# SIXTE: Docoding the XGIS mask

Philipp Thalhammer

June 2025

## 1 Introduction

The upcoming THESEUS satellite will be equipped with two main high-energy X-ray and $\gamma$-ray detectors. One of which is the coded mask instrument *xgis*. It is sensitive to both X-ray and $\gamma$-rays in the 2 keV–10 MeV range. It consists of a coded mask and a silicon drift detector layer. The most important parameters of the coded-mask and detector geometry are listed in Table 1.

Table 1: Relevant geometrical parameters of the *xgis* detector

| Mask | |
|---|---|
| Width | 56.1 cm |
| Distance | 63.0 cm |
| Pixel width | 1 mm |
| Pixel dimensions | 55 |
| Thickness | 1 mm |
| Material | Tungsten (W) |
| **Detector** | |
| Width | 45 cm |
| Pixel width | 0.5 mm |

This document explains the deconstruction algorithm implemented as part of *SIXTE* to recover the sky image from the shadow patter on the detector plane. The layout of the *xgis* detecor as used by *SIXTE* is shown in Fig. 1. It comprises $10 \times 10$ individual detector modules with $8 \times 8$ pixels each, with a 1 mm gap between individual modules.

## 2 The Projection:

To project a single photon form a position in the sky onto the detector the shadogram is shifted by

$$\delta x = d_{\mathrm{mask}} \tan(\delta\phi) \,. \tag{1}$$

Then a random x-/y-position on the mask is sampled and if his position corresponds to an open position on the mask, the impact position on the detector is

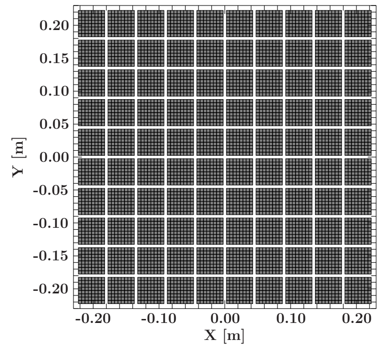


Figure 1: The layout of the *xgis* detector as used by *SIXTE*.

forwarded to the following modules of *SIXTE*. The photon energies are being sampled from the on-axis ARFs and RMFs, while the off-axis effects generated from the mask effects are being accounted for via vignetting. The detector readout is handled via the respective *SIXTE* module for a SDD detector.

Effects not included in the photon projections so far include:

- Scattering of photons within the mask material
- Abortion effects that depend on the thickness of the mask
- The scattering of photons within the detectors.

# 3 The Deprojection:

This results in each source throwing a shadow pattern on the detector - the so called shadowgram. Deconstruction of the original sky pattern requires to calculate the cross-correlation between the detector image and the shifted mask pattern for each position in the sky. This can be calculated by summation of element wise matrix multiplication.

$$S = D * G = \sum_{ij}^{N_{\mathrm{pix}}} D_{k,l} G_{i+k,j+l} \tag{2}$$

Here the matrix $G$ corresponds to the mask pattern rescaled to provide a correctly normalized devolution that is one average zero for an empty region of the sky. A simple way to achieve this is by defining $G = 2M - 1$.

It is, however, much more computationally efficient to calculate the convolution via a multiplication in Fourier space.

$$S = \mathcal{F}^{-1}\left(\mathcal{F}\left(M\right) \cdot \mathcal{F}\left(D\right)\right) \tag{3}$$

# 4 Implementation:

The implementation of the deconstruction algorithm is done in Python using common libraries such as the `numpy` and `scipy` libraries and can be called via *iros.py*.

# 5 Example:

In the following we illustrate an example run for both the Projection and the deprojection. Fist a singe off-axis source is defined in a respective simput file. The summation is run via *sixtesim*:

```
$SIXTE/bin/sixtesim \
  XMLFile=${xmldir}/xgis-x.xml \
  RA=0.0 Dec=0.0 \
  Exposure=1000 \
  Simput="${simputdir}/single_source.fits" \
  Prefix="output/" \
  EvtFile=test.fits \
  Background=y \
  clobber=y \
```

As the detector array of *xgis* consists of $10 \times 10$ individual detector arrays, *SIXTE* will provide an event-file for each individual detector. As a user this is likely inconvenient, and it is therefore recommended merging individual event-files by running a script similar to the following:

```
ls    test*.fits > evts.lst
fmerge  infiles=@evts.lst  \
        outfile=test_merged.fits columns=-
```

The merged event-file can then be used for the deconvolution and reconstruction of the sky image. For this a pythpn3 script can be provided, which requires the source position for the generation of spectra and light-curves and prefix for the output files names. Default ARF and RMF paths are assumed by

```
./deconvolve.py
    --ra_src 250 --dec_src -50
    --outfile "test"
    --infile "test_merged.fits"
```

Running *SIXTE* results in the following detector image:

For a more realistic representation of *xgis*, we simulate a pointing towards the galactic center based on the INTEGRAL reference catalog.

The detector and sky image are shown in the figure below.

