An Algorithm for X-ray Single-Photon Energy-Dispersive Spectroscopy

Candidate Number: 1076012, Project: B803, Supervisor: Sam Vinko

March 2025

Abstract

This report outlines a hybrid approach to X-ray spectroscopy, combining Bragg spectroscopy with single-photon counting — X-ray Single-Photon Energy-Dispersive Spectroscopy (XSPEDS). This algorithm was demonstrated on CCD images of X-rays produced from L-shell emission of germanium from the experiment Pérez-Callejo et al. (2020) [1]. By using the two known Lα and Lβ emission energies for germanium, the algorithm is able to fit the physical setup of the experiment and determine the iso-energy contours on the CCD from 1.1 to 1.65 keV. Simultaneously, the CCD image is cleaned by identifying the photons and removing background noise. This achieved the signal-to-noise ratios and spectral resolutions in Table 9a. This method aims to achieve high spectral resolution with a high signal-to-noise ratio, which is critical for the implementation of X-ray spectroscopy. This notably includes experiments in plasma and condensed matter physics, where detailed spectra are required at low signal or limited exposure times.

1 Introduction

Obtaining high-resolution X-ray spectra in low-signal conditions remains an important goal in modern day condensed matter and high-energy experiments. Achieving high resolution often comes with a trade-off with noise. Traditional Bragg spectroscopy suffers from this, especially in low-intensity experiments [2]. However, single-photon-counting (SPC) is a great tool for mitigating noise, as we can identify photons and their count with a very high degree of confidence. This is useful for low signal experiments. However, this noise reduction comes at the expense of high spectral resolution.

X-ray spectroscopy has broad application, as it is a good diagnostic tool to investigate complex systems. It is capable of probing many types of materials on the atomic scale, from solids to gases [3]. For plasma and condensed matter physics, high resolution is essential for resolving closely spaced spectral lines as well as the spectral line profile of systems. This can be used to determine the temperature, density and magnetic fields in plasmas [4].

The goal of this report is to come up with a method that combines SPC with Bragg spectroscopy to achieve high spectral resolution with low noise; X-ray Single-Photon Energy-Dispersive Spectroscopy (XSPEDS). We use the data from Pérez-Callejo et al. (2020) where photons are detected by a CCD camera, and test our algorithm on this [1]. There are many recent computational algorithms that aim to suppress noise in the high-resolution regime of X-ray spectroscopy. Most of these are machine-learning based, such as Konstantinova et al. (2021) and Fu et al. (2025) [5, 6]. These approaches operate on already processed spectroscopy data, where the spectra have already been constructed. In contrast, XSPEDS works with the raw CCD images to maximise the spectral resolution and signal-to-noise ratio (SNR) using SPC and Bragg techniques.

This report goes through the methods for dealing with the CCD data and SPC, along with Bragg spectroscopy mapping. It then goes over the results for the data from Pérez-Callejo et al. (2020), and finally presents a discussion and conclusion on the algorithm [1].

2 Method

The approach can broadly be separated into two branches: Single-photon counting (SPC) and Bragg spectroscopy mapping. The SPC part removes the background noise as much as possible, and identifies the photon clusters. The mapping maps each photon to its energy based on its location on the CCD camera. The implementation code is available in the XSPEDS GitHub repository [7].

2.1 Background Experimental Setup and Equipment

This algorithm is based on data from Pérez-Callejo et al. (2020) [1]. Here an emission spectrum of germanium is produced by an X-ray free electron laser (XFEL), of energy around 1.4 keV, being focused onto germanium foil. The resulting emission spectrum is then Bragg diffracted using a beryl $(10\bar{1}0)$ crystal (with inter-planar spacing of 7.98 Å) and then picked up by a Princeton Instruments CCD camera, as shown in Figure 2a. This CCD camera is a 2048 by 2048 pixels square detector, each of width 13.5 μm.

2.2 CCD Data

Each of the 20 CCD images is a 2048 by 2048 `NumPy` 2D array, filled with values representing signals measured in Analog-to-Digital Units (ADU). This is expected to be very noisy, with significant background contribution. If we were to plot a histogram of all the values, we would expect a significant background pedestal, in the form of a Gaussian. Signals from photons are expected to cause large ADU values. Considering we are in the regime of low signal spectroscopy, we expect the frequency of these large ADU values to be several orders of magnitude lower than the peak of the pedestal, as shown in Figure 3. Consequently, we can fit a Gaussian in the pedestal region without the photon signal affecting the fit significantly at all.

The next step is to threshold out the pedestal, so we are left with as little background noise as possible while keeping enough photon data. We will consider ADU values on the histogram larger than the pedestal to be photon data. We expected a normal distribution of photon ADU values well separated from the background pedestal. However, this was not the case (see Section 3.1), and the photon ADU values extended into the pedestal region. To maximise our signal-to-background ratio, we determine a threshold point, V, at which the number of entries in the histogram is at least twice the number predicted by the fitted Gaussian. In other words, V is the ADU value that satisfies

$$V = \max\Big\{ x \mid f_{\text{obs}}(x) \ \leq \ 2\, f_{\text{fitted Gauss}}(x) \Big\}, \tag{1}$$

where V is the threshold ADU value, $f_{\text{obs}}(x)$ is the raw histogram and $f_{\text{fitted Gauss}}(x)$ is the fitted background Gaussian. All values below the threshold are set to 0.

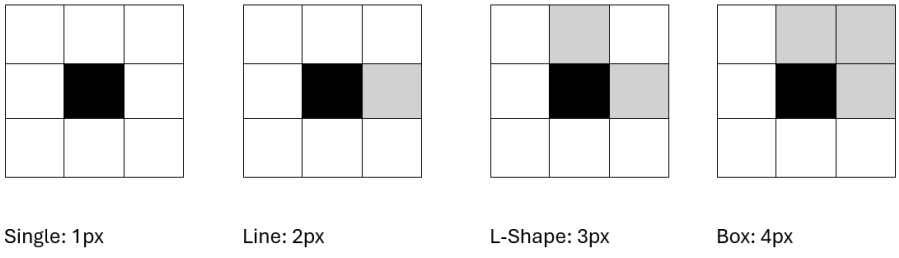


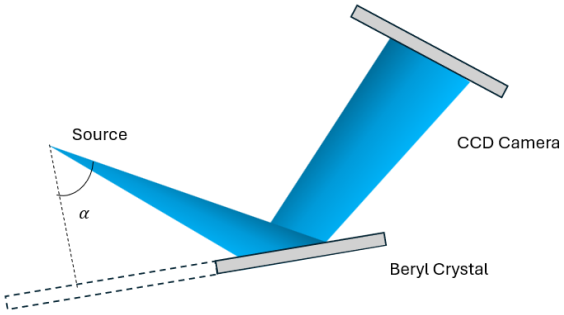
Figure 1: Photon cluster shapes. The black centre is the peak ADU value of the photon cluster.

