	IXPE	**Ver. 1.0E**
	The Imaging X-ray Polarimetry Explorer User Software Guide	Aug 2026

IXPE User Software Guide

Prepared by: IXPE Science Operations Center

NASA, Marshall Space Flight Center

August 01, 2026

Abstract

This document describes mission-specific FTOOLS tasks developed by the Imaging X-ray Polarimetry Explorer Science Operations Center team as an aid to the science community for data processing, visualization, and analysis of IXPE science observations.

Revision History

Revision	Author	Date	Change
Ver. 1.0E	IXPE SOC	01 Aug 2026	Initial Release*

*Supersedes Document No.: IXPE-SOC-DOC-009D_UG-Software

Contents

1 Introduction

The Imaging X-ray Polarimetry Explorer (IXPE) is a NASA Small Explorer mission, in collaboration with the Italian Space Agency (ASI), providing polarimetric imaging and spectral capabilities in the 2−8 keV energy range. IXPE was launched on 2021 Dec 9 into a 600-km-altitude, nearly equatorial, 96.6-minute-period circular orbit. Its 2-year prime science mission began 2022 Jan 11 after a brief commissioning and calibration phase. IXPE began its General Observer program 2024 Feb 03.

This document serves as a User Guide to software provided by the IXPE Science Operations Center (SOC) located at the NASA Marshall Space Flight Center (MSFC). The software is provided through NASA's High Energy Astrophysics Science Archive Research Center (HEASARC) in the form of stand-alone mission-specific FTOOLS tasks as part of the HEASoft packages. FTOOLS operate on Flexible Image Transport System (FITS) files. FITS files have one or more extensions called Header/Data Units (HDUs). FITS headers contain keywords and keyword values. Several keywords are discussed in this document. The IXPE mission science and engineering products and the IXPE calibration database (CALDB) are all FITS-formatted files meeting Office for Guest Investigators Programs (OGIP) standards.

All FTOOLS tasks are command-line executables with variable parameters either as given at the command line typically as `parameter=value` pairs or as stored in task-specific parameter files whose values can be listed with the `plist` command and redefined with the `punlearn` and `pset` commands.

This document provides an overview of each IXPE-specific task, briefly explains the purpose of the task, lists all required and certain optional task parameters and nominal parameter values (often paths to FITS files), and gives usage examples. The full documentation including all required, optional, and hidden task parameters can be displayed at the command line with the `fhelp` command and are also available online at the IXPE mission-specific FTOOLS page.

The IXPE mission-specific tasks are divided into two groups. The first group, described in §2, are stand-alone tasks that are useful aids to preparing IXPE data for analysis and for extracting basic physical properties of an observation from the data. The second group of tasks, §3, are intended to be executed as a prescribed sequence to recreate Level 2 files beginning with Level 1 files. Together, tasks in this second group essentially reproduce the SOC data production pipeline and can be used, e.g., to reprocess data using a different set of calibration files.

Many of the FTOOL task parameters default to pointers to CALDB files. The IXPE CALDB is available from the HEASARC. It contains basic calibration files (bcf) and calibration product files (cpf) relevant to the gas pixel detector (GPD) instruments and to the X-ray Telescope (XRT) mirror assemblies.

Details of the IXPE data file structures and data contents are given in the IXPE-SOC-DOC-007D_UG-DataFormats.pdf documentation. In the following discussion, the file naming conventions defined in the Data Formats guide are adopted with Detector Unit (DU) numbers denoted '`D`', observation identification numbers, i.e., Unique IDs, denoted '`UID`', and file version numbers denoted '`v??`', e.g., ixpe`UID`_det`D`_evt1_`v??`.fits. Where explicit values of these variables are used (for greater clarity), they are `UID=05005001`, `D=2`, and `v??=v01`, e.g., `ixpe05005001_det2_evt1_v01.fits`. FTOOLS tasks are written in *italics*, task parameters are written with a trailing '`=`' sign in typewriter font, e.g., `infile=`, and FITS column names and keywords are written as `ALL_CAPS`. The '`>`' symbol is used to denote the command line.

IXPE mission data are archived at the HEASARC. The basic IXPE directory structure has subdirectories for auxiliary files `/auxil/`, Level 1 (`/event_l1/`) and Level 2 (`/event_l2/`) event lists, and spacecraft housekeeping data files `/hk/`. All files are FITS formatted files.

2 IXPE Data Analysis Support Tasks

The first set of IXPE FTOOLS tasks, listed in Table 1, are tools to aid in preparing for spectro-polarimetric analysis including producing observation-specific ancillary response files (*ixpecalcarf*) and exposure maps (*ixpeexpmap*). Other tasks display detector fields of view for a given sky location, date, and time (*ipecalcfov*), display individual event images (*ixpeeventviewer*), and compute basic, model-independent, polarization statistics (*ixpepolarization*) and display the results (*ixpeplot_polarization*).

The comprehensive task *ixpeproduct* combines several tasks to produce *XSPEC*-ready[1] source and background Stokes spectra (and light curves) using *extractor*, the core program of HEASARC's *Xselect* tool, instrument response files using *ixpecalcarf* (and CALDB files), model-independent polarization statistics using *ixpepolarization*, and *XSPEC* initialization command files using *ixpemkxspec*. All tasks listed in Table 1 are described in the following subsections; several are also described in the IXPE Quick Start Analysis Guide, Ref. [1], and all are documented in *fhelp* command-line queries.

Table 1: IXPE Stand-alone Data Preparation and Analysis Support FTOOLS Tasks

ixpecalcarf	Corrects an on-axis ARF or MRF file for off-axis vignetting and extraction radius
ixpeexpmap	Creates an exposure map from aspect solution data
ixpecalcfov	Calculates a map of the nominal sky field of view for each detector
ixpeeventviewer	Displays images of raw event tracks with the event barycenter, absorption point, and initial electron direction overlaid
ixpepolarization	Calculates Stokes parameters for a user-defined region
ixpeplot_polarization	Plots Stokes parameters and MDP99 from *ixpepolarization* output in the (Q, U) plane
ixpeproduct	Extracts high-level data products including Stokes spectra, light curves, and Stokes parameters, from Level 2 data
ixpemkxspec	Generates an *XSPEC* command file from a list of IXPE spectral files
ixpeversion	Reports the IXPE software installation version

2.1 *ixpecalcarf*

Example command line usage:

```
> ixpecalcarf evtfile=./event_l2/ixpe05005001_det2_evt2_v01.fits.gz   \
>     pmapfile=pointingmap_det2.fits regfile=src.reg \
>     specfile=myspectrum.pi
```

or

```
> ixpecalcarf evtfile=./event_l2/ixpe05005001_det2_evt2_v01.fits.gz   \
>     pmapfile=pointingmap_det2.fits regfile=src.reg \
>     weight=1 resptype=mrf arfout=ixpe_det1.mrf specfile=none
```

or

```
> ixpecalcarf evtfile=./event_l2/ixpe05005001_det2_evt2_v01.fits.gz   \
>     attfiles=./hk/ixpe05005001_det2_att_v01.fits.gz radius=1.0 \
>     weight=1 resptype=mrf arfout=ixpe_det1.mrf specfile=none
```

The *ixpecalcarf* task generates observation-specific Ancillary Response Files (ARFs) for Stokes I spectral analysis and Modulation Response Files (MRFs) for Stokes Q and U. *ixpecalcarf* requires a Level 2 event list file specified by the `evtfile=` parameter value and the name of a spectral file given by the `specfile=` parameter value as in the first example above. If `specfile=none` as in the remaining examples, then additional parameters must be specified in order to provide the information that is expected from an existing spectral file (see **Inputting Spectral Files** below).

The IXPE response is not uniform across the detector because of the reflective properties of the telescope mirror assemblies which cause energy-dependent vignetting resulting in a gradual decrease in the effective area from the peak on-axis value towards the edges of the field of view. This effect must

[1] *XSPEC* is a spectral data fitting package available from the HEASARC

be accounted for even for an on-axis point source because of spacecraft dither. In addition, *ixpecalcarf* accounts for the finite mirror point spread function by correcting the instrument response by the encircled energy fraction contained within the specified source region.

Two of the most used means of specifying the sky region over which the response files are to be calculated are illustrated in the above examples. If a sky region is specified by the value of the `regfile=` parameter, then a pointing map file must also be specified by the `pmapfile=` parameter value. The file specified by `regfile=` must be an SAO-image DS9 formatted[2] region file. The pointing map file is an intermediate product of the task *ixpeexpmap*; see §2.2. Alternatively, the size of a circular on-axis region can be specified by the value of the `radius=` parameter (second example), in which case a Level 1 attitude file must be specified by the `attfiles=` parameter value. Note that some observations have multiple attitude files. These can be specified as a comma-separated list or using the standard FTOOLS syntax `parameter=@filename.txt` where `filename.txt` is an ASCII file containing a list of file name paths, one pathname per line. In either case, the pointing map or the attitude files will provide the time history of the spacecraft orientation needed to properly account for spacecraft dither.

The type of response file *ixpecalcarf* generates can be specified by the `resptype=` parameter value which can be either `=arf` or `=mrf`. The spectral weighting scheme to be applied is specified by the `weight=` parameter. Values can be `=0` for UNWEIGHTED, `=1` for NEFF weighting, or `=2` for SIMPLE weighting[3] If the gray filter was inserted during the observation, then the parameter `detfilter=1` should be specified (default: `=0` for no filter). The filter usage is documented in the Level 2 event list header keyword FCW_POS and has the value 0 if open (no filter) or 1 (gray filter). The name of the resulting response file is given by the `arfout=` parameter value.

Please refer to the *ixpecalcarf* `fhelp` file for the full listing of parameters and for several additional usage examples.

