Electronic Supplementary Material (ESI) for Journal of Analytical Atomic Spectrometry.

Supplementary Material

Photon Event Evaluation for Conventional Pixelated Detectors in Energy-Dispersive X-ray Applications

Jonas Baumann[1*], Richard Gnewkow[1], Steffen Staeck[1], Veronika Szwedowski-Rammert[1], Christopher Schlesiger[1], Ioanna Mantouvalou[1] and Birgit Kanngießer[1]

[1] Technical University of Berlin, Institute of Optics and Atomic Physics, Hardenbergstraße 36, D-10587 Berlin, Germany

Abstract:

Here we show details of the performance of photon event evaluation (PEE) algorithms with respect to photon density on the detector chip and applied noise thresholds for the evaluation of the data. We found that the photon recovery decreases linearly down to about 50 % for most algorithms when 50.000 photons are simulated on a CCD with 1024 × 1024 pixels. The noise thresholds drastically influence the shape of the spectra and have to be chosen with care for every application.

Furthermore, with the supplementary material comes a zip-file, containing all the energy-dispersive spectra of the investigation. To allow a comparison with the here-applied PEE algorithms, the python program used to generate the dark-frame corrected CCD frames and the data set with σ_{dark} = 9 e^- and σ_{cc} = 0.3 p is also included in the zip-file. More information on how to use the data can be found at the end of this document.

Investigation of resolving power and photon recovery with various photon densities:

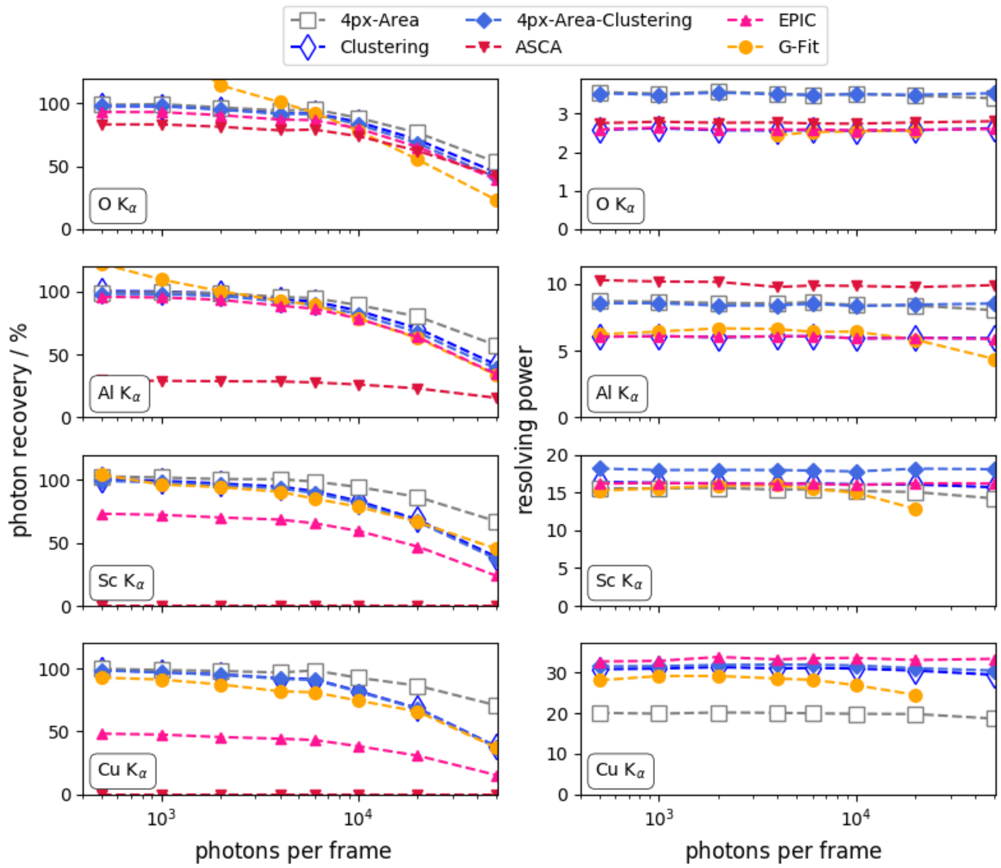

Figure S1: Results of the photon density study for the various PEE-algorithms with respect to photon recovery and resolving power using simulated data on a CCD with 1024 × 1024 pixels. Left column: Already above 1000 photons per frame the recovery of all algorithms starts to degenerate, reaching values of about 50 % at 50.000 photons per frame. The decrease is linear. Right column: The resolving power is hardly affected by an increase of the photon density. Only the high-energy fluorescence peaks (Cu and Sc K_α) in the spectrum of the *G-Fit* algorithm show increasing distortions at large photon densities (not shown) leading to a decrease in resolving power.

Influence of noise thresholds onto spectrum shape and cluster-size dependent contributions:

The simulated data set with 100 frames, σ_{dark}= 9 e^-, σ_{cc}= 0.3 *p* and 1000 photons per frame is used to investigate the influence of the noise thresholds. For this purpose, the data set is analyzed 9 times with every PEE algorithm, using a first noise threshold $T_1 \times \sigma_{dark}$ with $T_1 \in \{3, 5, 7\}$ and a second noise threshold $T_2 \times \sigma_{dark}$ with $T_2 \in \{1, 3, 5, 7\}$. From the evaluated data, spectra are computed and compared in Figures S2-S7 to qualitatively describe the influence of the noise thresholds. For this purpose, not only the actual spectrum of all evaluated photon events is shown (blue), but also the contributions to this spectrum from photon events consisting of n pixels ($n \in$ [1-5]). Note that contributions from higher n to the sum spectrum might be present as well, but are not plotted for reasons of clarity.