Next we start identifying the photons. We expect single photon hits to form clusters of up to 4 pixels, as shown in Figure 1. We can map a photon to a pixel by identifying and mapping it to the pixel that has the highest ADU value in its cluster. We can then output a binary `NumPy` array that has 1 for a photon pixel, and 0 for no photons. We will have to consider the background noise still, and two-photon coincidences.

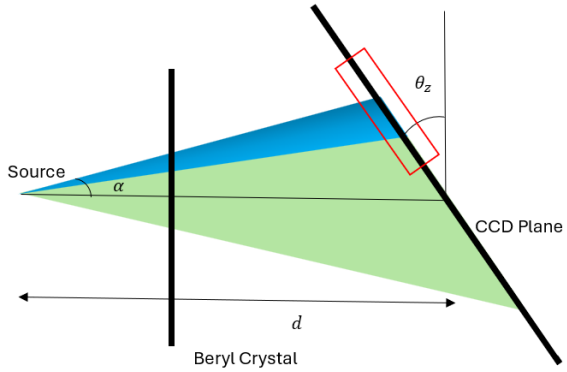
2.3 Bragg Spectroscopy and Mapping

By using known reference peaks in the germanium L-shell emission spectrum, we can solve for the geometry of the setup and map each photon to an associated energy. Figure 2a shows the setup. The resulting spectral pattern on the CCD is determined by the specific angles at which the X-rays hit the beryl crystal and end up being Bragg-reflected. These incidence angles are determined by the angular spread of X-rays from the source. With this, we can model Figure 2a using the schematic in Figure 2b, modeling the CCD camera and beryl crystal as a large plane. In this setup, all photons that make an angle α with the normal of the crystal plane hit the crystal plane. Bragg's law, $n\lambda = 2D\sin(\theta)$, determines the angle, α, from the normal, at which a photon with energy E is Bragg-reflected (let through), with $\theta = \frac{\pi}{2} - \alpha$. That is, the beryl crystal acts as a filter that maps E to a corresponding angle, α, from the crystal normal. From this we expect that these photons with energy E travel from the source to the CCD camera on the curved surface of a cone of half-angle α.

The CCD camera is initially setup parallel to the crystal plane, and then is tilted by θ_Z. The photons with energy E traveling along a curved surface will intersect with the CCD plane, producing conic sections. These conic sections will be elliptical and hyperbolic. Importantly, the CCD camera covers the vertices of the conic sections (see the the red box in Figure 2b). We can approximate the curves in this region with a parabola for a sufficiently large distance, d. From the X-Ray Data Booklet [8], there are two emission peaks at energies 1188 eV and 1218.5 eV for germanium (the $L\alpha$ and $L\beta$ emission lines). Using Bragg theory, we find that the Bragg angles are 40.86° and 39.63°, which means the conic half-angles, α, are 49.14° and 50.37° respectively. The resulting conic sections on the CCD camera is observed and used as a reference for finding the physical setup.



(a) The physical setup with the angular spread of the germanium X-ray source, Bragg-reflecting off the beryl crystal and being detected by the CCD camera. The dashed lines indicate the crystal normal, which is shown to the right in 2b.



(b) The equivalent setup of 2a. The blue projection indicates the X-ray spread that is picked up by the CCD camera (represented by the red box), as shown in 2a. The green part represents the cone at half angle α that is projected onto the CCD plane at a distance d. The resulting conic section is either elliptical or hyperbolic, depending on the tilt, θ_z, and α. The CCD camera picks up the vertices of these conic sections.

Figure 2: Setup and equivalent setup of the Bragg spectroscopy. Note that the red box is at the vertex of the resulting curve on the CCD.

For Figure 2b, we can define a set of global orthogonal coordinate axes, (x, y, z). We take the x axis to be normal to the crystal plane, and the z axis to be the rotation axis for θ_z. Considering the CCD images captures the vertices of the conic section curves, we can neglect a CCD plane rotation around the y axis. Further, we show later that we can neglect a rotation around the x axis (see Section 3.3). Defining a local orthogonal axes (u, v) on the CCD plane, where v is parallel to the z axis, we fit a parabola in the form

$$u = A_i(v - b)^2 + C_i \tag{2}$$

for each emission peak. Where $i \in \{1, 2\}$ labels the two emission lines Lα and Lβ respectively. Note that the vertex would be at (C_i, b) for each emission peak. The parameter A is related to the focal length of the ellipses/hyperbolas by $p = \frac{1}{4A}$, where p is the focal length.

To identify the emission peak curves on the CCD, we need to find the peak values in the 2048 by 2048 `NumPy` array. This is achieved by summing the values in batches of five rows and finding the column indices of the peak values for each of the 1188 eV and 1218 eV emission lines. We expect this to be particularly noisy, so a Gaussian filter (from `SciPy`) is applied to the sum plot for each 5-row batch to suppress spurious peaks.

This leads on to the main part of the mapping; modeling and finding the physical setup. First, we create a function, `CONICPARAM` that has the distance, d, the tilt of the CCD plane relative to the x=0 plane, θ_z, and the half-angle of the cone, α, as inputs. It outputs the resulting conic section equation in the basis of the CCD plane (See Appendix B for the detailed derivation). Next the goal is to find values of d and θ_z that accurately reproduces the two emission peak curves. Fitting parabolas (in the form of Equation 2) into the two scatter plots for each peak, we can then try to match their parameters to the curves generated from `CONICPARAM` for various values of d and θ_z. The algorithm iterates through parabola fittings, ds and θ_zs until a d and θ_z pair is found such that it accurately reproduces the fitted parabolas in the scatter plot. In order to converge on a valid solution for d and θ_z, we define the following residuals:

1. **Fitting residuals from the scatters:** Measures how well the parabolic curve fits the data.
2. **Focal length residual:** Ensures that the curvature of the curves from a d and θ_z pair matches the scatter-fit curve. This is based on the relationship Focal length = $\frac{1}{4A}$.
3. **Relative difference in vertices residual:** Compares $C_1 - C_2$ from the scatter to the difference in vertex location of the curves generated from a d and θ_z pair. In other words, it compares how well the gap between the curves matches for the scatter fit parabola and the generated curves from d and θ_z.

