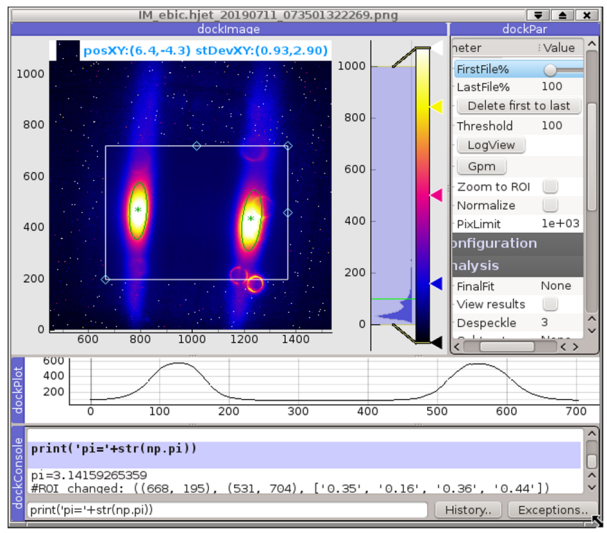

Figure 1: imageViewer window.

FEATURES

Image Delivery

The imageViewer is an event-driven application, the input back-end raises an event for the main thread to indicate arrival of new image.

The back-ends for following image sources are integrated:

- Camera server (RHIC Controls). The image is either PNG-compressed (default) or byte array.
- EPICS Area Detector, image is byte array.
- File system.
- Local USB camera.
- HTTP.

The default file format is PNG; it handles color depths higher than 8 bits. Other formats are supported as well.

Asynchronous delivery of large amounts of data from camera servers may cause buffer overloading of TCP stacks on the server or client side (when image size is large and client is slow to receive or process data). To throttle the input stream, the imageViewer asynchronously requests a small scalar parameter which is updated when a new image is ready, if imageViewer is busy processing previous images, then the request is dropped, otherwise the image array is received synchronously.

Rotation and Orientation

Rotation to an arbitrary angle and flipping can be applied to all images in the stream or interactively.

* Work supported by Brookhaven Science Associates, LLC under Contract No. DE-SC0012704 with the U.S. Department of Energy.
† sukhanov@bnl.gov

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-TUCPR06

Region Of Interest (ROI) and Partitioning

Image analysis is performed in an interactively-defined ROI. The ROI can be partitioned vertically or horizontally and each partition is analyzed separately.

ROI Projection Plots, ROI Intensity Plot

Projections of the ROI and pixel amplitudes of the ROI are plotted in separate windows and dynamically updated.

Isocurves

The isocurve lines visualize the threshold level for spot finding.

De-speckling and Blurring

The narrow median filter can be applied to remove speckles effectively and quickly. As an alternative, gaussian blurring can be performed.

Running Average

Pixel-wise running average for several frames can be applied to the input stream.

Reference Images and Pedestal Suppression

Displayed image can be stored in one of four reference slots and a pixel-wise arithmetic operation can be performed between input images and references.

Color Depth

More than 8 bits per channel is supported. The integrated PNG codec (from Pillow [3]) is fast and it supports an arbitrary number of bits per channel.

Contrast/Coloration Interactive Control

Pyqtgraph provides useful tools to view high-color-depth images:

- pixel intensity histogram,
- movable region over histogram to select black/white levels,
- color map selector and gradient editor for gray-scale images.

Interactive Calibration

Pixel/mm calibration is done by interactively adjusting a reference ring to a known shape.

Fast Browsing and Clean-up of Image Folders

The folders of stored images may fill up with hundreds of thousands of files, they can be browsed, viewed and deleted quickly (movie-style viewing frequency is 30-50 frames/s). Deletion of large sets of files can be done in several seconds.