Inputting Spectral Files: Often, invoking *ixpecalcarf* is done after Stokes spectral files have been extracted from Level 2 event lists. If the extraction software follows OGIP standards, then certain keywords are inserted into each spectral file header identifying what Stokes spectrum is contained in the file (I, Q, or U; keywords `XFLT0001` or `STOKESPR`, often both are present) and which weighting scheme was applied (NEFF, SIMPLE, or UNWEIGHTED; keyword `STOKESWT`). If this is the case, specifying such a spectral file with the `specfile=` parameter value (as in the first usage example above) will ensure the correct response file (MRF or ARF) with the correct weighting will be generated by *ixpecalcarf* for that particular spectral file. In this case, the optional parameters `resptype=` and `weight=` should not be specified. The *ixpeproduct* task, §2.7, extracts Stokes spectral files that follow the OGIP standards and that contain the proper header keywords. If in doubt, the keywords can be listed using either the FTOOL utility *fkeyprint* or *ftlist*.

2.2 *ixpeexpmap*

Example command line usage:

```
> ixpeexpmap infile=./hk/ixpe05005001_det2_att_v01.fits.gz \
>     gti=./event_l2/xpe05005001_det2_evt2_v01.fits.gz outfile=expmap.fits \
>     pntmap=yes pntname=pointingmap_det2.fits
```

ixpeexpmap creates an exposure map from housekeeping aspect solution data from a file named by the `infile=` parameter value for events recorded within Good Time Intervals (GTIs) specified by the `gti=` parameter value file name. Typically, the GTIs are taken from the corresponding cleaned and calibrated Level 2 event list GTI Header/Data Unit (HDU) extension. *ixpeexpmap* builds a pointing map and a pixel map and convolves the two to produce the final exposure map FITS file with a name specified by the `outfile=` parameter value. The usage example given here will also save the intermediate pointing map to a file specified by the `pntname=` parameter value when `pntmap=yes` (default: `=no`). This intermediate file is used by the *ixpecalcarf* task under certain circumstances, see §2.1.

[2] SAOImage DS9 is a widely used astronomical imaging and data visualization application available from the Center for Astrophysics.

[3] Weighting is an integral part of IXPE spectro-polarimetric data analysis. However, the advantages and disadvantages of different weighting schemes in various contexts is beyond the scope of this document. See Ref. [2] for discussion of NEFF weighting and Ref. [3] for SIMPLE weighting.

The pointing map contains J2000 sky tangent-plane coordinates with the value at each coordinate given by the exposure time at that location. This is essentially a sky map of the location of the mirror module optical axis. *ixpeexpmap* then calculates a pixel map (with bad pixels excluded and "gray" pixels that buffer bad pixels excluded if `exc_graypix=yes` (default: `=no`)) and then convolves the pointing and pixel maps to generate the final exposure map specified by `outfile=` with the appropriate rotations, offsets, and pixel-to-sky scalings applied to convert raw pixels to sky coordinates.

Note: The typical usage of *ixpeexpmap* described here uses the Level 1 aspect solution data contained in a housekeeping file as indicated by the `infile=` parameter value. The housekeeping file contains a record of the offset of the target from the optical axis of the mirror module assembly which is the correct information needed to calculate the pointing maps used by the task *ixpecalcarf* in order to account for vignetting effects in the generation of ARFs and MRFs. However, the Level 2 products have corrections applied to the aspect solution by tasks *ixpeboomdriftcorr* (§3.10) and *ixpeaspcorr* (§3.12). These tasks correct the position of an event in the detector plane due to distortions of the structure that affect the alignment of the detector relative to the mirror module assembly. They tend to sharpen the X-ray image; particularly when data from individual detector units are coadded. Since the beginning of IXPE Cycle 2 in early 2025, the processing pipeline has generated exposure maps incorporating these corrections. They are archived with names given generically as `./auxil/ixpeUID_detD_expmap2_v??.fits.gz`. For earlier observations, if corrected tangent-plane positions of the pointing direction (i.e., a corrected aspect solution) are available as output from either of these tasks or, preferably, both in series, then the value of the `infile=` parameter can be set to specify the corrected aspect file instead of the Level 1 housekeeping file. To do so, specify the *ixpeexpmap* hidden parameter `use_corr=yes` (default: `=no`) at the command line then *ixpeexpmap* will use the data in column `TXYDZ_CORR` (added by *ixpeaspcorr*) if present, otherwise *ixpeexpmap* will use `TXYDZ_CORRBD` (added by *ixpeboomdriftcorr*) if present. Set `use_corr=no` if only the `TXYDZ` column (included in the original Level 1 housekeeping attitude file) is present.

2.3 *ixpecalcfov*

Example command line usage:

```
> ixpecalcfov time=2026-04-15T12:24:46.001 ra=166.1138 dec=38.2088 outfile=fov.reg
```

ixpecalcfov calculates the nominal (undithered) sky field of view (FOV) for each IXPE detector and writes the result as a composite DS9-formatted region file specified by the `outfile=` parameter value with separate bounds for each detector with the proper rotation corresponding to the spacecraft roll at the time given by `time=` and the sky position given by `ra=`, `dec=` (the FOVs are squares "clocked" 120° relative to each other). *ixpecalcfov* is particularly useful for planning observations of extended sources. For example, the `outfile=` region file can be applied to an existing image (not necessarily an IXPE image) to help determine if the intended target region is contained within the IXPE FOV.

Though the `outfile=` parameter is optional, it is recommended that a meaningful region file name be specified as the default file name can be cumbersome.

2.4 *ixpeeventviewer*

Example command line usage:

```
> ixpeeventviewer pixpha=yes numclusters=4 \
>     evt1list=ixpe05005001_det1_evt1_v01.fits.gz \
>     evt2list=ixpe05005001_det1_evt2_v01.fits.gz
```

ixpeeventviewer displays 2-dimensional images of raw event tracks stored in a Level 1 event file given by the `evt1list=` parameter value with the computed event barycenter, absorption point, and initial electron direction overlaid. In addition, several quantities associated with the event are displayed. By specifying the corresponding Level 2 event list through the optional parameter `evt2list=`, only events with common `TIME` values, i.e, only valid cleaned and calibrated Level 2 events are displayed. In this case, the other quantities displayed include the event energy (in keV) and raw PHA value (in ADC counts), sky position (R.A. and Decl. in decimal degree) and detector location (in mm), Stokes q and

u values[4] observation start and end dates (in ISO 8601 format), and detector unit (DU) identifier. Furthermore, in this case, the optional energy range parameters `emin=` and `emax=` are implemented such that *ixpeeventviewer* only displays events with energies within the specified range (default: `=2.0` and `=8.0` keV, respectively). By default, *ixpeeventviewer* displays the chronologically first event and can advance to subsequent events through the "Previous" and "Next" buttons displayed on the graphics display window. The displayed image can be written (saved) to a file in one of several common formats.

ixpeeventviewer uses the same methodology applied by the pipeline task *ixpeevtrecon* (§3.1) to determine the photon interaction location, the initial direction of the photoelectron, and the event PHA value. *ixpeeventviewer* has many hidden parameters that can be adjusted to alter the behavior of the event reconstruction such as the zero suppression threshold (`zsupthresh=`) and certain clustering algorithm parameters (`clumindensity=`, `numclusters=`, and `cluminsize=`). Thus, *ixpeeventviewer* is a useful tool for gaining insight into the fundamentals of how IXPE data reduction determines the linear polarization of X-radiation. See §3.1 for more information. Example *ixpeeventviewer* output is shown in Fig. 1.

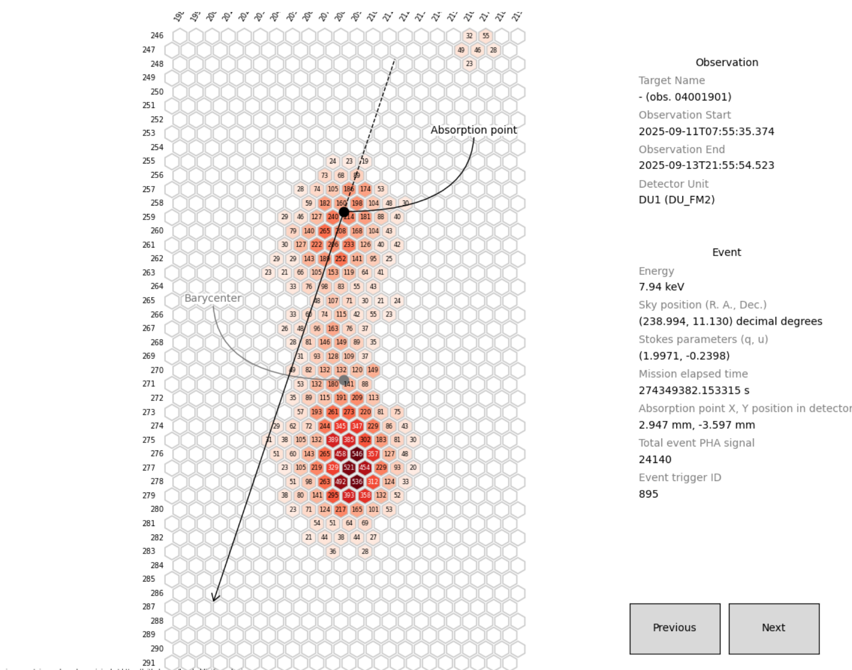

Figure 1: Track image display generated by *ixpeeventviewer*

2.5 *ixpepolarization*

Example command line usage:

```
> ixpepolarization infile=@level2files.txt regfile=src.reg \
>     pi_lo=50 pi_up=200 weight_scheme=unweighted \
>     outfile=ixpepolarization_out
```

ixpepolarization calculates the mean and standard deviation of the Stokes Q and U parameters and related quantities for a sky region specified by `regfile=` (in DS9 format), over a time interval `t_lo` to `t_up` (defaults: `TSTART` and `TSTOP`, respectively, i.e., the beginning and end of the observation in IXPE time units; stored as keywords in Level 2 event lists), and an energy range `pi_lo=` to `pi_up=` (defaults: `=0` and `=374`, respectively, i.e., the full IXPE operational energy range). The Stokes parameters are extracted

[4] Stokes q and u are in sky coordinates, i.e., $\psi = (1/2)\arctan(u/q)$ is measured counterclockwise from celestial North whereas *ixpeeventviewer* displays event electron tracks in detector coordinates.

directly from a set of Level 2 event list files specified by `infile=` (given in the `@level2files.txt` syntax in the example shown, where `level2files.txt` is an ASCII file containing a list of file name paths, one pathname per line; or as a comma-separated list at the command line), typically one file pathname is specified for each of the three DUs. The data extraction applies the weighting scheme specified by `weight_scheme=`. The weighting scheme values can be either `=UNWEIGHTED`, `=NEFF`, or `=SIMPLE` (case insensitive). Thus, the Stokes Q and U calculated by *ixpepolarization* are the weighted sums of the individual event q and u values over the specified region, time interval, and energy range. The task also returns the polarization magnitude, Π, polarization angle, ψ, the minimum detectable polarization at the 99% confidence level (MDP_{99}), and the mean modulation factor, $\langle\mu\rangle$, for each `infile=` and for the combination of all input event lists. See Ref. [1] for information on weighting schemes, polarization parameters, and related quantities.