Since 2 and 3 are relative, they remove the need to precisely determine the CCD location. Unfortunately, this process is quite volatile, i.e. it is very sensitive to minor changes in the parabolic fits. In order to converge on a valid solution, the optimisation is performed in two stages: global optimisation with differential evolution, and local refinement with least squares. The global optimisation identifies the global minimum. It uses a stochastic evolutionary algorithm from `SciPy`. The least squares approach uses a local gradient-based approach, trust-region reflective, also from `SciPy`.

2.4 Spectral Lineout

Once we have the optimal d and θ_z parameters, we then create iso-energy curves along the CCD 2048 by 2048 grid using the `CONICPARAM` function. We can sum the number of photons along these iso-energy curves, with some tolerance. Here the tolerance is the lateral (u axis) extent around each pixel on the iso-energy curve. To determine the tolerance, we can look at one of the reference peaks and determine the SNR for a range of tolerances. However, we will have to keep the full-width at half-maximum (FWHM) as low as possible by not choosing too broad a tolerance, which would to artificially increase the FWHM beyond the minimum. This minimum will be due to factors such as source broadening and crystal mosaicity. To calculate the signal-to-noise ratio (SNR) for a peak, we use

$$\mathrm{SNR} = \frac{S - B}{\sigma_B},$$

where S is the peak amplitude, B is the local baseline level, and σ_B the standard deviation of the background fluctuations (i.e. the noise).

Since the spectrum is continuous, we use the `pybaselines` library to correct for the background level. The `pybaselines` library uses a rolling window approach to determine the local minimum in each segment of the spectrum, and then subtracts this from the segment. The identified FWHM gives us the spectral resolution.

The local dispersion (the energy interval spanned by one pixel along the dispersion direction), changes significantly across the spectrum. Hence, summing the photon counts within a constant pixel tolerance would be integrating over different energy ranges. Consequently, we need to normalise by dividing the raw sums along the curves by the energy bin width to get the true differential spectrum (i.e. the counts per eV). We define a small change in energy, ΔE, which corresponds to a pixel shift along the dispersion axis of $\Delta p = u(E + \Delta E) - u(E)$. Here $u(E)$ is the pixel position of the vertex of the resulting curve on the CCD for energy E, given by:

$$u(E) = \texttt{CONICPARAM}\big(d, \theta_z, \alpha(E)\big).$$

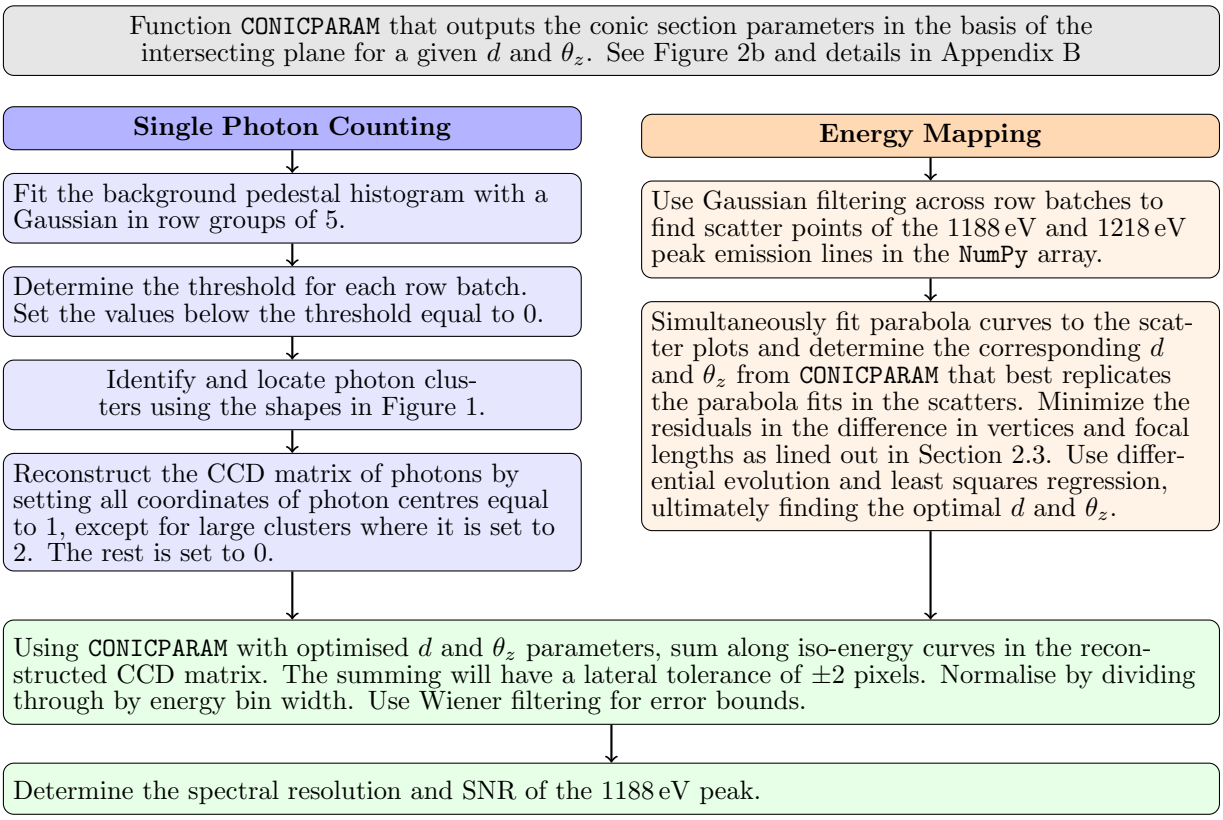
Note that the energy is converted to the conic half angle, α, using Bragg's law (section 2.3). With this,we obtain the width of the energy bin at energy E:

$$W(E) \;=\; (2\,\mathrm{tol} + 1)\,\frac{dE}{dp} \quad \text{with} \quad \frac{dE}{dp} \approx \frac{\Delta E}{\Delta p} = \frac{\Delta E}{u\big(E + \Delta E\big) \;-\; u(E)}, \tag{3}$$

where tol is the tolerance.

Finally, the expected error in the photon counts is mainly due to shot noise, so is expected to follow Poisson statistics. Consequently, the standard deviation would be $\sqrt{N}$, where N is the photon count. Wiener filtering from `SciPy` was used with a window size equal to the FWHM to plot the error bounds. This is justified by the large size of the FWHM compared to the energy bin width of the iso-energy curves. The filtering reduces some of the noise whilst preserving the peak shapes.

