International Photoacoustic Standardisation Consortium

Photoacoustic Data and Device Parameters

IPASC-DAM Consensus Document Version 2.0

Last changed: 16/09/2021
Contact persons: Janek Gröhl, Lina Hacker, Ben Cox

This document defines metadata for photoacoustic/optoacoustic (PA) devices to enable description, interoperability and data exchange of PA imaging data.

Table of Contents:

Definitions

Detection Element: A specific material element capable of converting mechanical energy to electrical energy and in some cases reciprocally converting electrical energy to mechanical energy (IEC TR 60854 1986)[2]. Here, it refers to e.g. a piezoelectric crystal or a laser-light interferometer.

Device: A specific make and model of a photoacoustic device (hardware and/or software) including devices which could be regulated in the United States of America (USA) by Food and Drug Administration (FDA) Center for Devices and Radiological Health by 510(k) clearance24 and in the European Union (EU) by CE marking[1]. A distinction between *full scan* (image acquisition without sequential movement of an illumination-/detection array or target) and *raster scan* devices (image acquisition by the sequential movement of an illumination-/detection array or target) is drawn.

Frame: A set of raw time series data corresponding to a single acquisition step of the imaging *device*. The *Frame Acquisition Spatial Positions* metadatum can be used to assign the relative spatial position of the frame. For full scan devices, one frame defines the full image, whilst for raster-scan devices, an image is composed of several frames.

Illumination Element: A specific material element capable of emitting light to illuminate the target (for example, an optical fibre that conveys light generated by a laser source or a light-emitting diode (LED)). The spatial positions of all illumination elements must be fixed throughout the acquisition of one frame.

Image: An array of values varying in two or more spatial dimensions derived from analysis of an imaging signal and corresponding to an array of spatial locations in the imaged object [1]. In the context of this document, a PA image refers to the result of mapping raw time series data into the spatial domain. An *image* can be composed of one or multiple *frames*.

Modality: A category of an imaging device, characterised by a distinct physical principle[1] (e.g. PA imaging).

Raw Time Series Data: A '*time series*' refers to the time-sampled signal from one detection element. '*Time series data*' refers to a set of such time series, one for each detection element. The descriptor 'raw' denotes that the time series data is unprocessed.

Further definitions can be found in the **[IPASC 'Terms and Definitions' consensus document]**.

Attributes

Each metadatum in this document is characterised by a series of attributes with the purpose of describing and defining its use and boundary conditions. If necessary, further specifications by nested attributes can be given.

Necessity: 'Minimal' or 'Report if present' condition for the metadatum. Minimal parameters are all parameters that are required to reconstruct an image from the raw time series data. Any additional information should be reported in the metadata if available.
dtype: Data type of the attribute
Units: SI units of the attribute if applicable
Description: A short description of the attribute
Method Name: The name of the function/method that can be called in any programming language to obtain information on a specific attribute.
Condition: Constraints of an attribute that limit its value range (e.g. the acquisition wavelengths must the of the same size as the acquired frames)
Nested Attribute: A sub-attribute that further describes an attribute
Measurement Device Attribute: A specific type of nested attribute that describes measurement device details if required. Measurement device attributes are always optional. They include:

> **Measurement Device Type:** A string literal describing the measurement device for this attribute, e.g. 'pyroelectric sensor' or 'wavemeter'.
> **Measurement Device Manufacturer:** A string literal describing the manufacturer of the measurement device, e.g. 'Thorlabs'.
> **Measurement Device Serial Number:** A string literal comprising the serial number of the measurement device.
> **Calibration Date:** A timestamp referring to the date when the measurement device was last calibrated. Timestamps are given in seconds with the elapsed time since epoch (Jan 1st 1970, 00:00).

Minimal parameters

The minimal parameters refer to the bare minimum set of parameters that are needed to read the data and reconstruct an image from PA raw time series data. Please note that the listing below is a summary only of metadata which have a “minimal” necessity. Full details on each of the attributes can be found in Part 1 and Part 2 of this document. The minimal parameter set contains:

- Container Format Metadata: The *container format metadata* refer to the inherent features of the file format which specify mandatory parameters. They include the UUID (1), the type of compression (2) and the type of encoding (3).