The *ixpepolarization* task creates a FITS file with a name derived from the `outfile=` parameter by adding or changing the filename extension to `.fits`. The output file contains a separate HDU for each input Level 2 event list file, each containing a row for each sub-region specified in the `regfile=` region file, and an additional HDU named `COMBINED FILES` that contains the results of combining all event list files specified by the `infile=` parameter value. In the event that more than one Level 2 event list file is given for the same DU, then an additional HDU is included named `COMBINED DU?` where `?=1, 2, or 3` as appropriate. *ixpepolarization* can thus combine results from multiple observations of a target.

ixpepolarization also creates a DS9-style region file that consists of a line segment oriented parallel to the computed ψ with a length proportional to Π for the region (or for each member of a set of regions if the input source region file `regfile=` contains multiple regions). The output region file name is derived from the `outfile=` parameter (prepending `region_` to the filename and forcing a `.reg` filename extension). *ixpepolarization* can thus be applied to spatially resolve the polarization properties of an extended source region by utilizing the sub-region capability to, for example, define a grid on the sky that maps over the full region of interest.

It is recommended that an energy range be specified that lies within the 2 to 8 keV (PI channels 50 to 200) IXPE polarization-sensitive energy range and that a `regfile=` also be specified in order for *ixpepolarization* to return meaningful results as the default region is the entire sky region subtended by the detector FOV.

2.6 *ixpeplot_polarization*

Example command line usage:

```
> ixpeplot_polarization infile=ixpepolarization_out.fits
```

ixpeplot_polarization displays Stokes parameters and the MDP_{99} values contained in the output FITS file generated by the *ixpepolarization* task, §2.5. The task creates a graphical display of the (Q, U) plane showing the parameter pair means and 1-σ standard deviations (shown as error bars) for each DU and for all DUs combined. Also displayed are the individual and combined computed MDP_{99} values which appear as circles centered on the (Q, U) plane origin. The displayed image can be written (saved) to a file in one of several common formats. An example *ixpeplot_polarization* output is shown in Fig. 2.

2.7 *ixpeproduct*

Example command line usage:

```
> ixpeproduct indir=./mydata/05005001/ outdir=./mydata/05005001/level3/ \
>     weight=1 srcregfile=src.reg bgregfile=bkg.reg
```

ixpeproduct extracts high-level data products from IXPE Level 2 event list FITS files and the IXPE CALDB including Stokes spectra and light curves (using the FTOOL *extractor*, the core utility of the interactive command-line *Xselect* task), observation-specific instrument response functions (using *ixpecalcarf*, §2.1), and model-independent statistical analysis of Stokes parameters (using *ixpepolarization*, §2.5). *ixpeproduct* is primarily used to prepare data for spectro-polarimetric analysis with the *XSPEC* analysis package; it also generates *XSPEC* initialization command scripts (so-called `xcm` files).

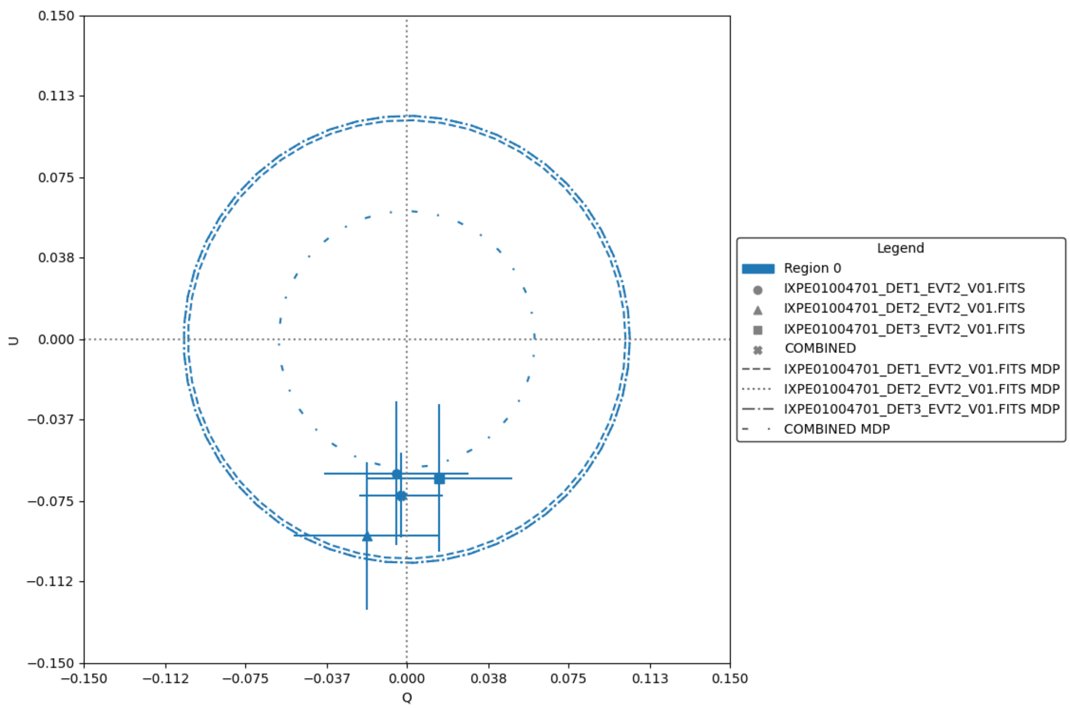

Figure 2: *ixpeplot_polarization* display of the (Q, U) plane with the parameter pair means and one-σ standard deviations shown for the individual DUs (circle, triangle, and square symbols) and for all three combined (cross). Also displayed are the individual and combined computed MDP_{99} values, shown as dotted and dot-dashed circles, respectively.

The *ixpeproduct* task assumes a data directory structure and file naming convention analogous to that used by the IXPE archive at the HEASARC. Specifically, given the directory specified by the `indir=./mydata/05005001/` parameter value in the above command line usage example, it assumes Level 2 event list files are located in subdirectory `./mydata/05005001/event_l2/` and that corresponding Level 1 attitude files are located in subdirectory `./mydata/05005001/hk/`. *ixpeproduct* operates on all Level 2 event list files present; typically, one file for each of the three detector units. *ixpeproduct* places all products generated in a directory specified by the `outdir=` parameter value. *ixpeproduct* will create the directory if it does not exist.

Sky Regions: *ixpeproduct* utilizes IXPE's imaging capabilities to extract data products from regions of the sky specified by the `srcregfile=` and `bgregfile=` parameter values which should be DS9-formatted region files describing source and background sky regions, respectively. If, instead, the `radius=` parameter is specified with a numerical value in arcminute, then *ixpeproduct* assumes an on-axis circular source region of the specified radius; a background region can still be specified by providing the `bgregfile=` parameter a valid value. Thus, Level 2 event list files should first be examined by displaying them with a data visualization application such as SAOImage DS9 in order to generate appropriate region files or to estimate an on-axis source region size. Note that, although there are measurable differences between each telescope's point spread function, Level 2 event lists are constructed such that the coordinates of the sky tangent plane are the same for all three telescopes so that, in general, the same region definition can be applied to all three DUs whether the region file specifies sky locations in, e.g., (x, y) physical pixel coordinates or in celestial $(RA, Decl.)$ coordinates such as the ICRS or FK5/J2000 reference frame (though the latter is preferred because it is detector and mission independent). Caution: If neither `srcregfile=` nor `radius=` parameters are specified, then *ixpeproduct* will extract products from the entire sky region subtended by the detector field of view.

Stokes Spectra: *ixpeproduct* uses the FTOOL *extractor* to extract Stokes I, Q, and U spectra from each Level 2 event list file located in the `/event_l2/` subdirectory of `indir=` from sky regions as described above. If, in addition, the parameter `bgregfile=` is specified, *ixpeproduct* will also extract background spectra from these regions and insert the `BACKFILE` keyword into the source spectral files with a value given by the name of the background spectral file it generates. Note that a `bgregfile=` parameter value can be specified regardless of whether the `srcregfile=` or the `radius=` parameter values are utilized.

The resulting Stokes spectra are recognizable by *XSPEC* from certain FITS header keywords inserted by *extractor* including the spectral type (I, Q, or U) and weighting scheme applied. The weighting scheme is specified by the `weight=` parameter value and can be set `=0` for UNWEIGHTED, `=1` for NEFF weighting or `=2` for SIMPLE weighting. textitixpeproduct also inserts header keywords `RESPFILE` and `ANCRFILE` specifying the associated IRFs into each source spectral file. Note that, for Stokes spectra, the resulting FITS spectral HDU columns are `CHANNEL`, `RATE`, and `STAT_ERR` rather than the more familiar `CHANNEL`, `COUNTS` pairs. This informs *XSPEC* not to use $\sqrt{N}$ for the uncertainty as that is not appropriate for weighted I spectra nor, in general, for Stokes Q and U spectra which could have negative values. Instead, *extractor* computes the proper statistical uncertainties depending on the specified `weight=` weighting scheme and includes them as the column `STAT_ERR`. See §4.1 of Ref. [1] for formal definitions of the weighted Stokes parameters and their uncertainties.