IMAGE ANALYSIS

The standard analysis is illustrated on Fig. 2 it includes:

- Conversion of color (RGBA) array to grayscale.
- Connected objects are located and measured using Computer Vision algorithms (thresholding and labelling) from scipy [4].
- Parameters of fitted ellipses for found objects are calculated using Image Moment Statistics:
  - Center Of Gravity: $x = M_{10}/M_{00}$, $y = M_{01}/M_{00}$,
  - Half widths (RMS): M_{20}, M_{02},
  - Pearson Correlation Coefficient: $$\rho = \frac{M_{11} - M_{10}M_{01}}{\sqrt{M_{20} - M_{10}^2}\sqrt{M_{02} - M_{01}^2}}$$
  - Tilt angle of the ellipse: $$\theta = atan(\frac{2\rho \cdot M_{20} \cdot M_{02}}{M_{20}^2 - M_{02}^2})$$

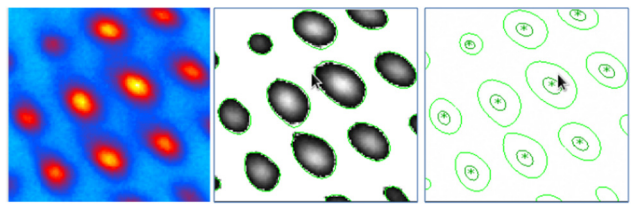

Figure 2: Image analysis stages: original image (left); gray-scaled, thresholded and labelled (middle); centroids (stars) and fitted ellipses (dark green) plotted (right).

The standard analysis provides fast and robust estimation of the object position, the speckles in the image do not bias the calculation since they are disconnected from the objects. However, the calculated width of the object may be biased if background level is significant. In that case, the curve fitting can be applied to ROI projection using gaussian with a pedestal. The fitting range is extended 6 times the RMS to provide necessary information about the background. Precision of the width measurement could be further improved using a full two-dimensional fit of a 2D gaussian.

The calculated position and width of the brightest objects are displayed at the top of the image, all parameters also available in a JSON-formatted log file.

ADD-ONS

An add-on is a python file, which is specified in a command line argument and is imported during initialization. An add-on can instantiate (import) other add-ons dynamically. Add-on adds additional GUI elements into the control pane. It should be sub-classed from an AddonBase object:

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-TUCPR06


```
class AddonBase():
  def __init__(self,imager):
    #Constructor, inherits the Imager object
    self.imager = imager

  def cprint(self,msg):
    #Thread-safe printing in the console dock.

  def entryName(self):
    #Returns group name in the parameter tree

  def controlPane(self):
    #Define user items in the control pane.

  def addon_clicked(self,parItem,itemData):
    #Called when user item in control pane is
     clicked

  def process(self):
    #Called when data region is updated.

  def stop(self):
    #Called when imager is stopped.

  def exit(self):
    #Called when application exits.
```

Several examples of add-ons are shown below.

RFDiag Add-on

The LEReC RF diagnostic line is designed for fine tuning of the RF required to produce 2 MeV electron bunches with an energy spread better than $5*10^{-4}$. The images are taken from a YAG screen. The ROI is sliced into 32 strips. The strip position corresponds to the time inside a bunch and horizontal RMS – to momentum spread. Results of the analysis is shown on Fig. 3.

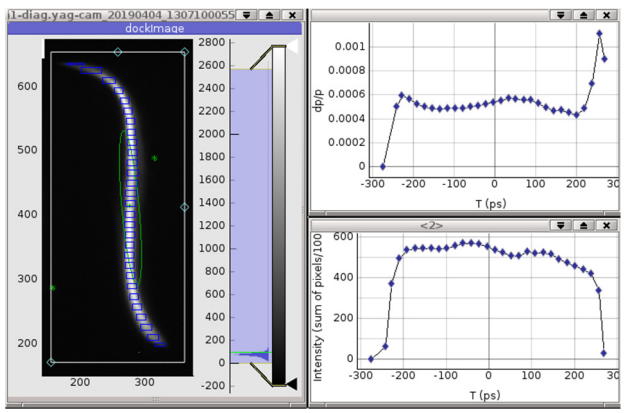

Figure 3: Image from a YAG screen (left). Momentum spread (top right), bunch intensity over time (bottom right).