- Binary Data Metadata: The *binary data metadata* refer to the metadata that make the binary data machine-readable. They include specifications on data type (1), dimensionality (2) and the sizes (3) of each dimension.

- Sampling Rate: The *sampling rate* refers to the rate at which samples of the analog signal are taken to be converted into digital form.

- Acquisition Wavelengths: The *acquisition wavelengths* field is a one-dimensional (1D) array that contains all wavelengths used for image acquisition.

- Detector Positions: The *detector position* defines the position of the detection element in three-dimensional (3D) cartesian coordinates [x_1, x_2, x_3].

Part 1 - Photoacoustic Raw Time Series Data

The first part of the document focuses on the standardized description of PA raw time series data. PA raw time series data refer to the unprocessed signals recorded by the detection elements of an PA system. On a data level (Figure 1), PA raw time series data are a block of binary data. The accompanying metadata are characterised by three main components: (1) The metadata of the file container, including a *universally unique identifier* (UUID) of the data and a UUID reference to the PA imaging device that was used, (2) the binary data metadata that make the binary data machine-readable, and (3) the acquisition metadata that enable interpretability of the recorded data and give context on the imaging settings.

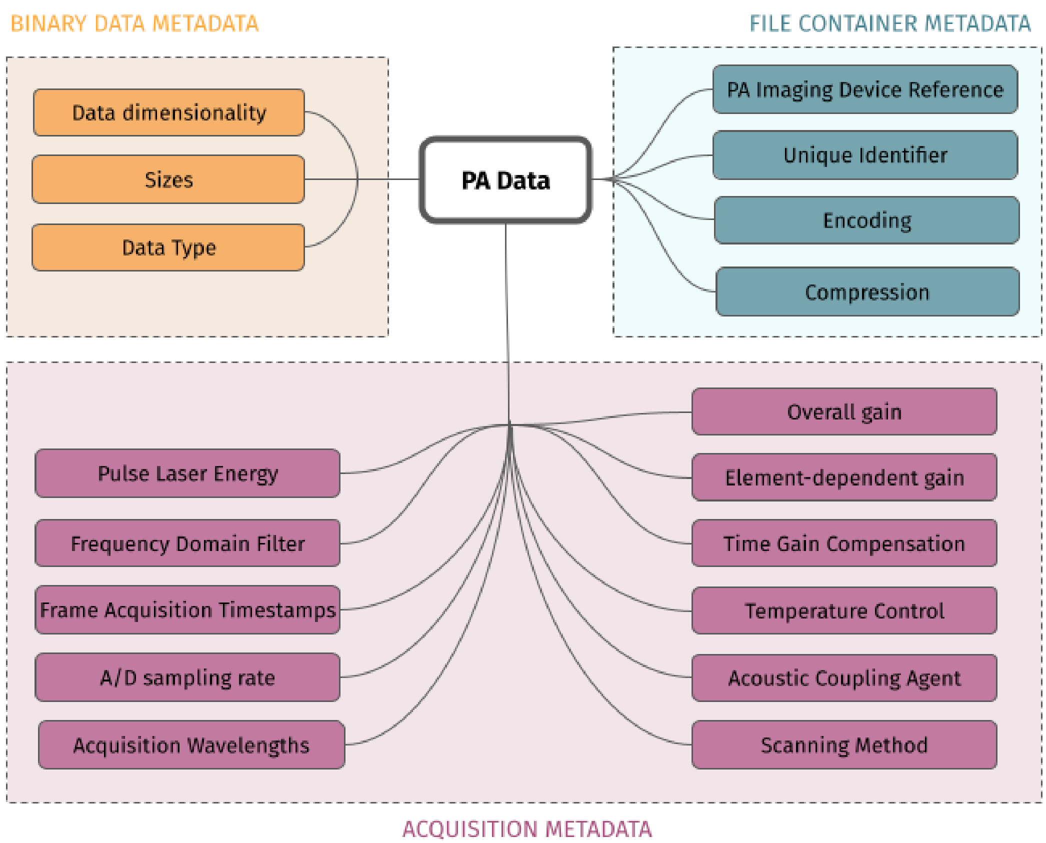


Figure 1: Graphical overview of the proposed three main elements of PA raw time series data: (1) file container format metadata, (2) acquisition metadata and (3) PA binary metadata.