For IXPE data processed on or after 2024 Dec 18, instrumental background events are identified and flagged using the algorithm described in Ref. [4] (see also §3.11). If the *ixpeproduct* parameter `backrej=yes` (default: `=no`), then all events flagged as background are excluded from the Stokes spectra.

Instrument Response Files: *ixpeproduct* builds ancillary response files (ARFs) for Stokes I spectra, modulation response files (MRFs) for Stokes Q and U, and retrieves redistribution matrix files (RMFs) for all Stokes spectra from the CALDB appropriate for the time of observation, the region of the sky specified as described above, and the `weight=` parameter value selection. These files are necessary for spectro-polarimetric analysis using *XSPEC*. The names of these instrument response files (IRFs) are added as keyword values `ANCRFILE` (for ARFs and MRFs) and `RESPFILE` (for RMFs) in the spectral file headers as appropriate. The ARFs and MRFs are built using *ixpecalcarf* (§2.1). *ixpeproduct* queries the Level 2 header keyword `FCW_POS` to determine whether the neutral (gray) filter was deployed during the observation ensure the correct IRFs are generated.

When `srcregfile=` is specified, pointing map files(s) are used in generating instrument response files. Pointing map files are intermediate products of the *ixpeexpmap* task, §2.2. Pointing map files can be specified by the `pmapfile=` parameter values (as a comma-separated list or using the standard FTOOLS syntax `parameter=@filename.txt` where `filename.txt` is an ASCII file containing a list of file name paths, one pathname per line). If pointing map file names are not specified, *ixpeproduct* will search the `/hk/` subdirectory of the path specified by the `indir=` parameter value for pointing map files. If none are found, *ixpeproduct* will generate them by applying *ixpeexpmap* using Level 1 attitude files. When, instead, the `radius=` parameter is specified, Level 1 attitude files are used directly in generating instrument response files. Level 1 attitude files are required by *ixpeproduct* to be present in the `/hk/` subdirectory.

Light Curves: *ixpeproduct* uses the FTOOL *extractor* to generate tables of the time history of the intensity (count rate) of Level 2 events within the source sky region (as described above) and over Good Time Intervals contained in the GTI extension of the Level 2 event list files. By default, light curve time bins are 100 second intervals and bins with fractional bin size less than 0.2 are ignored. These values may be set by providing real number values to the `binlc=` and the `lcthresh=` parameters, respectively. Only events in the energy range specified by the parameters `en_lo=` and `en_up=` (defauls `=2.0` and `8.0`) are included in the light curve products. Event weighting is not applied to light curves.

Again, for IXPE data processed on or after 2024 Dec 18, instrumental background events are identified and flagged using the algorithm described in Ref. [4] (see also §3.11). If the *ixpeproduct* parameter `backrej=yes`, then all events flagged as background are excluded from the light curves.

Polarization Properties: The *ixpeproduct* task also performs a statistical analysis of the Stokes parameters using *ixpepolarization*, §2.5, for a sky region specified by `srcregfile=` or `radius=` (see above). For meaningful results, it is recommended that an energy range also be specified by the `en_lo=` and `en_up=` parameter values that lie within the 2 to 8 keV IXPE operational range (as already noted, those are the default values for these parameters, respectively).

Products: The resulting contents of the directory specified by the `outdir=` parameter value have the naming structure as listed in Table 2. (Occasionally, an observation must be separated into components for pipeline processing. When this is the case, the string `_c??` is added to certain file names, e.g., `ixpeUID_detD_evt2_v??_c??.fits`, and the file names and numbers of files in Table 2 are modified accordingly.)

Table 2: *ixpeproduct*-Generated File Naming Convention

File Name Syntax[A]	No. of Files	Description
ixpe[UID]_[D].arf	3	Ancillary Response File for DU [D]
ixpe[UID]_[D].mrf	3	Modulation Response File for DU [D]
ixpe_d[D]_[date]_[v].rmf[B]	3	Redistribution Matrix File for DU [D]
ixpe[UID]_DU[D]_[S].pi	3×3	Source Stokes [S] spectrum for DU [D]
ixpe[UID]_DU[D]_bg_[S].pi	3×3	Background Stokes [S] spectrum for DU [D]
ixpe[UID]_lc_DU[D][S].fits	3×3	Stokes [S] light curve for DU [D]
ixpexspec.xcm	1	*XSPEC* initialization command file
ixpexspec_py.xcm	1	*PyXspec* initialization command file
ixpe[UID]_polarization.fits	1	*ixpepolarization* output FITS file
region_ixpe[UID]_polarization.reg	1	*ixpepolarization* output region file

[A]where [UID]=observation number; [D]=DU number 1, 2, or 3; [S]=Stokes spectrum I, Q, or U
[B]This is the CALDB syntax where, e.g., [date]=YYYMMDD and version number [v]=v01.

2.8 *ixpemkxspec*

Example command line usage:

```
> ixpemkxspec specfiles=@filelist.txt
```

ixpemkxspec generates an *XSPEC* command file for a list of IXPE Stokes spectral FITS files (I, Q, and U) specified by the `specfiles=` parameter. In the standard FTOOLS parameter value specification, the file list can either be a comma-separated list at the command line or be given, one per line, in a file named, e.g., `filelist.txt` and specified using the syntax `specfiles=@filelist.txt`. *ixpemkxspec* searches all input files for the keywords `DETNAM`, `STOKESPR`, `STOKESWT`, and `DATE-OBS` to determine the DU, the Stokes spectral type (I, Q, or U), the weighting factor (`UNWEIGHTED`, `NEFF`, or `SIMPLE`), and the observation date so as to determine the correct corresponding CALDB Redistribution Matrix File (RMF). *ixpemkxspec* generates both an *XSPEC* and a *pyXSPEC* `.xcm` command file specified by `xcmname=`, (default: `=ixpexspec.xcm` and `=ixpexspec_py.xcm`, respectively). The command file groups the spectra by DU resulting, typically, in three groups for the three DUs each group containing the three spectral files corresponding to the I, Q, and U spectra. In addition, if the keyword `RESPFILE` is not present in the input spectral file(s) or `caldbresp=yes` (the default), then its value is determined from the values of the `DETNAM`, `STOKESWT`, and `DATE-OBS` keywords and the contents of the CALDB and this information is specified in the *XSPEC* command file as a `response` command. Note *ixpemkxspec* does not assign background spectra nor Ancillary or Modulation Response Files to source spectral files.

2.9 *ixpeversion*

Usage:

```
> ixpeversion
```

ixpeversion reports the IXPE software installation version. There is no *fhelp* entry for this task.

3 IXPE Level 1 to Level 2 Data Processing Tasks

The IXPE Science Operations Center (SOC) receives raw telemetry packets from the Mission Operations Center (MOC) at the Laboratory for Atmospheric and Space Physics (LASP) of the University of Colorado (under contract to BAE, formerly Ball Aerospace, the IXPE spacecraft manufacturer). The SOC converts these telemetry streams into FITS files for each observation or observation segment. These files include both science data and spacecraft housekeeping data. The SOC then executes a processing pipeline that constructs Level 1 FITS files containing the time of arrival, location in the detector, pulse height amplitude, and related information of each detected event and also stores an image of the photoelectron track produced by each event registered during the observation. The pipeline also produces several auxiliary files containing calibration information obtained during the observation, files documenting the detector's charging history, and exposure maps. The pipeline then applies various instrument gain corrections, calculates weighted Stokes parameters for each event (adjusted for spurious modulation), applies pointing corrections and an aspect solution to convert event detector locations to celestial sky positions, and recomputes good time intervals (GTIs) among other data refinements and data cleaning to produce a cleaned and calibrated Level 2 FITS event list for each of the three detector units. Observations that are segmented will have as many Level 1 files per detector as there are segments but, nominally, only a single resultant Level 2 file per detector.

Users can reconstruct near-exact duplicate Level 2 files from Level 1 files, several of the housekeeping and auxiliary files, and the IXPE mission CALDB using the suite of IXPE-specific FTOOLS tasks and some general-purpose FTOOLS utilities as illustrated in Figures 3 and 4 and listed in Table 3. The processing sequence must be performed as shown here because of many data dependencies among the tasks. Of course there is no need to reproduce the official pipeline products since they are openly retrievable from the IXPE archive maintained at the HEASARC. However, advanced users may wish to modify some of the parameter values used in the pipeline process and/or use non-CALDB versions of files such as pedestal maps or pixel equalization maps, or, perhaps, one may wish to replace the moments analysis formalism used by the pipeline task *ixpeevtrecon* to determine Stokes parameters and initial photoelectron direction and location.

The basic function of each of the tasks listed in Table 3 is provided below beginning with a command-line example explicitly listing key parameters and their values. Note that all tasks require an input FITS file that begins with a Level 1 event list (input to *ixpeevtrecon*) and builds from there from task to task.

Much of the instrument-related IXPE software and CALDB were produced by the Space Science Data Center (SSDC) of the Agenzia Spaziale Italiana (ASI) with substantial contributions by the IXPE Instrument Team at Istituto Nazionale di Astrofisica (INAF) and at Istituto Nazionale di Fisica Nucleare (INFN). The X-ray optics CALDB, the aspect solution software, and the IXPE production pipeline were produced by the SOC.