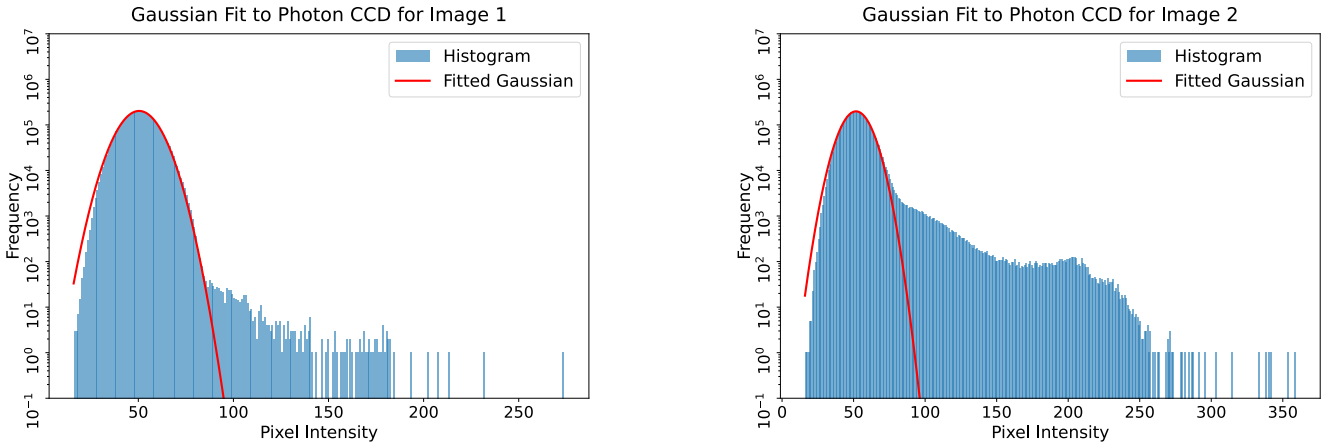
2.5 Code diagram



3 Results

3.1 CCD Data

First the goal is to isolate the photon signals by removing the background noise. Taking an ADU histogram of the CCD images, we observe Figure 3 for dark frame (no X-ray illumination) and light frame (X-ray illuminated) images. This confirms that the pedestal is a Gaussian, from the dark frame histogram, and that the frequency of photon signals are around 2 orders of magnitude lower than the background, from the light frame histogram. We do observe an overlap of the photon signals with the background pedestal. Notably, there is a lack of uniformity of the ADU values across the different images. When we plot the mean of the pedestal in the rows, we observe Figure 4.



(a) Dark-frame background histogram (Image 1).

(b) X-ray illuminated frame histogram (Image 2).

Figure 3: Gaussian fittings for Image 1 (dark frame) and Image 2 (light frame). We observe that the dark frame histogram has relatively few values beyond the Gaussian fit to the right, compared to the light frame.

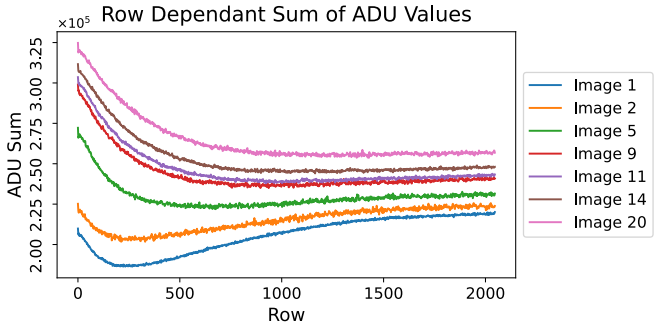


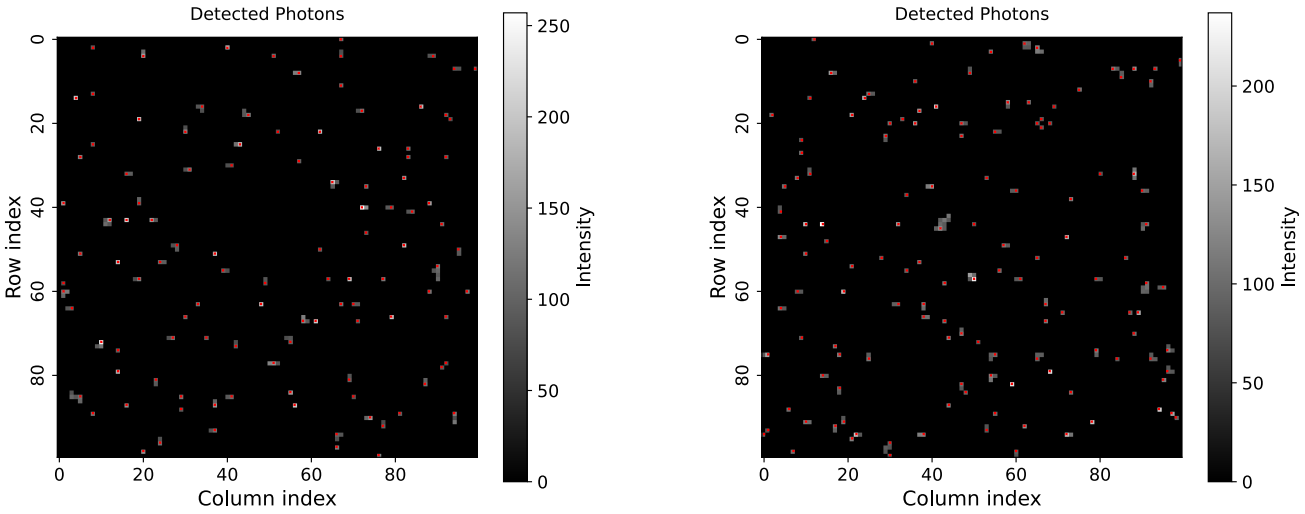
Figure 4: Pedestal function variation across rows. We observe the increasingly exponential decay behaviour of the row mean as we go up the image indices (i.e. from 1 to 20).

To deal with this lack of uniformity, we use dynamic thresholding of the pedestal. To achieve this, we fit the pedestal for batches of 5 rows and identify the optimal threshold for this batch. For this, we find the ADU value, V, that satisfies Equation 1. All values below the threshold are set to 0. We find that the optimum threshold leaves 1-3 % of the counts as background and the density of 'non-zero' values less than 1%. In the peak regions, this density never exceeds 4%. See table 1 in Appendix A.

3.2 Identifying the Clusters

Once we have removed the pedestal, we can identify photon clusters. Photon clusters will be in the shape of Figure 1. However, we do expect some clusters not to conform to the shapes due to background noise

and overlapping photons. These odd shapes make up less than 3% of the clusters, even in the peak regions. Consequently, we can ignore the scenario of photons completely overlapping each other. See Figure 5 for an example.



(a) Edge section of the CCD.