File Container Format

The container format metadata refer to the inherent features of the file format which specify how the different elements of metadata are combined in a computer file.

Encoding

Necessity: Minimal
dtype: String
Description: The *encoding* field defines the character set that was used to encode the binary data and the metadata, e.g. one of 'UTF-8', 'ASCII', 'CP-1252' etc.
Method Name: get_encoding()

Compression

Necessity: Minimal
dtype: String
Description: The *compression* field defines the compression method that was used to compress the binary data, e.g. one of 'raw', 'gzip etc.
Method Name: get_compression()

Universally Unique Identifier

Necessity: Minimal
dtype: String
Description: The universally unique identifier (*UUID)* is a unique identifier of the data that can be referenced.
Condition: 128-bit Integer displayed as a hexadecimal string in 5 groups separated by hyphens, in the form 8-4-4-4-12 for a total of 36 characters. The UUID is randomly generated using the UUID version 4 standard.
Method Name: get_data_UUID()

Binary Data Metadata

The binary data are formatted as: [detectors, samples, wavelength, frames]. Depending on the binary data metadata, the size of these arrays varies. The interpretation of the frames field depends on the *dimensionality* field.

Data Type

Necessity: Minimal
dtype: String

Description: The *data type* field represents the datatype of the binary data. This field is given in the C++ data type naming convention, e.g. ‘short’, ‘unsigned short’, ‘int’, ‘unsigned int’, ‘long’, ‘unsigned long’, ‘long long’, ‘float’, ‘double’, ‘long double’.
Method Name: get_data_type()

Dimensionality

Necessity: Minimal
dtype: String
Description: The *dimensionality* field represents the definition of the ‘frames’ field and can be either [‘time’, ‘space’, or ‘time and space’].
Method Name: get_dimensionality()

Sizes

Necessity: Minimal
dtype: Integer array
Units: Dimensionless Quantity (the units can be inferred in combination with *Dimensionality* and the detection and illumination geometry).
Description: The *sizes* field quantifies the number of data points in each of the dimensions specified in the *dimensionality* field. As such, it defines the respective sizes of each element of the binary data which are: [detectors, samples, wavelengths, frames].
Method Name: get_sizes()

Acquisition Metadata

Regions of Interest

Necessity: Report if present
dtype: Dictionary [String, 2D double array (6, 3)] where the first number in the array represents the number of coordinates and the second number represents the coordinate values.
Units: Meter [m]
Description: The *regions of interest* field is a list of named areas of interest within the device coordinate system. Each area of interest is of approximate cubic volume described by the coordinates: [x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}].
Method Name: get_region_of_interest()

Photoacoustic Imaging Device Reference

Necessity: Report if present
dtype: String
Description: The *Photoacoustic Imaging Device Reference* field specifies a reference to the UUID of the PA imaging device description as defined in part 1. *This field will be used for future*

versions of the data format, where the device metadata may not be stored within the file but will be accessible via a web service.
Method Name: get_device_reference()

Pulse Laser Energy

Necessity: Report if present
dtype: Double array
Units: Joule [J]
Description: The *pulse laser energy* field specifies the pulse-to-pulse laser energy that is measured during acquisition of the raw time series data. If the pulse laser energies are averaged over many pulses, an average value must be specified. If the pulse laser energy has already been accounted for, the array must read [0].
Condition: Array size must be the same as the size of '*frames*' specified in the *sizes* field, except for the case of [0]. It *can* also be of shape [detection_elements, frames] in case that laser pulses are fired individually for each detection element.
Method Name: get_pulse_laser_energy()