Table 3: IXPE Level 1 to Level 2 Data Processing FTOOLS tasks.

ixpeevtrecon	Extracts time, position, PHA, and initial photoelectron direction from event images
ixpeinterptemp	Interpolates over temporal gaps in Level 1 housekeeping temperature data
ixpegaincorrtemp	Corrects PHA values for time-dependent temperature effects on detector gain
ixpechgcorr	Corrects PHA values for time-dependent charging effects on detector gain
ixpegaincorrpkmap	Corrects PHA values for high voltage settings and applies gain corrections derived from onboard calibration source measurements made during the observation
ixpecalcstokes	Calculates modulation Stokes parameters
ixpeadjmod	Removes spatial- and energy-dependent spurious modulation factors
ixpeweights	Calculates weighting factors from the moments analysis of event images
ixpedet2j2000	Transforms event positions and Stokes parameters from detector to sky coordinates
ixpeboomdriftcorr	Corrects for residual aspect errors due to periodic distortions in the telescope boom
ixpeflagbgd	Flags background events in a Level 1 events file
ixpeaspcorr	Corrects for residual aspect errors by interpolating average event positions over time
ixpemkevt2gti	Creates Level 2 GTI HDUs from valid event, housekeeping, and attitude time intervals

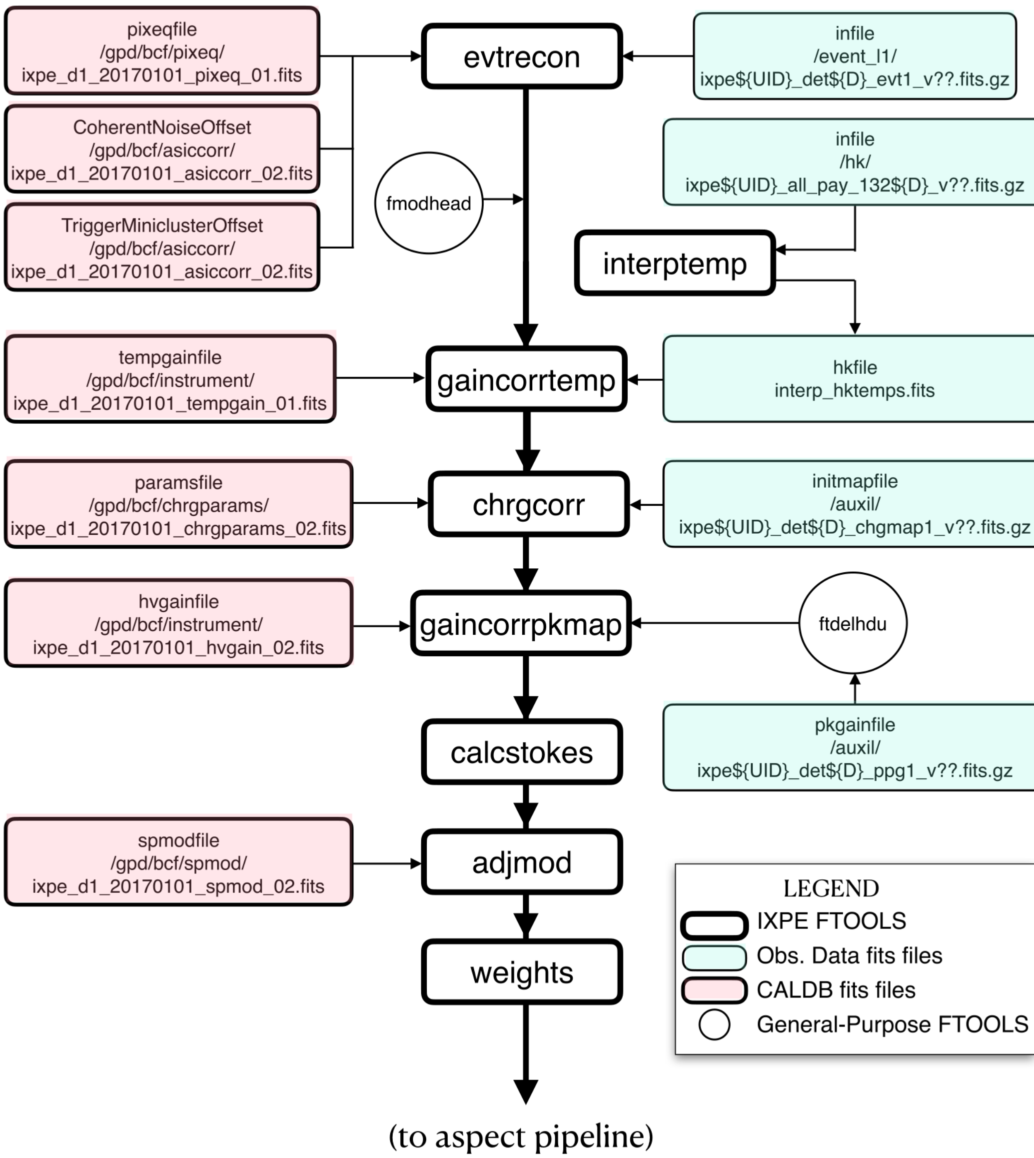

Figure 3: Flowchart describing the instrument portion of the IXPE data processing pipeline. Mission-specific IXPE FTOOLS are denoted with thick outline (mostly in center of the chart), observation specific data files are denoted with thin outline (mostly on the right hand side), and static IXPE calibration database (CALDB) data files are denoted with medium-thickness outline (left hand side). General purpose FTOOL utilities are denoted with circular outlines.

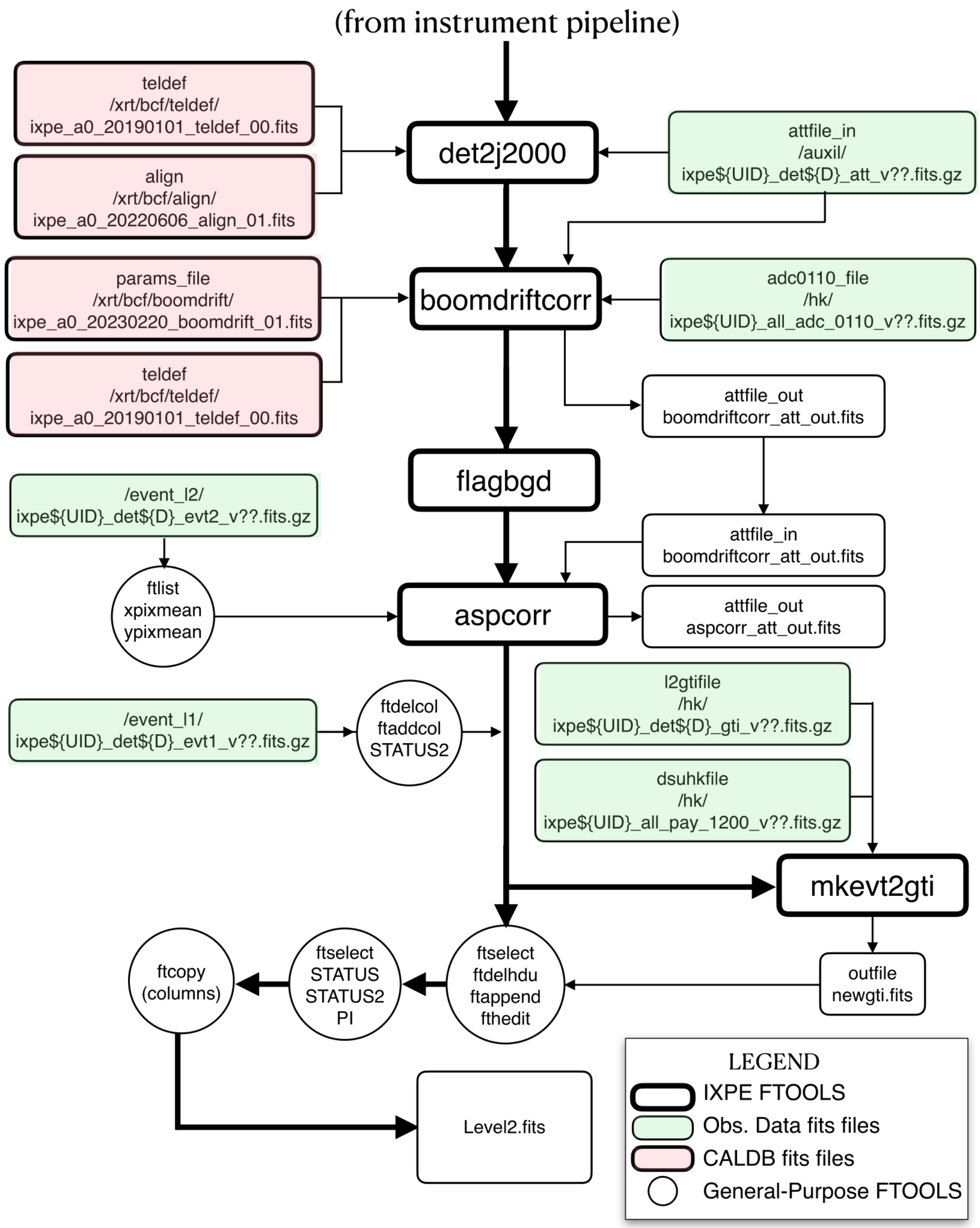

Figure 4: Flowchart describing the aspect portion of the IXPE data processing pipeline. Mission-specific IXPE FTOOLS are denoted with thick outline (mostly in center of the chart), observation specific data files are denoted with thin outline (mostly on the right hand side), and static IXPE calibration database (CALDB) data files are denoted with medium-thickness outline (left hand side). General purpose FTOOL utilities are denoted with circular outlines.

3.1 *ixpeevtrecon*

Example command line usage:

```
> ixpeevtrecon infile=./event_l1/ixpe05005001_det2_evt2_v01.fits.gz \
>     outfile=evtrecon_out.fits
```

The *ixpeevtrecon* task derives the X-ray event absorption point and initial photoelectron direction (both in detector coordinates) and the event energy (uncalibrated PHA value) from raw ionization track image data for each event in the Level 1 event list file specified by the `infile=` parameter value.