(b) Emission peak section of the CCD.

Figure 5: Detected photons in the peak region (b) vs. non-peak region (a) for Image 2. The density of the peak region has 2.5% of the grid section filled in, double the image average.

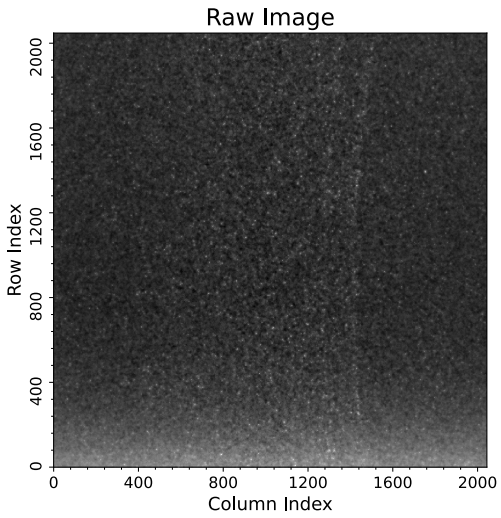
Finally, we reconstruct the `NumPy` array for photon events. The algorithm sets the pixel with the maximum ADU value in the cluster as the coordinate of the photon. If it detects a cluster larger than 4 pixels, it will assume that there are 2 photons. However, this probably is not always the case; it is possible to have some background mixed in. But since these shapes make up less than 3% of the total counts, and the background only makes up 1-3% and the density of clusters is around 1%, we can ignore these scenarios. We then modify the `NumPy` array to have a value of 1 at the coordinates of photons and 2 at the larger clusters. The rest will be 0.

3.3 Mapping

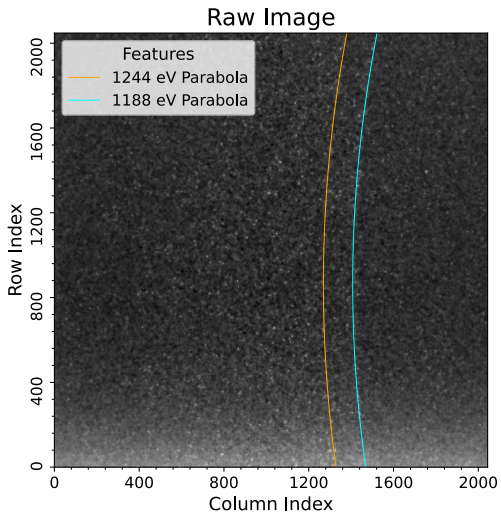
To map the pixel location to an associated energy value, we use the two reference peaks of 1188 eV and 1218.5 eV for germanium. The curves of these emission peaks are visible in Figure 6a, and the direction of the dispersion is horizontal. This justifies neglecting a CCD rotation around the x axis in Section 2.3. Image 9 was used as a reference for mapping. When identifying the emission peak values across the rows for this image, the algorithm takes the spectral lineout of the `NumPy` array perpendicular to the dispersion direction for batches of 50 rows. Since this is noisy, Gaussian filtering is used to determine the peaks accurately. A scatter plot is obtained for both peaks, as shown in Figure 7.

Using the iterative method outlined in 2.3, we obtain parabolic fits for the 2 reference peaks, Figure 7, in the form $u = A(v-b)^2 + C$. The algorithm found that with $d = 11.3\,\text{cm}$ and $\theta_z = 40.86°$, we can simulate these parabolic fits with a 0.0039 pixel difference in the relative vertex location (i.e. u axis location) of the 2 curves. Further these d and θ_z parameters simulate the fitted focal lengths correctly to 0.0085 pixels. The focal length is related to the curvature, by $F = \frac{1}{4A}$, where F is the focal length. Note that the found d and θ_z parameters found are very unlikely to be close to the real values, but they are a good solution to simulate these curves to a high degree of accuracy. That is, if we input the d and θ_z values found in `CONICPARAM`, we can simulate the fitted scatters to a very high degree of accuracy. For

the scatter fit, we have errors of around 2 pixel lengths for the C and b parameters, and a 6% error in the A parameter. This may affect the tolerance in the spectral lineout, as will be discussed.



(a) Raw Image 9.



(b) Parabolic curves on raw Image 9.

Figure 6: (a) Raw Image 9 and with (b) parabolic overlays.The 2 curves are visible around the 1500 column index on the raw image.

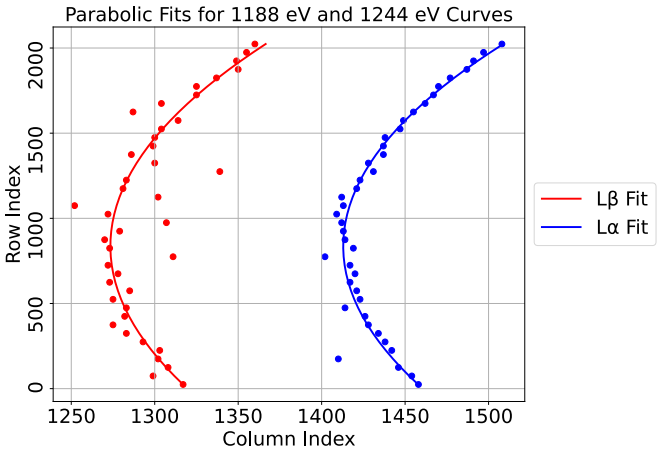


Figure 7: Parabolic fitting on the scatters for the $L\alpha$ and $L\beta$ emission lines.

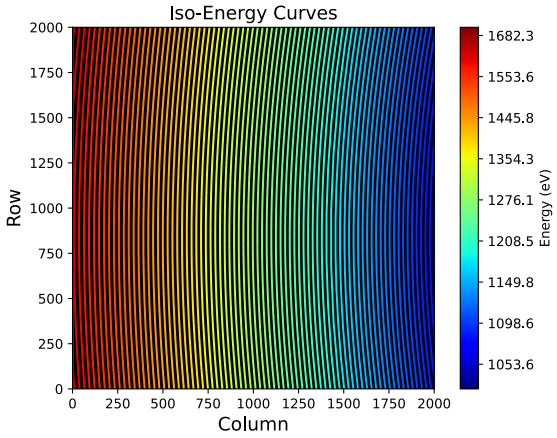


Figure 8: Iso-energy contours. Energy ranges from around 1100 eV to 1600 eV.