Frame Acquisition Timestamps

Necessity: Report if present
dtype: Double array
Units: seconds [s]
Description: The *frame acquisition timestamps* field indicates the timestamp of the acquisition system.
Condition: Array size must be the same as the size of '*frames*' specified in the *sizes* field. Timestamps are given in seconds with the elapsed time since *epoch* (Jan 1st 1970, 00:00).
Method Name: get_time_stamps()

Frame Acquisition Spatial Positions

Necessity: Report if present
dtype: 2D double array of 6D coordinates (N, 6)
Units: Meters [m]
Description: The *frame acquisition spatial positions* field indicates a relative movement of the acquisition system to the frame of reference (first frame). The entire coordinate system is moved based on the spatial positions. If the frame stays constant, N equals 0.
Condition: Array size must be the same as the size of '*frames*' specified in the *sizes* field.
Method Name: get_frame_spatial_positions()

Acquisition Optical Wavelengths

Necessity: Minimal
dtype: Array
Units: Meters [m]

Description: The *acquisition optical wavelengths* field is an array of all wavelengths used for the image acquisition.
Method Name: get_wavelengths()

Time Gain Compensation

Necessity: Report if present
dtype: Double array
Units: Dimensionless unit
Description: The *time gain compensation* field is a 1D array that contains relative factors which are used to correct the time series data for the effect of acoustic attenuation.
Condition: The *time gain compensation* array has the same dimension as the *samples* dimension [samples]. It *can* also be of shape [detection_elements, samples] if measurements are acquired individually for each detection element.
Method Name: get_time_gain_compensation()

Overall Gain

Necessity: Report if present
dtype: Double
Units: Dimensionless unit
Description: The *overall gain* is a single value describing a factor that has been applied to all values of the raw time series data.
Method Name: get_overall_gain()

Element-dependent Gain

Necessity: Report if present
dtype: Double array [num_detectors]
Units: Dimensionless unit
Description: The *element-dependent gain* field is a 2D array that contains the relative factors which are used to perform apodization.
Condition: The *element-dependent gain* is a double array that has the same dimension as the number of detectors.
Method Name: get_element_dependent_gain()

Temperature Control

Necessity: Report if present
dtype: Double array
Units: Kelvin [K]
Description: The *temperature control* field indicates the temperature during image acquisition.
Condition: The *temperature control* array either has the same dimension as the number of '*frames*', or is a single value indicating a constant temperature over all frames.
Method Name: get_temperature()

Acoustic Coupling Agent

Necessity: Report if present
dtype: String
Description: The *Acoustic Coupling Agent* field is a string representation of the acoustic coupling agent that is used, e.g. D_2O, H_2O, gel, etc.
Method Name: get_coupling_agent()

Assumed Global Speed of Sound

Necessity: Report if present
dtype: Double
Units: Meters per second [m/s]
Description: The *Assumed Global Speed of Sound* field represents the assumed speed of sound in the imaged space, covering both the imaged medium and the coupling agent.
Method Name: get_assumed_speed_of_sound()

Scanning Method

Necessity: Report if present
dtype: String
Description: The *Scanning Method* field is a string representation of the *scanning method* that was used, i.e. “raster_scan” or“full_scan”. This flag determines how the field “frame” is defined.
Method Name: get_scanning_method()

Frames Per Image

Necessity: Report if present
dtype: Integer
Units: Dimensionless unit
Description: The *Frames Per Image* field describes the number of scan frames that constitutes one image.
Method Name: get_frames_per_image()

A/D (Analog/Digital) Sampling Rate

Necessity: Minimal
dtype: Double
Units: Hertz [Hz] (samples / second)
Description: The *A/D sampling rate* refers to the rate at which samples of the analogue signal are taken to be converted into digital form.
Method Name: get_sampling_rate()

Frequency Domain Filter

Necessity: Report if present

dtype: Double array
Units: Hertz [Hz] (samples / second)
Description: The *Frequency Domain Filter* field specifies the frequency threshold levels that are applied to filter the raw time series data. The threshold levels are given in the form of [lower, upper] limits of the filter. [lower, -1] denotes a low-pass filter and [-1, higher] denotes a high-pass filter.
Method Name: get_frequency_filter()