IXPE detectors are similar to classic proportional counters in that they use a gas medium to absorb incident X-rays and an applied voltage to cause the resulting electron charge cloud to drift toward an anode readout. However, IXPE's novel design includes a Gas Electron Multiplier (GEM), to provide charge gain while preserving the track shape, and a finely pixellated anode (a custom ASIC readout) to *image* the photoelectron track (Fig. 5) in the readout plane perpendicular to the incident X-ray direction. The shape of, and charge distribution in, these photoelectron tracks are analyzed by the *ixpeevtrecon* task to derive the total collected charge (and hence the X-ray event energy), absorption point, and initial photoelectron direction (and hence a measure of the incident X-ray electric field direction).

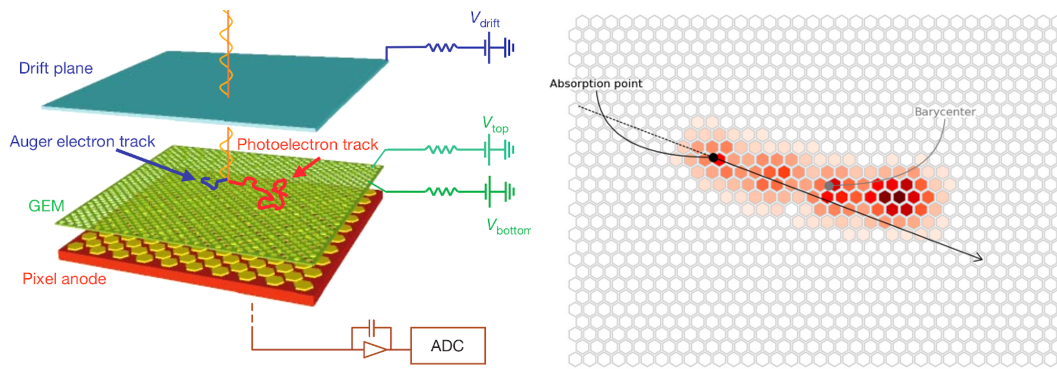

Figure 5: *Left:* Schematic describing how IXPE images the track of a photoelectrically-absorbed photon. Ideally, an incident X-ray is absorbed in the region between the entrance window and the GEM top. The ionization track generated by the photoelectron drifts toward the GEM where the charge is amplified while the shape is preserved and collected on the finely pixelated readout ASIC. (from Ref. [5]). *Right:* Example image of a photoelectron track. Note the hexagonal pixels.

There are three primary functions performed by *ixpeevtrecon*: (1) a calibration and zero suppression adjustment of the charge in the pixels in and around the track, (2) an analysis of the charge morphology in the track and in surrounding pixels in order to separate the true contiguous ionization trail from residual noise pixels and determine the total charge collected, and (3) an analysis of the track topology to determine the X-ray absorption point and initial photoelectron direction. See the GPDWorkbook, Ref. [5], for detailed information on the IXPE detectors, and onboard data extraction and processing.

Pixel Charge Adjustments Following an event trigger, IXPE onboard processing identifies a rectangular subset of the full pixel array, called the region of interest (ROI) that contains the event then performs a serial readout of the event signal in the ROI followed by a readout of the pedestal; the equivalent of a dark field in a CCD. The pedestal is subtracted from the signal. Finally, a fixed-threshold zero suppression is performed where pixels in the ROI whose pedestal-subtracted signal is at or below a given value (set to 18 on board, a value roughly 2σ above the mean noise level[5]) are discarded to save telemetry bandwidth. The *ixpeevtrecon* parameter `threshold=` (default: `=20` can be set to raise the per pixel ADC threshold (e.g., if background noise is higher than normal).

[5]Following the DU2 anomaly, the zero suppression threshold for DU2 was set to 26. This value is stored in the event list header keyword `ZSUPTHR` beginning at Level 1.

ixpeevtrecon also applies a gain adjustment that corrects for large-scale non-uniformities imprinted in the manufacturing process of the GEM that amplifies the charge in the detector and for pixel-to-pixel gain variations in the ASIC. This step uses the pixel equalization file defined by the `pixeqfile=` parameter value (default: `=CALDB`). An additional pedestal subtraction is also applied using the file specified by the `xpolpedestalfile=` (default: `=CALDB`). The CALDB pedestal maps consist of all zeros except for DU2 post-anomaly (14 April 2025).

Clustering *ixpeevtrecon* then applies an algorithm to group pixels in the ROI into clusters based on characteristics of immediately adjacent pixels (with signal above threshold). By design, clusters define the track of charge deposited by the primary photoelectron cloud and thus are the basis for determining the event location, energy, and the initial direction of the photoelectron.

ixpeevtrecon applies the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to perform the clustering (Ref. [6]). A cluster is defined as a group of contiguous pixels at least one of which has at least `minDensityPoints=` (default: `=4`) immediately contiguous pixels above threshold. For the hexagonal IXPE pixels, the density around a pixel could be as high as six. A cluster may contain many such pixels and will also contain any pixel that is reachable by these core pixels; that is, any pixel for which a contiguous path of core pixels exists to that pixel is a member of the cluster. This basically defines a cluster: all pixels but those on the edge of the cluster meet the minimum density requirement. Clusters with a pixel count less than or equal to the value of the parameter `minTrackHits=` (default: `=6`) are rejected as noise.

The summation of the charge in a cluster determines the cluster pulse height amplitude. The cluster with the highest charge is assumed to be the true X-ray signal for the event. *ixpeevtrecon* records the (integer) total charge for each event as column `PHA` and the (real) pixel-equalization-adjusted charge as column `PHA_EQ` in the file specified by the `outfile=` parameter value.

Moments Analysis *ixpeevtrecon* applies a two-stage moment analysis of the charge-weighted spatial distribution of the cluster track to determine the incident photon location and the photoelectron initial (emission) direction. In the first stage, the first moment defines the (charge-weighted) barycenter. Maximizing the second moment (the variance of the projected one-dimensional charge distribution) about the barycenter provides the direction of the principal axis of the cluster. The longitudinal and transverse second moments along and perpendicular to this principal axis define an ellipsoid of inertia of the two-dimensional charge distribution and their square roots define the lengths of the semi-major and semi-minor axes of the ellipsoid, respectively. These are stored as columns `TRK_M2L` and `TRK_M2T`, respectively, and are used to define event weights by the task *ixpeweights* (§3.8). The third moment determines the direction toward the incident location along the principal axis by utilizing the fact that the bulk of the charge deposited by the photoelectron is at the terminus of the track (the so-called Bragg peak). *ixpeevtrecon* records the location of the barycenter and angle of the principal axis for the event as columns `BARX`, `BARY`, and `DETPHI1` in the file specified by the `outfile=` parameter value.

In the second stage in the moments analysis, those cluster pixels located in the half-plane extending from the barycenter in the direction of the principal axis away from the Bragg peak (as indicated by the skewness) and in the semi-annulus defined by the `dmin=` and `dmax=` parameter values (defaults: `=1.5` and `=3.5` in units of the longitudinal second moment, respectively) are used to find a new (charge-weighted) mean. Then, weights for all pixels in the cluster are defined by an exponentially-decreasing function of the distance from this point with an e-folding length specified by the `weightScale=` parameter value (default: `=0.05` μm or one pixel). A new (charge- and distance-weighted) barycenter is calculated that is the best estimate of the incident photon interaction point and used as the pivot point for a new (weighted) second moment determination resulting in a principal axis that is the best estimate of the initial photoelectron direction. Fig. 6 shows the moments analysis steps graphically.

ixpeevtrecon records the location of this new barycenter and angle of the principal axis for the event as columns `ABSX`, `ABSY`, and `DETPHI2` in the file specified by the `outfile=` parameter value. Also recorded in the output are several quantities characterizing the region of interest and its location on the detector, the number of pixels above threshold, the final longitudinal and transverse second moments, among other quantities.

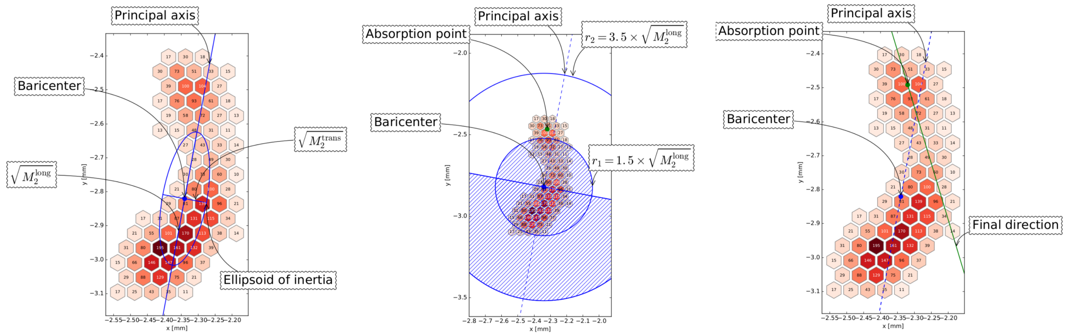

Figure 6: Sequence showing the steps in the moments analysis of a photoelectron ionization track. *Left:* In the first step, the first and second moments define the barycenter, the longitudinal second moment (M_2^{long}), and the direction of the principal axis (ϕ such that $\partial M_2(\phi)/\partial\phi = 0$ about the barycenter). *Center:* The charge-weighted mean of the subregion defined as the semi-annulus extending from `dmin=` to `dmax=` from the barycenter in the half-plane towards the incident photon location is computed. *Right:* In the second step, weights for all pixels in the cluster are defined as $w = \exp(-d/d_0)$ where d is the physical distance from the charge-weighted mean of the subregion and d_0 is specified by the `weightScale=` parameter value. Then a new, weighted, barycenter and principal axis direction are determined that are the best estimates of the incident photon interaction point (Absorption point) and of the initial photoelectron direction (Final direction), respectively. Graphics from Ref. [5].