3.4 Spectral Lineout

Given our iso-energy contours on the photon matrices, we can obtain the spectrum by summing the values within a fixed lateral window, of $\pm$tol pixels around each contour (where tol is the tolerance and will be determined). However, as mentioned in 2.4, the local dispersion is not constant. We find that for our optimal d and θ_z, the dispersion relation $\frac{\Delta E}{\Delta p}$ scales from 0.17 eV to 2 eV from 1100 eV to 1600 eV. In order to correct for this and obtain the true count per energy bin, we will need to divide the sums through by Equation 3.

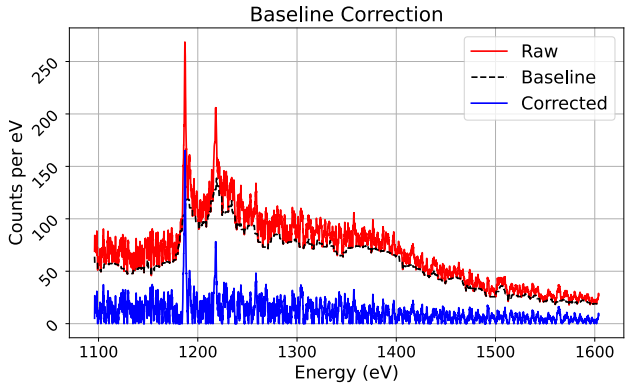
To determine the lateral tolerance, we need to maximise the signal-to-noise ratio (SNR). To do this, we can look at our peak emission line (1188 eV) and calculate the SNR here. We choose the lateral tolerance by maximising the SNR of the 1188 eV line, while ensuring that the FWHM does not increase compared to smaller tolerances, to avoid artificial broadening. As mentioned, we will need a baseline correction

using `pybaseline`. A tolerance of 2 pixels gave the best SNR of 16.0 while keeping the FWHM around 2 eV for Image 2. For all images, the FWHM ranged from 1.9 to 2.7 eV for the 1188 eV peak, as shown in Table 9a. These FWHM correspond to the effective spectral resolution.

Note that source broadening is irrelevant here. From Pérez-Callejo et al. (2020), we know that the source is 28.5μm^2 in size which corresponds to around 6 μm in length, less than the pixel size of 13.5 μm [1]. Considering that a pixel separation along the dispersion direction corresponds to roughly 0.2 eV, the finite source size would introduce under 0.1 eV of broadening, far less than the measured FWHM.

Image	FWHM (eV)	Noise	SNR
2	1.98	8.81	16.0
3	2.04	6.41	18.2
5	2.03	6.67	22.1
7	1.92	7.68	17.7
8	1.84	6.69	16.1
9	1.95	6.56	19.3
15	2.14	6.9	16.8
18	2.15	11.74	11.9
20	1.94	5.79	24.2

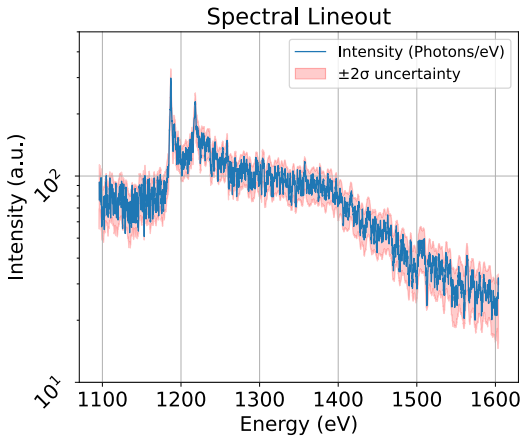
(a) Measured FWHM, Noise and SNR for the 1188 eV line.



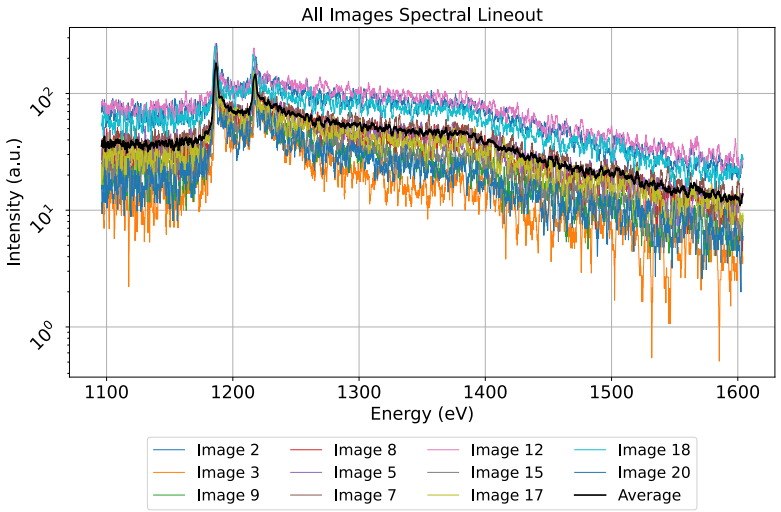
(b) Example pybaselines correction for Image 2

Figure 9: (a) Table of SNR and FWHM for light-frame images. (b) Example of a baseline correction for Image 2. Image 18 seems to have a significantly lower SNR compared to the other images. However its spectral lineout is similar to Image 2 as shown in Figure 10b.

Due to the dynamic thresholding from Section 3.3, less than 3% of the data is background noise. Consequently, the dominant uncertainty is the photon shot noise. This follows Poisson statistics, where the standard deviation is given by $\sigma = \sqrt{N}$, where N is the summed count for an iso-energy curve. We will also have to propagate the error to obtain the uncertainty error with the energy bin normalisation. With this,we observe Figure 10 with Wiener smoothening used on the error bounds to reduce spurious noise.



(a) Spectrum for Image 2 with Wiener–smoothed error bounds.



(b) Spectra for all images.

Figure 10: (a) Single spectrum with error bounds and (b) spectra for all images along with the average. The 1188 eV and 1218.5 eV emission lines are visible.

4 Conclusion

In this report, we proposed and tested an algorithm for a hybrid approach to X-ray spectroscopy, that combined SPC and Bragg spectroscopy. Testing this algorithm on data from Pérez-Callejo et al. (2020), we obtained the spectra in Figure 10, with spectral resolutions around 2 eV, as shown in Table 1 [1]. Through iterative geometry fitting, the mapping algorithm simulates the $L\alpha$ and $L\beta$ emission peak curves on the CCD to sub-pixel accuracy. Simultaneously, with dynamic thresholding and cluster identification, the SPC algorithm maximises the number of photon events while leaving only 3% of residual background noise in the identified photon events.

For future work, we could also allow for more degrees of freedom in the simulation, like allowing for a θ_x rotation, as well as testing on other setups/materials. The sensitivity of the geometric simulation could also be improved upon. For experiments where other emission lines are visible (e.g. K-shell transitions), incorporating them as additional reference peaks could make the optimisation more constrained and less sensitive to misfittings in any single peak.