Part 2 - Photoacoustic Imaging Device

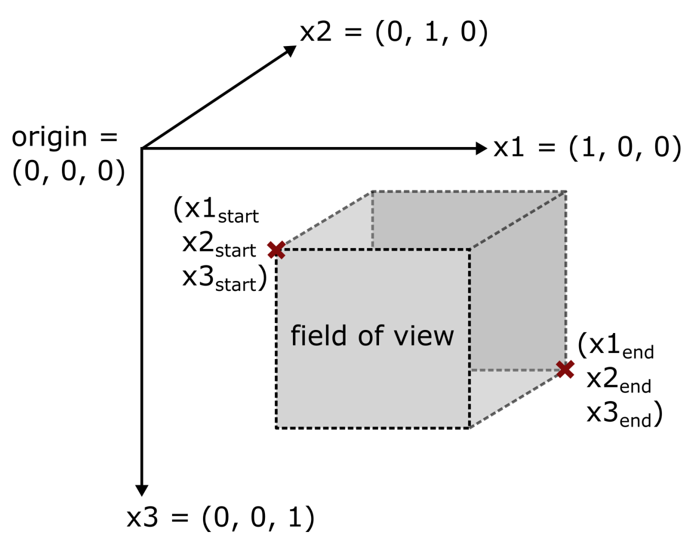
The second part of this document defines all the information necessary to describe a PA imaging device. By collecting this information, a *digital twin* of the imaging device hardware can be created and potentially stored in a database of commercial and home-built PA devices in future versions of the format. Each system is assigned a unique identifier which is referenced in the recorded PA data.

To enable image reconstruction from time series data, the positions of the detector elements need to be defined. To this end, a coordinate system is used that meets the following conditions: The axes of the coordinate systems are x1=(1, 0, 0), x2=(0, 1, 0), and x3=(0, 0, 1) and are defined in units of meters [m].

Apart from these boundary conditions, the metadata standard does not prescribe a standard coordinate system definition as it is sufficient when all axes definitions remain consistent with the field of view definition of the referenced device metadata.

For ease of use, we suggest using the following convention for devices that collect time series data to be reconstructed into 2D images:

- the x1 axis should be defined as the horizontal axis of the imaging plane
- the x2 axis should be defined as the vertical axis of the imaging plane
- the x3 axis should be defined as the normal to the imaging plane



In summary, all coordinate positions of individual elements will be given in cartesian coordinates consistent with the field of view definition of the device. Since the field of view and element positions of the imaging device reside in this same coordinate system, the origin of the imaging system can be defined differently for different devices. In every case, the relative positions must be consistent.