3.2 *ixpeinterptemp*

Example command line usage:

```
> ixpeinterptemp infile=./hk/ixpe05005001_all_pay_1322_v01.fits.gz \
>     outfile=interp_hktemps.fits
```

ixpeinterptemp performs a linear interpolation of the temperature values in the spacecraft housekeeping file specified by `infile=/hk/ixpeUID_all_pay_132D_v??.fits.gz`, where `D` = 1, 2, or 3 for DU1, DU2, and DU3, respectively, over time gaps less than a given maximum duration specified by `maxgap=` (default: `=150` s). Interpolated data are inserted into the output file at the nominal housekeeping cadence of 1 Hz. The output of *ixpeinterptemp* is a FITS file containing the original temperature data and the interpolated temperatures where needed. The purpose of *ixpeinterptemp* is to ensure adequate time resolution of the instrument temperature is available for making temperature-dependent gain corrections using *ixpegaincorrtemp*, §3.3.

3.3 *ixpegaincorrtemp*

Example command line usage:

```
> ixpegaincorrtemp infile=evtrecon_out.fits outfile=gaincorrtemp_out.fits \
>     hkfile=interp_hktemps.fits
```

ixpegaincorrtemp corrects the measured gain for effects due to variations in the detector temperature. For each event in the list specified by `infile=`, *ixpegaincorrtemp* linearly interpolates the detector temperature from the temperature history contained in the output from *ixpeinterptemp* (see §3.2), specified as input parameter `hkfile=`, and applies a linear relation between the interpolated temperature and the detector gain using coefficients given in a file specified by parameter `tempgainfile=` (default: `=CALDB`) to obtain the temperature-corrected PHA value for the event. These values are placed in the output event list (`outfile=`) as a new column, `PHA_T`. The coefficients in the Calibration Database file were derived from ground calibration measurements of the individual IXPE detector units.

3.4 *ixpechrgcorr*

Example command line usage:

```
> ixpechrgcorr infile=gaincorrtemp_out.fits outfile=chrgcorr_out.fits \
>     initmapfile=./auxil/ixpe05005001_det2_chgmap1_v01.fits.gz
```

ixpechrgcorr corrects the gain for effects caused by the build-up of charge within the Gas Electron Multiplier (GEM) and its slow discharge over time. The GEM provides the gas multiplication of the primary ionization which is necessary to generate a detectable electrical signal in the readout ASIC. However, some of the charge deposits in the dielectric material of the GEM during times of high event rate and is only slowly dissipated during times of low event rate.

The charging and discharging is a continuous process. Therefore, *ixpechrgcorr* requires an initial map of the charge state, `initmapfile=./auxil/ixpeUID_detD_chgmap1_v??.fits`, which describes the charge status of the detector at the beginning of the observation. *ixpechrgcorr* compiles the event rate history of the observation at a time resolution specified by `timebinsize=` (default: `=30` s) then applies a rate-dependent gain correction using model coefficients given in a file specified by parameter `paramsfile=` (default: `=CALDB`). For more details on the charging effect and the gain correction model see Ref. [7] and Chapter 12 of Ref. [5]. The *ixpechrgcorr* correction should be applied to `phacol=PHA_T` (see §3.3). Corrected values are placed in the output event list (`outfile=`) as a new column, `PHA_CHG`.

3.5 *ixpegaincorrpkmap*

Example command line usage:

```
> ixpegaincorrpkmap infile=chrgcorr_out.fits outfile=gaincorrpkmap_out.fits \
>     pkgainfile=./auxil/ixpe05005001_det2_ppg1_v01.fits.gz
```

ixpegaincorrpkmap calculates the final Pulse Invariant (PI) values of IXPE events. *ixpegaincorrpkmap* applies two corrections to the `PHA_CHG` values of each event in the `infile=` event list. One correction is a gain scaling correction for the high voltage setting of the GEM using parameters in the file specified by `hvgainfile=` (default: `=CALDB`) and either the GEM voltage specified by the keyword `S_VGEM` contained in the input event list (`infile=`) header if `dvgemcaldb=no` (the default) or, if `dvgemcaldb=yes`, specified by the Calibration Database file given by the `hvgainfile=` parameter. The `S_VGEM` keyword should be used as it reflects the value of the high voltage setting of the GEM at the time of the observation.

The other correction is the final gain-to-energy scaling. It is calculated using information stored in peak gain map calibration files specified by `pkgainfile=./auxil/ixpeUID_detD_ppg1_v??.fits`. This is a required parameter. The `pkgainfile=` input file contains a spatial grid of linear coefficients, in detector coordinates, describing the relationship between the measured Mn K_α emission line and the Si K_α fluorescence line and their true energies. The `pkgainfile=` map may contain multiple FITS extensions; one for each set of calibration measurements performed during or slightly before the science observation. In this case, *ixpegaincorrpkmap* interpolates between the maps appropriate to the event arrival time. The final values are placed in the output event list (`outfile=`) as a new column, `PI`.

Note: The `pkgainfile=` specification is a required parameter. However, pipeline processing of roughly 25 of the earliest IXPE observations, those prior to early Jun 2022, did not generate the required auxiliary file. *ixpegaincorrpkap* will fail to process data from those observations at this time. When IXPE data is reprocessed, scheduled for the Fall of 2026, these auxiliary files will be created and available from the HEASARC archive.

ixpegaincorrpkmap operates on the `PHA_CHG` column of the input event list. Therefore, *ixpegaincorrpkmap* should only be invoked with `infile=` set to the output event list from *ixpechrgcorr*, §3.4. See Ref. [8] for details of the on-board calibration sources and their usage.

3.6 *ixpecalcstokes*

Example command line usage:

```
> ixpecalcstokes infile=gaincorrpkmap_out.fits outfile=calcstokes_out.fits
```

ixpecalcstokes calculates the Stokes Q and U parameter values of IXPE events in detector coordinates. *ixpecalcstokes* first copies columns `DETPHI2`, `ABSX` and `ABSY` , calculated in the second pass of the moment analysis by *ixpeevtrecon* (§3.1), to new columns `DETPHI`, `DETX`, and `DETY`, respectively. Then, the Stokes parameters are defined as new columns `DETQ` and `DETU` defined as

$$\begin{aligned} \mathtt{DETQ} &= 2\cos(2 * \mathtt{DETPHI}) \\ \mathtt{DETU} &= 2\sin(2 * \mathtt{DETPHI}) \end{aligned} \tag{1}$$

3.7 *ixpeadjmod*

Example command line usage:

```
> ixpeadjmod infile=calcstokes_out.fits outfile=adjmod_out.fits
```

ixpeadjmod corrects the Stokes parameters in detector coordinates for spurious modulation. Spurious modulation is the instrumental response to unpolarized radiation and is a function of photon energy and location on the detector. *ixpeadjmod* applies spatial and (linearly-interpolated) energy correction coefficients stored in the spurious modulation calibration file specified by `spmodfile=` (default: `=CALDB`) to each event in the `infile=` event list. The algorithm is applied to the values of the `DETQ` and `DETU` columns as

$$\begin{aligned} \mathtt{DETQ} &= \mathtt{DETQ}_0 - q_{sm}[\mathtt{DETX}, \mathtt{DETY}, \mathtt{PI}] \quad \text{and} \\ \mathtt{DETU} &= \mathtt{DETU}_0 - u_{sm}[\mathtt{DETX}, \mathtt{DETY}, \mathtt{PI}] \end{aligned} \tag{2}$$

where the original values are denoted with the subscript 0 and the spurious modulation correction values are denoted q_{sm} and u_{sm} with the dependence on detector location and event energy shown explicitly.

Note that, unlike Eqn. 1, the quantities `DETQ` and `DETU` are no longer properly normalized, i.e., in general, $\mathtt{DETQ}^2 + \mathtt{DETU}^2 \neq 4$ and cannot be easily transformed back into a polarization position angle. However, it has been shown that the correct angle is provided, on average, for an ensemble of events. For more details on the spurious modulation calibration and correction see Ref. [9].

3.8 *ixpeweights*

Example command line usage:

```
> ixpeweights infile=adjmod_out.fits outfile=weights_out.fits
```

ixpeweights calculates event weighting factors from quantities obtained from the event track reconstruction moments analysis. Specifically, the task calculates the ellipticity, α, of each event defined by $\alpha = ((\mathtt{TRK\_M2L} - \mathtt{TRK\_M2T})/(\mathtt{TRK\_M2L} + \mathtt{TRK\_M2T}))$ where `TRK_M2L` is the longitudinal second moment (track length) and `TRK_M2T` is the transverse second moment (track width) associated with the event as calculated by *ixpeevtrecon* (§3.1). The weight is then defined as α^λ where the optimal value of λ was determined by Monte Carlo simulation to be $\lambda = 0.75$.

Computed weight values are placed in the output event list (`outfile=`) as a new column, `W_MOM`. For more details on the weighting analysis see Ref. [10].

3.9 *ixpedet2j2000*

Example command line usage:

```
> ixpedet2j2000 infile=weights_out.fits outfile=det2j2000_out.fits \
>     attfile_in=ixpe05005001_det2_att_v01.fits
```

ixpedet2j2000 transforms IXPE event positions and modulation Stokes parameters from detector coordinates to sky coordinates. *ixpedet2j2000* uses the event time stamp from the `infile=` event list and the rotation quaternions from the attitude housekeeping file specified by the `attfile_in=` parameter value to perform this transformation. The quaternions are calculated by combining time-independent quaternions representing the transformation from detector frame to star tracker frame with the time-varying quaternions produced by the star tracker that rotate from the star tracker frame to the J2000 sky frame. The latter quaternions are generated at a rate of 10 Hz. The transformation quaternion is found by linearly interpolating between the two attitude solutions that bracket the event time component-by-component.