However, XSPEDS shows promise as a good tool for enhancing spectroscopic measurements. It can be implemented in real time in experiments to correct for changes such as thermal drifts or other setup instabilities. It could be especially valuable in achieving high resolution in low-signal regimes in plasma and condensed matter physics.

References

[1] Gabriel Pérez-Callejo et al. "X-ray Spectroscopic Studies of a Solid-Density Germanium Plasma Created by a Free Electron Laser". In: *Applied Sciences* 10.22 (2020). ISSN: 2076-3417. DOI: 10.3390/app10228153. URL: https://www.mdpi.com/2076-3417/10/22/8153.

[2] Shilpa Pradhan et al. "Intensity noise reduction in a multiwavelength distributed Bragg reflector fiber laser". In: *Optics Letters* 31 (Sept. 2006), pp. 2963–2965. DOI: 10.1364/OL.31.002963.

[3] Sadia Bari et al. "Soft X-ray Spectroscopy as a Probe for Gas-Phase Protein Structure: Electron Impact Ionization from Within". In: *Chemistry – A European Journal* 24.30 (2018), pp. 7631–7636. DOI: https://doi.org/10.1002/chem.201801440. eprint: https://chemistry-europe.onlinelibrary.wiley.com/doi/pdf/10.1002/chem.201801440. URL: https://chemistry-europe.onlinelibrary.wiley.com/doi/abs/10.1002/chem.201801440.

[4] A. Ya. Faenov. "X-ray spectroscopic methods for measuring the parameters of high-temperature dense plasma". In: *Measurement Techniques* 40 (1997), pp. 94–100. URL: https://api.semanticscholar.org/CorpusID:120021382.

[5] Tatiana Konstantinova et al. "Noise reduction in X-ray photon correlation spectroscopy with convolutional neural networks encoder–decoder models". In: *Scientific Reports* 11.1 (July 2021). ISSN: 2045-2322. DOI: 10.1038/s41598-021-93747-y. URL: http://dx.doi.org/10.1038/s41598-021-93747-y.

[6] Tianyu Fu et al. "Deep Learning for Spectroscopic X-ray Nano-Imaging Denoising". In: *Advanced Intelligent Systems* 7.1 (2025), p. 2400318. DOI: https://doi.org/10.1002/aisy.202400318. eprint: https://advanced.onlinelibrary.wiley.com/doi/pdf/10.1002/aisy.202400318. URL: https://advanced.onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202400318.

[7] B803 Candidate. *XSPEDS B803*. github.com/xspeds. My first name is in the username! Apr. 2025.

[8] Center for X-Ray Optics, Lawrence Berkeley National Laboratory. *X-Ray Data Booklet*. Table 1-2: Photon energies of principal K-, L-, and M-shell emission lines. Berkeley, CA, 2009.

A Photon Cluster Statistics

Table 1: Table of photon data. The background/photon cluster ratio is the ratio of the clusters that are left over background noise from the thresholding. The density of clusters is the fraction of non-zero pixels.

Image	Background/Photon Cluster Ratio	Density of Clusters	Total Clusters	Odd Shape Clusters
2	0.028	0.014	41026	1222
3	0.014	0.003	10408	39
5	0.023	0.007	23002	242
7	0.021	0.007	24817	243
8	0.02	0.006	20018	126
9	0.015	0.004	14643	87
15	0.014	0.004	14799	40
17	0.019	0.005	18537	78
18	0.024	0.011	34086	419
20	0.015	0.003	12125	26

B `CONICPARAM` Function

The function `CONICPARAM` computes the parameters of the conic section resulting from the intersection of the cone (with vertex located at $(-d, 0, 0)$ and half-angle α) with a plane that goes through the origin and is rotated around the x, y, z axes with respect to the $x = 0$ plane. This models the setup as in Figure 2b, where the cone and half angle represent the source of the photons and Bragg angle, with the plane representing the CCD camera. This appendix goes through the mathematical background for the resulting conic sections, which are either in the form of an ellipse, parabola or hyperbola. The parameters outputted include the focal lengths, the centre and vertex's location, and other relevant geometric quantities in the basis of the CCD plane. We determine the centre by finding the stationary point of the quadratic equations, then by diagonalising the normalised quadratic coefficient matrix the semi axes and transverse/conjugate axes are determined. From this we get the vertex and foci locations in the basis of the CCD plane.

This function was created with the intention of having θ_x and θ_y rotations in the simulation. Since these rotations were ultimately neglected, there are faster, less complex ways to determine these parameters.

First, the basis vectors in the CCD plane are found. This is by using rotation matrices:

$$R_z(\theta_z) = \begin{pmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad R_y(\theta_y) = \begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{pmatrix} \quad R_x(\theta_x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{pmatrix}.$$

The basis vectors in the CCD plane are then given by the product of these with the global basis:

$$\hat{\mathbf{e}}_u = R_x(\theta_x)\, R_z(\theta_z)\, \hat{y} = \big(e_{\mathrm{u1}},\, e_{\mathrm{u2}},\, e_{\mathrm{u3}}\big), \qquad \hat{\mathbf{e}}_v = R_x(\theta_x)\, R_y(\theta_y)\, \hat{z} = \big(e_{\mathrm{v1}},\, e_{\mathrm{v2}},\, e_{\mathrm{v3}}\big).$$

where $\hat{y}$ and $\hat{z}$ are part of the global basis. Note that the $\hat{x}$ transformation is neglected here as it gives the unit vector normal to the plane (since the unrotated plane is x = 0).

The cone with half angle α can be described by the equation in the global basis:

$$y^2 + z^2 = \tan^2(\alpha)(x - d)^2, \tag{4}$$

where d is the distance of the cone's vertex from the origin (located at $(-d, 0, 0)$). See Figure 2b. Next the conic section is found. By substitution into Equation (4), we can express the conic section equation in the quadratic form

$$F(u, v) = A\, u^2 + B\, uv + C\, v^2 + D\, u + E\, v + F = 0, \tag{5}$$

where u) and v are the coordinates in the basis of the CCD plane. The coefficients are given by

$$\begin{aligned}
A &= e_{\mathrm{u}2}^2 + e_{\mathrm{u}3}^2 - \tan^2(\alpha)\, e_{\mathrm{u}1}^2, \\
B &= 2\Big[(e_{\mathrm{u}2}\, e_{\mathrm{v}2} + e_{\mathrm{u}3}\, e_{\mathrm{v}3}) - \tan^2(\alpha)\, e_{\mathrm{u}1}\, e_{\mathrm{v}1}\Big], \\
C &= e_{\mathrm{v}2}^2 + e_{\mathrm{v}3}^2 - \tan^2(\alpha)\, e_{\mathrm{v}1}^2, \\
D &= 2\, \tan^2(\alpha)\, d\, e_{\mathrm{u}1}, \\
E &= 2\, \tan^2(\alpha)\, d\, e_{\mathrm{v}1}, \\
F &= -\tan^2(\alpha)\, d^2.
\end{aligned}$$

The conic sections are either in the form of an ellipse, hyperbola, or parabola. These are classified by the discriminant:

$$\Delta = B^2 - 4AC.$$

If $\Delta = 0$, then the conic section is a parabola, if $\Delta < 0$ it is an ellipse, and if $\Delta > 0$ it is a hyperbola.