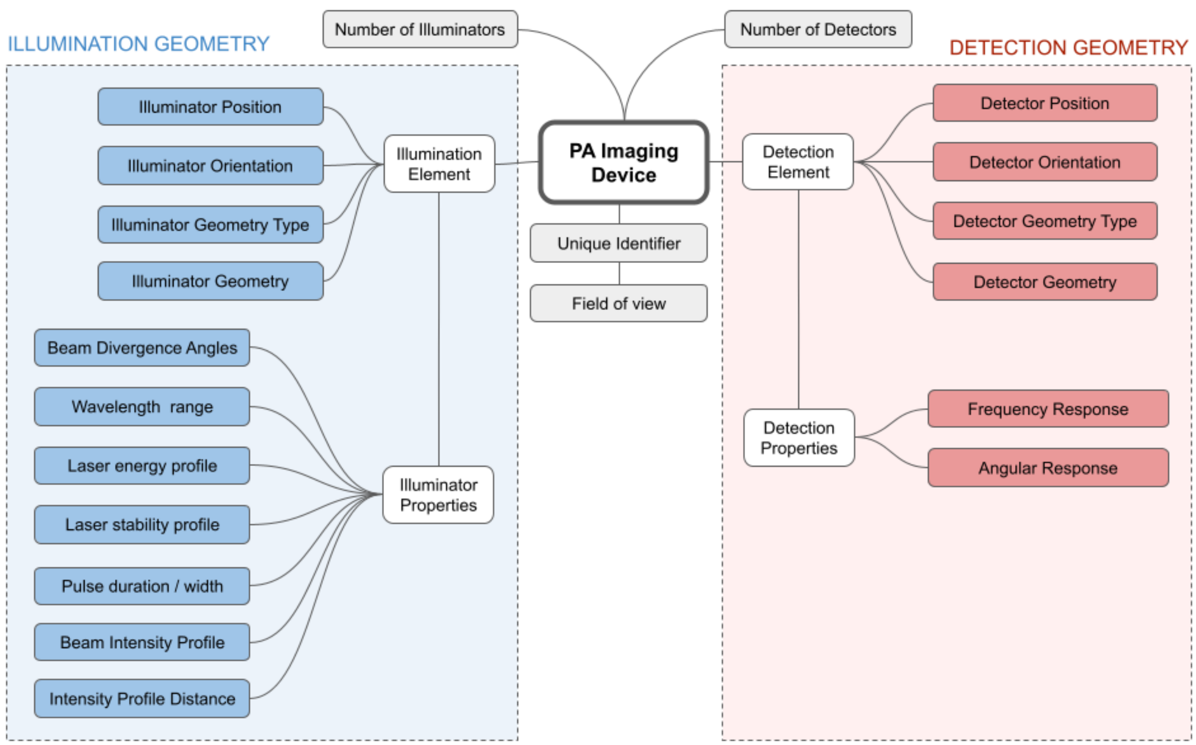


Figure 2: Graphical overview of the proposed *Imaging Device Metadata* divided into (1) the illumination geometry and (2) the detection geometry.

Universally Unique Identifier

Necessity: Minimal
dtype: String
Description: The universally unique identifier (UUID) for the device that can be referenced.
Method Name: get_device_uuid()
Condition: 128-bit Integer displayed as a hexadecimal string in 5 groups separated by hyphens, in the form 8-4-4-4-12 for a total of 36 characters. The UUID is randomly generated using the UUID version 4 standard.

Field of View

Necessity: Report if present
dtype: 2D double array of length 6
Units: Meter [m]
Description: The *field of view* defines an approximate cube of the area detectable by the PA imaging device in 3D cartesian coordinates [x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}].
Method Name: get_field_of_view()

Number of Illumination Elements

Necessity: Report if present
dtype: Integer
Units: Dimensionless unit
Description: The *number of illumination elements* quantifies the number of illuminators that are used in the PA imaging device. Each of these illuminators is described by a set of illumination geometry parameters.
Method Name: get_number_of_illumination_elements()

Number of Detection Elements

Necessity: Minimal
dtype: Integer
Units: Dimensionless unit
Description: The *number of detection elements* quantifies the number of transducer elements used for detection in the PA imaging device. Each of these transducer elements is described by a set of detection geometry parameters.
Method Name: get_number_of_detection_elements()

Illumination Element

Illuminator Position

Necessity: Report if present
dtype: 1D double array

Units: Meter [m]
Description: The *illuminator position* defines the position of the illuminator centroid in 3D cartesian coordinates [x_1, x_2, x_3] .
Method Name: get_illuminator_position()

Illuminator Orientation

Necessity: Report if present
dtype: 1D double array
Units: Meters [m]
Description: The *illuminator orientation* defines the direction unit vector of the illuminator in 3D cartesian coordinates [x_d, y_d, z_d] . It is the normal of the planar illuminator surface.
Method Name: get_illuminator_orientation()

Illuminator Geometry Type

Necessity: Report if present
dtype: String
Description: The *illuminator geometry type* defines the shape of the optical fibre (bundle) output. It determines the interpretation of the data in the *illuminator geometry* field. The following geometry types are currently supported:

- "CIRCULAR" - defined by a single value that determines the radius of the circle
- "SPHERE" - defined by a single value that determines the radius of the sphere
- "CUBOID" - defined by three values that determine the extent of the cuboid in x, y, and z dimensions before the position and orientation transforms.
- "MESH" - defined by an *STL-formatted* string that determines the positions of points and faces before the position and orientation transforms.