The resulting sky coordinates are added as new columns `X` and `Y` and the transformed Stokes parameters are added as new columns `Q` and `U` to the `outfile=` event list along with World Coordinate System keywords defining the tangent plane reference point coordinates, array locations, and increments, etc. The `(X,Y)` axes lie along lines of constant Declination and Right Ascension, respectively, at the tangent plane location and $\mathtt{Q} \equiv 2\cos(2\psi)$ and $\mathtt{U} \equiv 2\sin(2\psi)$ where ψ is the polarization position angle measured, as per IAU convention, counterclockwise from celestial North.

3.10 *ixpeboomdriftcorr*

Example command line usage:

```
> ixpeboomdriftcorr infile=det2j2000_out.fits outfile=boomdriftcorr_out.fits \
>     adc0110_file=./hk/ixpe05005001_all_adc_0110_v01.fits.gz \
>     attfile_in=ixpe05005001_det2_att_v01.fits attfile_out=boomdriftcorr_att_out.fits
```

ixpeboomdriftcorr corrects event sky coordinates for residual aspect errors caused by thermal distortions in the telescope boom. The IXPE spacecraft experiences significant temperature variations due to changing sunlight exposure as the spacecraft orbits Earth. *ixpeboomdriftcorr* applies a model of the resulting periodic boom distortions to correct each event's sky coordinates using model parameters and telescope geometry parameters contained in CALDB files. The task correlates event timestamps from the `infile=` event list with the trajectory of the spacecraft as given by the orbital state vector history in the `adc0110_path=` Level 1 housekeeping file and the spacecraft orientation as given by the field rotation history in the `attfile_in=` attitude file. In addition, *ixpeboomdriftcorr* calculates the interpolated position of the spacecraft pointing direction on the sky plane and adds these values as a new column, `TXYDZ_CORRBD`, to the output attitude file `attfile_out=`.

3.11 *ixpeflagbgd*

Example command line usage:

```
> ixpeflagbgd infile=boomdriftcorr_out.fits outfile=flagbgd_out.fits \
```

ixpeflagbgd applies an instrumental background identification algorithm to events in the `infile=` input event list and records background events by setting the eighth bit (0x0100) of the `STATUS` column value for the event. Background events are retained in the IXPE SOC production pipeline final product Level 2 event lists but can be removed using the FTOOLS utility *ftselect* by setting `expression=STATUS.eq.b0`. The algorithm uses parameter values stored in CALDB files.

ixpeflagbgd uses three discriminators to identify background events. These are (a) the number of triggered pixels, `NUM_PIX`, in an event track as a function of event energy, (b) the fraction of the event energy, `EVT_FRA`, that is contained in the main pixel cluster as a function of energy, and (c) the number of triggered pixels on the edge of the event track, `TRK_BORD`. These quantities are derived by *ixpeeventrecon* processing, §3.1, and stored in data columns for each event but these columns are currently removed from the final Level 2 event lists. By default, all three discriminators are applied by *ixpeflagbgd* but the parameters `use_num_pix=`, `use_efrac=`, and `use_border_pix=` can be reset (default: `=yes`). For details of the background identification and the rejection algorithm see Ref. [4].

3.12 *ixpeaspcorr*

Example command line usage:

```
> ixpeaspcorr infile=flagbgd_out.fits outfile=aspcorr_out.fits \
>     attfile_in=boomdriftcorr_att_out.fits attfile_out=aspcorr_att_out.fits
```

ixpeaspcorr corrects for residual aspect errors using interpolated event position averages. These residual errors tend to cause small but measurable broadening of the distribution of events for point-like X-ray sources. The *ixpeaspcorr* algorithm is based on the assumption of a point source so that the majority of events are expected to be localized. Thus, it is recommended that *ixpeaspcorr* not be used on observations of very dim or extended sources (the production pipeline follows this recommendation; if *ixpeaspcorr* was applied, then the keyword `XPASPCOR=T` is stored in the ł2 event list HDU for the observation). If instrumental background events have been identified (§3.11), then they are omitted from the *ixpeaspcorr* computation.

The task correlates event timestamps from the `infile=` event list with the pointing history contained in the attitude file `attfile_in=`. Furthermore, the attitude file is used to identify which one of the two (fore- or aft-facing) optical star tracker heads is being used in each 0.1 s time interval of aspect solution. The *ixpeaspcorr* event position averaging treats events received from the fore- and aft-facing heads independently, which significantly reduces position errors near times of camera transitioning. The time intervals over which the event position averaging is computed is set by the hidden parameter `n=` (default: `=300` s). If the hidden parameters `x_pix_mean=` and/or `y_pix_mean=` are given, these values will be used instead of the mean of all events. It is recommended that the values of these parameters be specified as given in the original (production pipeline) Level 2 event lists by the keywords `XPIXMEAN` and `YPIXMEAN`.[6] The FTOOLS utility *ftlist* can be used to obtain these values, e.g.,

```
> ftlist ixpe05005001_det2_evt2_v01.fits[1] K include='*pixmean'
```

In addition, the interpolated position of the spacecraft pointing direction on the sky plane will be added as a new column, `TXYDZ_CORR`, to the attitude file.

3.13 *ixpemkevt2gti*

Example command line usage:

```
> ixpemkevt2gti infile=aspcorr_out.fits outfile=newgti.fits \
>     l2gtifile=./hk/ixpe05005001_det2_gti_v01.fits.gz \
>     attfile_in=ixpe05005001_det2_att_v01.fits  \
>     dsuhkfile=./hk/ixpe05005001_all_pay_1200_v01.fits.gz
```

ixpemkevt2gti finds the intersection of the Good Time Interval (GTI) information from an unfiltered intermediate event list `infile=`, the GTI information from the housekeeping-defined Level 2 GTI file `l2gtifile=`, and the valid time intervals it derives from the Level 1 attitude file `attfile_in=` to derive the final GTI for the Level 2 event file, `outfile=`. The unfiltered events are used to calculate the keyword values `LIVETIME`, `ONTIME`, and `DEADC` which are added to the output GTI. In addition, the Detector Service Unit (DSU) housekeeping file, `dsuhkfile=`, is used with the derived GTI data to calculate the filter calibration wheel potentiometer setting and update the `FCW_ENC` header keyword value.

The output file contains a primary HDU with all the Level 2 event file primary header keywords, and a GTI binary table extension which contains the GTIs that are to be applied and appended to the event list using general-purpose FTOOLS tasks:

```
> ftselect aspcorr_out.fits tmpfile.fits 'gtifilter("newgti.fits")
> ftdelhdu 'tmpfile.fits[GTI]' mkevt2gti_out.fits
> ftappend newgti.fits[GTI] mkevt2gti_out.fits
> rm tmpfile.fits
```

[6] These keywords are not present in Level 2 files processed early in the mission.

The first command limits events to only those within valid good time intervals. The second two commands delete the original GTI header/data unit (HDU) extension and appends the new GTI extension to the event list file. The updated keywords should also be added (note the use of the asterisk wildcard):

```
> ftlist newgti.fits[1] K include='DEADC,*TIME,TST*,TELAPSE,DATE-*' \
>     outfile=fix.keys clobber=yes
> fthedit mkevt2gti_out.fits[0] @fix.keys
> fthedit mkevt2gti_out.fits[1] @fix.keys
> fthedit mkevt2gti_out.fits[2] @fix.keys
> rm fix.keys
```

3.14 Final Level 2 File Generation: *ixpepipe*

After all the FTOOLS tasks described in this section, §3, have been applied beginning with a Level 1 FITS event list, it remains to apply some straightforward filtering and removal of certain columns to create a Level 2 FITS event list comparable to the SOC production pipeline output in both content and quality. This is accomplished with general-purpose FTOOLS tasks:

```
> ftselect mkevt2gti_out.fits tmpfile.fits \
>     status.EQ.b0000000x00000000.AND.status2.EQ.b0x0000000000x00x.AND.pi.gt.0.AND.pi.le.374
> ftcopy \
>     'tmpfile.fits[EVENTS][COL TRG_ID; TIME; STATUS; STATUS2; PI; W_MOM; X; Y; Q; U]' \
>     level2_05005001_det2.fits
> rm tmpfile.fits
```

The first command removes events with bad status flags and those outside the valid energy range. The second command removes data columns not needed for spectro-polarimetric data analysis.

A user-contributed `bash` shell script, *ixpepipe*, is available from the IXPE contributed software site that performs all the pipeline steps described above to produce a Level 2 file given the necessary inputs beginning with Level 1 event list files.

4 References

[1] IXPE Quick Start guide ixpe_quickstart.pdf

[2] Baldini, L. et al. 2022, SoftwareX 19

[3] Kislat, F., et al. 2015, AstropartPhys, 68, 45

[4] Di Marco, A., et al. 2023, AJ, 165, 143

[5] Gas Pixel Detector Workbook gpdworkbook_0.11.0.pdf

[6] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. 1996, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), 226

[7] Baldini, L., et al. 2021, Astroparticle Physics, 133, 102628

[8] Ferrazzoli, R., et al. 2020, JATIS 6, 048002

[9] Rankin, J., et al. 2022, AJ, 163, 39

[10] Di Marco, A., et al. 2022, AJ, 163, 170

5 Acronyms

ARF	Ancillary Response File
ASIC	Application Specific Integrated Circuit
CALDB	CALibration Database
DS9	SAOImageDS9 data visualization application
DSU	Detector Service Unit
DU	Detector Unit
FITS	Flexible Image Transport System
FOV	Field Of View
FTOOLS	Software tools to manipulate FITS files
GEM	Gas Electron Multiplier
GPD	Gas Pixel Detector
GTI	Good Time Interval
HEASARC	High Energy Astrophysics Science Archive Research Center
HDU	FITS Header/Data Unit
IXPE	Imaging X-ray Polarimetry Explorer
IRF	Instrument Response File
MRF	Modulation Response File
OGIP	Office for Guest Investigators Programs
PHA	Pulse Height Analysis
PI	Pulse Invariant (event energy)
RMF	Redistribution Matrix File
SOC	Science Operations Center
XRT	X-ray Telescope