The centres can then be found. For the case of the hyperbola, the center is the middle point between the vertices of the 2 branches (note that we observe the other branch from the negative projection of the cone and so is not 'real'). For a parabolic section, there is no centre, so the vertex is found instead. The centres (or, in the case of a parabola, the vertex) can be determined by finding the stationary point of $F(u, v)$. Taking partial derivatives and setting them equal to zero,

$$\frac{\partial F}{\partial u} = 2Au + Bv + D = 0 \quad \text{and} \quad \frac{\partial F}{\partial v} = Bu + 2Cv + E = 0,$$

gives a system of linear equations. Solving results in the coordinates of the centre for an ellipse and hyperbola, and the vertex for a parabola.

Next, we find the properties specific to ellipses and hyperbolas. These are the semi-major/minor axes for ellipses and transverse/conjugate axes for the hyperbolas. The method for finding these properties is the same; the normalized quadratic coefficient matrix is diagonalised, and the eigenvalues relate to the semi-major/minor and transverse/conjugate axes.

The quadratic coefficient matrix is the matrix form of the quadratic part of Equation (5), so that

$$\begin{pmatrix} u & v \end{pmatrix} Q \begin{pmatrix} u \\ v \end{pmatrix} = Au^2 + B\, uv + Cv^2, \quad \text{where} \quad Q = \begin{pmatrix} A & \frac{B}{2} \\ \frac{B}{2} & C \end{pmatrix}.$$

This is normalised by dividing by F in Equation (5). An ellipse and a hyperbola can be written in the form

$$\frac{u'^2}{a^2} \pm \frac{v'^2}{b^2} = 1,$$

where the $+$ corresponds to an ellipse and the $-$ to a hyperbola, and u' and v' are the transformed coordinates. By diagonalising the matrix, we transform the quadratic equation into this form. With this, the eigenvalues relate to the semi-major/minor axes (ellipse) and transverse and conjugate (axes) by:

$$\lambda_1 = \frac{1}{a^2} \quad \text{and} \quad \lambda_2 = \pm\frac{1}{b^2},$$

where the $+$ corresponds to an ellipse and $-$ to a hyperbola. To determine the vertex, the semi-major and transverse axes are used. For an ellipse, the larger eigenvalue corresponds to the semi major axis. For a hyperbola, there will be negative and positive eigenvalues, and the positive one corresponds to the transverse axis. The coordinates of the vertices, $\mathbf{t}$, are given by

$$\mathbf{t} = \mathbf{c} \pm a, \hat{\boldsymbol{v}}_1.$$

where a is the semi-transverse, or semi-major, axis length, $\hat{\boldsymbol{v}}_1$ is the eigenvector in the corresponding direction of the major/transverse axis, $\mathbf{c}$ denotes the centre coordinates. Similarly, the focal points, $\mathbf{f}$, can also be determined. In the case of an ellipse, the focal points are located at a distance

$$\mathbf{f} = \mathbf{c} \pm a\, e\, \hat{\boldsymbol{v}}_1,$$

from the centre, where e is the eccentricity. For a hyperbola, the focal points are at

$$\mathbf{f} = \mathbf{c} \pm c\, \hat{\boldsymbol{v}}_1,$$

where $c = \sqrt{a^2 + b^2}$.

In all we get the relevant geometric properties for a conic intersection on the CCD plane. This helps with detemining the geometry of the setup so the iso-energy curves can be determined. An example of this is given in Figure 11.

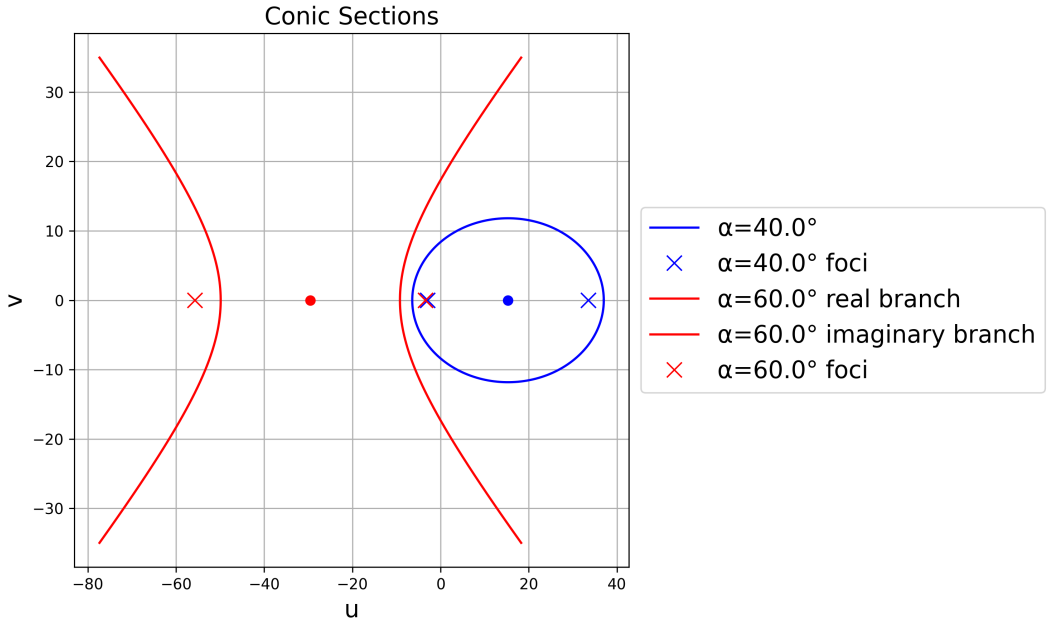


Figure 11: Conic section example output for $d = 10$ pixels and $\alpha = 40°$ and $60°$. Only the right branch matters are the real conic section, as the left branch is from the negative projection. This part corresponds to the red box part in Figure 2b