The tool deconvolve.py also generates light-curves and spectra for a specific sources position. For this the deconvolution is performed for each spectral

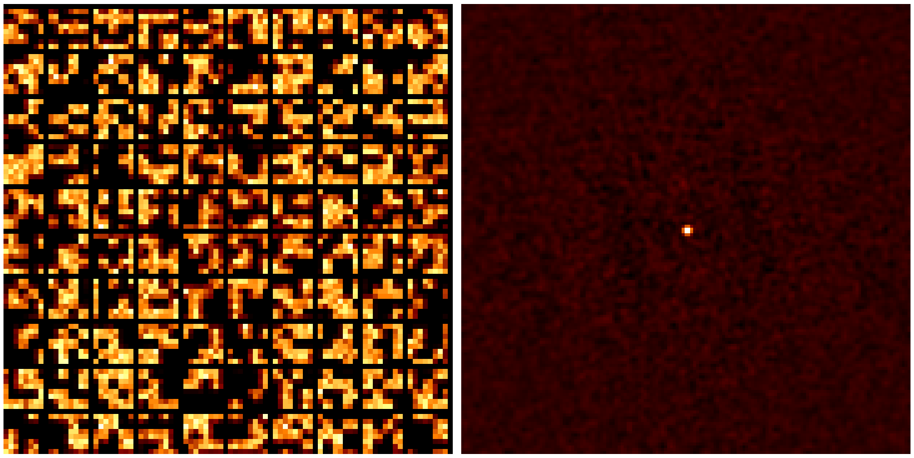

Figure 2: The detector image (left) and the reconstructed sky image (right) for a simulation with a single off-axis source.

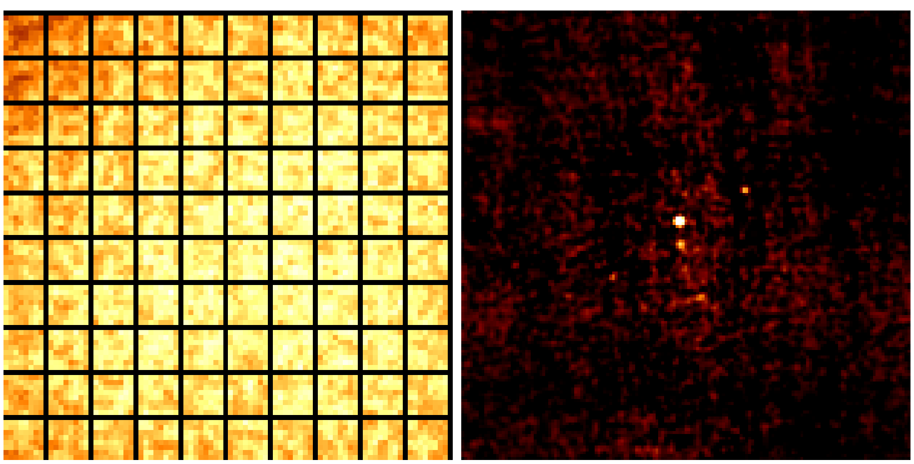

Figure 3: The detector image (left) and the reconstructed sky image (right) for a simulation of the galactic center. The detector image shows the shadow pattern cast by the coded mask, while the sky image shows the reconstructed source distribution.

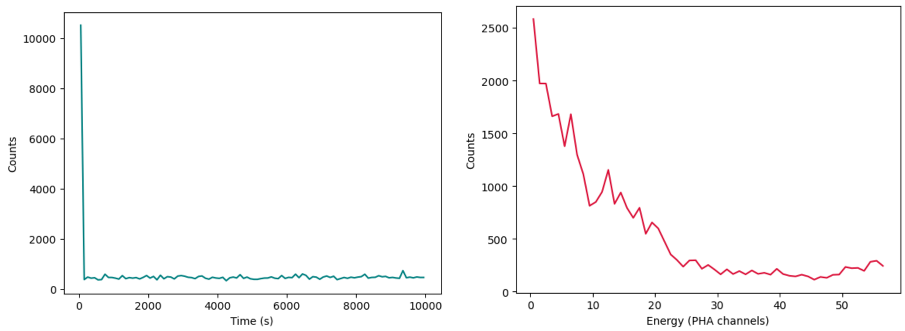


Figure 4: Light-curve and spectrum for a simulated GRB at the very beginning of a 10 ks observation

channel and *nbins* time bins. Appropriate header keywords pointing towards the response files are set, such that the pha-file can be used an any spectral fitting software, such as ISIS or XSPEC. Since the reconstruction algorithm has to run many times for spectra and light-curve extraction this is the slowest part of the script. For a faster generation of sky images the keywords *nospec* and *nolc* can be used. The final image will be saves as a FITS file and supplies with WCS coordinates. In the current iteration this step assumes that the roll angle of the telescope is aligned with the equatorial coordinate grid, i.e., x-z coordinates are aligned with RA-DEC. Below is the extracted light-curve and spectrum for a simulated GRB with a flux of $10^{-6}$ erg/s/cm$^2$.