Multi-slit and Moving-slit Emittance Add-ons

Emittance measurement devices made of slit/slits are used to measure beam emittance along the beamline [5]. The multi-slit add-on calculates emittance for multi-slit device online, as shown on Fig. 4 and Fig. 5. For moving-slit measurements it is important to synchronize slit movement with camera shutter. Two add-ons are involved in that case, one add-on is controlling the slit motion online, the second add-on is processing the accumulated images offline. The results of the multi-slit emittance measurement are shown on Fig. 6.

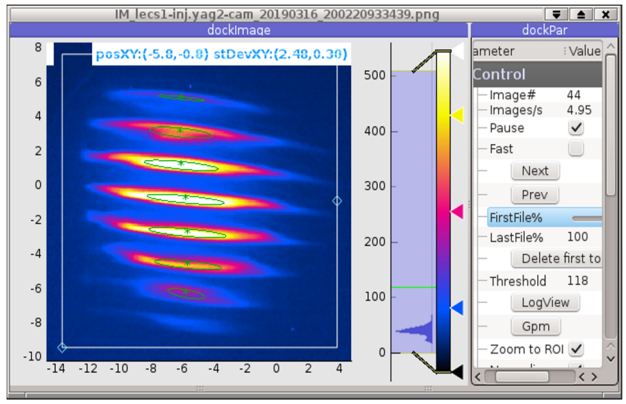

Figure 4: Image from a YAG screen 2m downstream of a slit mask.

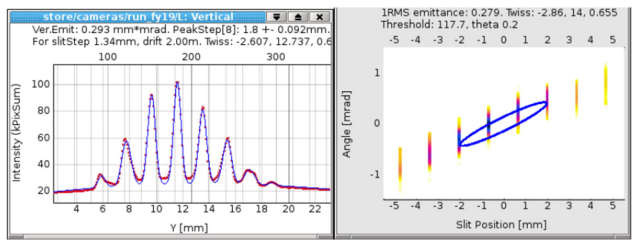

Figure 5: Beamlets fitted with eight gaussians (left), Twiss parameters and phase-space graph (right).

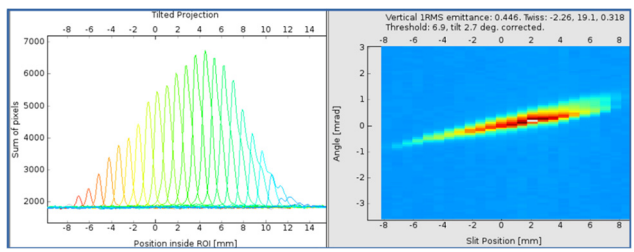

Figure 6: Beam profiles at different slit positions (left). Reconstructed phase space (right).

LOGGING OF CALCULATED PARAMETERS

Currently about 30 cameras are instrumented in the RHIC complex. Each of these cameras is served by a corresponding imageMan program, which is a graphic-less version of the imageViewer. It continuously publishes results of the image analysis and calculated beam-related parameters to the RHIC logging system. The interactively adjusted parameters of an imageViewer can be synchronized with the corresponding imageMan.

CONCLUSION

Pure Python, general purpose image analysis application is used in all camera installations at the RHIC complex. It provides fast analysis of multiple image objects (the typical standard analysis time of a 2.4 mega-pixel 12-bit image on Fig. 1 is 11 ms, de-speckling time of the ROI is 70 ms).

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-TUCPR06

Additional features can be added as user-supplied importable modules.

REFERENCES

[1] http://www.pyqtgraph.org

[2] https://www.scipy.org

[3] https://python-pillow.org/

[4] https://docs.scipy.org/doc/scipy/reference

[5] C. Liu *et al.*, "Transverse Beam Emittance Measurements with Multi-Slit and Moving-Slit Devices for LEReC", in *Proc. 7th Int. Beam Instrumentation Conf. (IBIC'18)*, Shanghai, China, Sep. 2018, pp. 486-489, doi:10.18429/JACoW-IBIC2018-WEPB21