Method Name: get_illuminator_geometry_type()

Illuminator Geometry

Necessity: Report if present
dtype: Double, double array, or byte array
Units: Meter [m]
Description: The *illuminator geometry* defines the numerical geometry of the optical fibre (bundle) output. The data type and content of this metadatum are determined by the *illuminator geometry type* field. The given coordinates are interpreted relative to the *illuminator position*.
Method Name: get_illuminator_geometry()

Illuminator Properties

Wavelength Range

Necessity: Report if present
dtype: 1D double array
Units: Meters [m]
Description: The *wavelength range* quantifies the wavelengths λ that can be generated by the illuminator. Three values can be reported: the minimum wavelength λ_{max}, the maximum wavelength λ_{max} and a metric for the accuracy $\lambda_{accuracy}$: (λ_{min}, λ_{max}, $\lambda_{accuracy}$). These parameters could be for instance (700, 900, 1.2), meaning that this illuminator can be tuned from 700 nm to 900 nm with an accuracy of 1.2 nm. Single-wavelength elements are specified as: (λ_{actual}, λ_{actual}, $\lambda_{accuracy}$).
Method Name: get_wavelength_range()

Laser Energy Profile

Necessity: Report if present
dtype: Array of two 1D double arrays [wavelengths, energies], where the first array comprises the wavelengths and the second array comprises the laser energies.
Units: Joule [J]
Description: The *laser energy profile* field is a discretized function of the wavelength (nm) describing the laser energy of the illuminator. Thereby, systematic differences in multispectral image acquisitions can be accounted for.
Condition: The *laser energy profile* function is well defined and non-negative in the *wavelength range*.
Method Name: get_energy_profile()

Laser Stability Profile

Necessity: Report if present
dtype: Array of two 1D double arrays [wavelengths, standard_deviations], where the first array comprises the wavelengths and the second array comprises the standard deviations of the laser energies.
Units: Joule [J]
Description: The *laser stability profile* field is a function of the wavelength (nm) and represents the standard deviation of the pulse-to-pulse laser energy of the illuminator.
Condition: The *laser stability profile* function is well defined and non-negative in the *wavelength range*.
Method Name: get_stability_profile()

Pulse Duration / Width

Necessity: Report if present
dtype: Double
Units: Seconds [s]
Description: The *pulse duration* or *pulse width* describes the total length of a laser pulse measured as the time interval between the half-power points on the leading and trailing edges of the pulse.
Method Name: get_pulse_width()

Beam Intensity Profile

Necessity: Report if present
dtype: Array of two double arrays [positions, intensities] with intensities and their corresponding positions.
Units: Normalised units (to the maximum intensity)
Description: The *beam intensity profile* is a function of a spatial position that specifies the relative laser beam intensity according to the planar emitting surface of the *illuminator shape* at the distance defined in *intensity profile distance*. For points between specified positions, it is assumed that the values are linearly interpolated from their closest neighbours. The positions are generally in 2D.
Method Name: get_beam_profile()

Intensity Profile Distance

Necessity: Report if present
dtype: Double
Units: Meters [m]
Description: The distance from the light source for measuring its beam intensity profile. This distance is to be measured from the *Illuminator Position* along with the *Illuminator Orientation*.
Method Name: get_beam_profile_distance()

Beam Divergence Angles

Necessity: Report if present
dtype: Double
Units: Radians [rad]
Description: The *beam divergence angles* represent the opening angles of the laser beam from the *illuminator shape* with respect to the orientation vector. This angle is represented by the standard deviation of the beam divergence.
Method Name: get_beam_divergence()

Detection Element

Detector Position

Necessity: Minimal
dtype: Double array
Units: Meter [m]
Description: The *detector position* defines the position of the detection element centroid in 3D cartesian coordinates [x_1, x_2, x_3] .
Method Name: get_detector_position()

