Pylinac: Image analysis for routine quality assurance in radiotherapy

James R. Kerns

1 Radformation, New York, NY, United States of America ¶ Corresponding author

Summary

Pylinac is a Python library that analyzes routine quality assurance images generated by devices in the radiotherapy treatment domain. It contains multiple independent modules that map to different quality assurance tasks and will ingest and analyze the images for common metrics as required by the medical physics society. The library is designed to be concise and easy to use as the target audience is not developers. Thus, most workflows can be implemented in a few lines of code. At the same time, the library is modular and allows for easy extension via plugins and configuration settings. Comprehensive documentation is available with usage examples and algorithm explanations.

Statement of need

Within the therapeutic medical physics domain, verification that radiotherapy machines are performing in accordance with specification is an obvious need considering their use on humans and is required by relevant societies (Dieterich & Pawlicki, 2008; Klein et al., 2009; Kutcher et al., 1994). This involves routine quality assurance (QA) at regular intervals by medical physicists. A subset of this QA involves acquisition and analysis of images generated by the radiotherapy devices. This includes the mechanical size of the “isocenter” of the linear accelerator (Winston & Lutz, 1988), dosimetric performance of the accelerator as it rotates around the patient (Ling et al., 2008), and examination of the individual “leaves” of the multileaf collimator that shape the radiation (Calvo-Ortega et al., 2014). These images and data test the various mechanical and dosimetric performance dimensions of the machine. The images are usually the same pattern at every interval and are used for constancy testing. Manual examination of images is subject to interpersonal interpretation (Ho et al., 1995; Kerns & Anand, 2013). Performing this quality assurance has been examined as being quantifiable by image or digital analysis in the past (Depuydt et al., 2012; Du & Yang, 2009; Eckhause et al., 2015; Jørgensen et al., 2011; Kerns et al., 2014; Rowshanfarzad et al., 2011). Commercial applications exist but can be prohibitively expensive and at the time the library was written no open-source alternatives existed. Medical physicists usually do not have computer science training and creating their own in-house software for such evaluation can be difficult to justify. There is thus a need for software for budget-constrained radiotherapy clinics as well as an open standard for analysis of these data instead of proprietary programs made by individual authors and clinics.

Example usage

Although pylinac contains multiple independent modules focused on analyzing different images, this example will focus on one: planar image analysis for image metrics of a radiation source and camera combination. Linear accelerators have a built-in scintillation camera that can...
record and visualize radiation. As part of the routine quality assurance, the performance of the camera is measured monthly and annually. This is performed with a device that can measure contrast and spatial resolution or more, also known as a "phantom". A device is placed in the path of the beam and the image is captured on the scintillating camera (Figure 1).

![Figure 1: A DICOM image with a phantom in place](image)

The image can be exported in the format of the Digital Imaging and Communications in Medicine (DICOM). This DICOM image can be passed to pylinac and analyzed. The only input required is the image and the type of phantom being analyzed.

```python
from pylinac import LeedsTOR
dicom_path = r"path\to\dicom.dcm"
leeds = LeedsTOR(dicom_path)
leeds.analyze()
leeds.plot_analyzed_image()
```

Pylinac will localize the phantom within the image, meaning the user’s placement of the phantom is not a variable. Rotation can be corrected within a certain range, usually within 5 degrees for most phantoms. This also removes the placement technique of the user as a result variable. After localization and rotational correction, each region of interest (ROI) is then sampled, of which there can be several. Each phantom’s ROIs are known ahead of time so simple offsets based on the phantom center and angle can be utilized. After the ROIs are sampled (Figure 2) the metrics can be computed (Figure 3).
The most common metrics are contrast and spatial resolution. Contrast can be defined many ways, but the default one in pylinac is:

\[ \frac{I_{\text{mean}} - R_{\text{mean}}}{I_{\text{mean}} + R_{\text{mean}}} \]

where \( I \) is the ROI of the contrast region in question and \( R \) is the background ROI, usually placed somewhere within the phantom area that is uniform.

This corresponds to the circular ROIs at the outer edge of the phantom (Figure 2). The contrast is calculated for each ROI and can then be plotted as a curve. Spatial resolution is defined as

\[ \frac{I_{\text{max}} - I_{\text{min}}}{\max \left( \frac{I_{\text{max}} - I_{\text{min}}}{I_{\text{max}} + I_{\text{min}}} \right)} \]

where \( I = 1...n \) line pair ROIs. This is also called the modulation transfer function (MTF) (Schroeder, 1981). The ROIs at the center of the phantom with the quickly-alternating lines define the spatial resolution. For each ROI, the spacing of a high-density and low-density material is fixed. The spatial resolution of each ROI is calculated and can be plotted as a curve. Typically, the medical physicist is looking at the resolution value at the 50% line of the curve (Figure 3).
After analysis, these values are used by the medical physicist to compare to previous values or expected values. From there, calibration of the camera may be necessary. The results can also be saved as records that may be audited by government authorities.

Adoption and impact

Pylinac has been used widely in literature since its release in 2014, either validation of the algorithms for use by individual clinics (Boudet et al., 2022; Bredikin & Walsh, 2022; Ji & Cong, 2022; Lay, Chuang, Wu, et al., 2022), as a research tool for other ends (Alexander et al., 2021; Al-Kabkabi et al., 2022; Bozhikov et al., 2019; Cullom et al., 2021; Hu et al., 2022; Huang et al., 2021; Mendes et al., 2022; Pant et al., 2020; Pearson et al., 2022; Salari et al., 2023; Tegtmeier et al., 2022; Wang et al., 2020; Wojtasik et al., 2020), or used within other packages (Chuang et al., 2021; Lay, Chuang, Giles, et al., 2022; Oliver et al., 2022)

Author contribution statement

Conceptualization, coding, development, and paper writing by James Kerns.

Acknowledgements

We thank Radformation for continuing to support the open-source work of this project and Dr. Stephen Kry for providing protected time to initially develop the project. Several individuals have contributed to the project over the years, which can be viewed here: https://github.com/jrkerns/pylinac/graphs/contributors.

References