Detector Orientation

Necessity: Report if present
dtype: Double array
Units: Meters [m]
Description: The *detector orientation* defines the direction unit vector of the detector in 3D cartesian coordinates [x_d, y_d, z_d] .
Method Name: get_detector_orientation()

Detector Geometry Type

Necessity: Report if present
Dtype: String
Description: The *detector geometry type* defines how to interpret the data in the *detector geometry* field. The following geometry types are currently supported:

- "CIRCULAR" - defined by a single value that determines the radius of the circle
- "SPHERE" - defined by a single value that determines the radius of the sphere
- "CUBOID" - defined by three values that determine the extent of the cuboid in x, y, and z dimensions before the position and orientation transforms.
- "MESH" - defined by an *STL-formatted* string that determines the positions of points and faces before the position and orientation transforms.

Method Name: get_detector_geometry_type()

Detector Geometry

Necessity: Report if present
Dtype: Double, double array, or byte array
Units: Meter [m]
Description: The *detector geometry* defines the shape of the detector elements. The data type and the contents of the shape field are determined bythe *detector geometry type* field. The given coordinates are interpreted relative to the d*etector position*.
Method Name: get_detector_geometry()

Detection Element Properties

Frequency Response

Necessity: Report if present
dtype: Array with two components, where the first component is the frequency (in Hertz [s^{-1}]) and the second component is the response value (in normalised units).
Units: Hertz [s^{-1}], normalised units (to the maximum intensity)
Description: The *frequency response* is a function of frequency that characterises the response of the detection element according to the frequency of incident pressure waves. In the case of a sparse definition of this value, the actual values are linearly interpolated between the closest neighbours. If the value is of shape [c, b] then it is assumed to be interpreted as c=centre frequency and b=bandwidth (measured at -6 dB).
Method Name: get_frequency_response()

Angular Response

Necessity: Report if present
dtype: Array with two components, where the first component is the incident angle in radians and the second component is the normalised response value.
Units: Radians [rad], Normalised Units (to the maximum efficiency)
Description: The *angular response* field characterises the angular sensitivity of the detection element to the incident angle (relative to the element's orientation) of the incoming pressure wave. If only one value (the angle [a]) is given, then the value is interpreted as a=limiting angle (where the response drops to -6 dB).
Method Name: get_angular_response()

References

[1] O'Connor, J., Aboagye, E., Adams, J. *et al.* Imaging biomarker roadmap for cancer studies. *Nat Rev Clin Oncol* 14, 169–186 (2017). https://doi.org/10.1038/nrclinonc.2016.162

[2] IEC TR 60854 1986, Methods of measuring the performance of ultrasonic pulse-echo diagnostic equipment, (1986). https://webstore.iec.ch/publication/3713 *Accessed 17/06/2021*

Version History

Version 2.0:

Feedback from the second consensus vote was implemented.
Editing: Rephrased some definitions for conciseness. Updated texts to have consistent spellings, usage of upper case, enumerations, etc.

Version 1.2:

Changed Regions of Interest definition. Changed "Shape" definition to be the "Geometry" definition and allow for different "GeometryTypes" to facilitate defining simple setups while still allowing arbitrary designs in the form of an ASCII STL file.
Integrated further feedback from the consortium.

Version 1.1:

Feedback from DAM Theme conversion tool hacking integrated. Standardized the method names for ALL programming languages already in this document.
Changed definitions of orientations and better integrated the concept of "frames" throughout the standard. Fixed the internal formatting of the binary data. Some wording and spelling error changes. Added more details on the layout of arrays.

Version 1.0:

Feedback from the first consensus vote was implemented.

Version 0.3:

Document revised after the first annual meeting. Addition of minimal vs. optional parameters. Consistent formatting.

Version 0.2:

Document revised mainly by dividing it into two main parts: The description of PA devices and the metadata that accompanies recorded data. Removed the ultrasound data in the current version. The first release should deal with PA raw time series data primarily.

Version 0.1:

Document revised after extensive feedback from the IPASC DAM theme members. Added ‘Necessity’ and ‘Nested Attributes’.

Version 0:

The document was initially drafted by Janek Gröhl and Lina Hacker.