The comparison reveals that the first noise threshold only defines the cut-off energy at the low-energy side, but otherwise leaves the spectra unchanged. The second noise threshold however, which is often set equal to the first noise threshold, dramatically influences the spectral shape. Often, there exists an (qualitative) optimum with hardly any distortion of the spectra. In the case of the present study, the graphs show that the chosen noise thresholds of $T_1 = T_2 = 3$ are reasonable. Note that the actual values of the noise thresholds might depend on the camera properties.

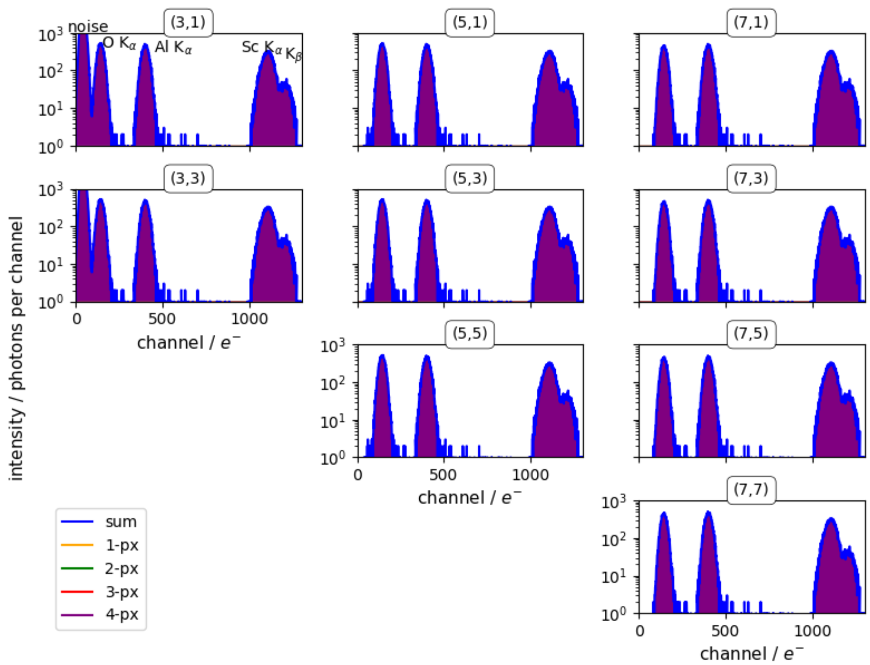

Figure S2: Spectra for various noise thresholds (T_1,T_2) for the *4px-Area* algorithm. For details see text. There is no influence on the spectra due to the second noise threshold, since it is not applied in this PEE algorithm. The cut-off of the noise peak at about 0 e^- is removed by increasing the first noise threshold.

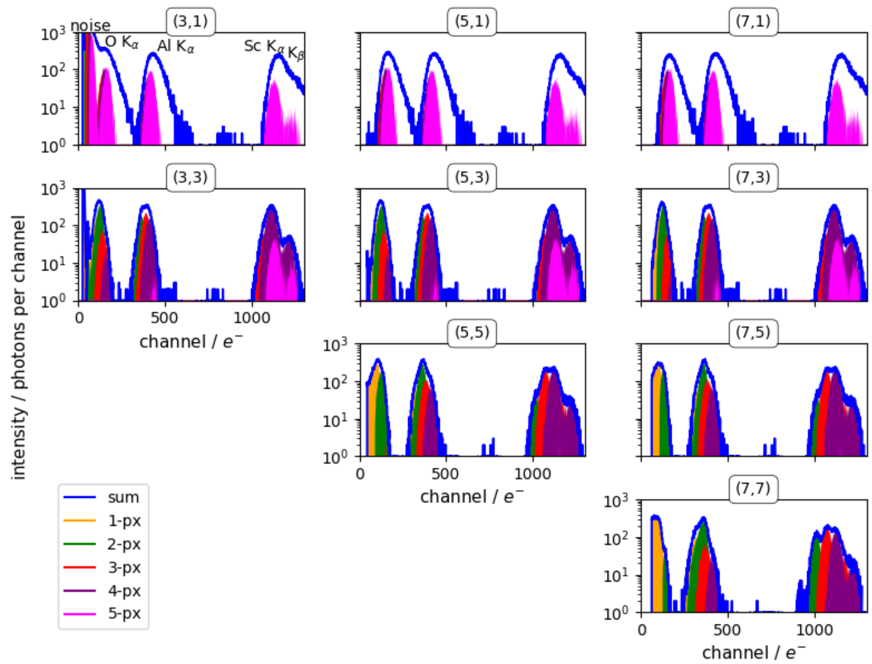

Figure S3: As S2, but results are shown for the *Clustering* algorithm. If the second noise threshold is chosen too low (T_2 = 1) or too high (T_2 = 5 or 7) the shape of the fluorescence peaks distorts, especially of those with high energy. In the first case, too many noisy pixels contribute to each photon event (the spectrum consists partly of photon events with n > 5). In the last case, not the whole charge cloud is collected for the low n-px events (incomplete charge collection). This leads to a relative shift of the various peaks of about $\Delta n \times T_2 \times \sigma_{dark}$. If a correction of this factor can enhance the spectral shape and resolving power will be tested in future studies.

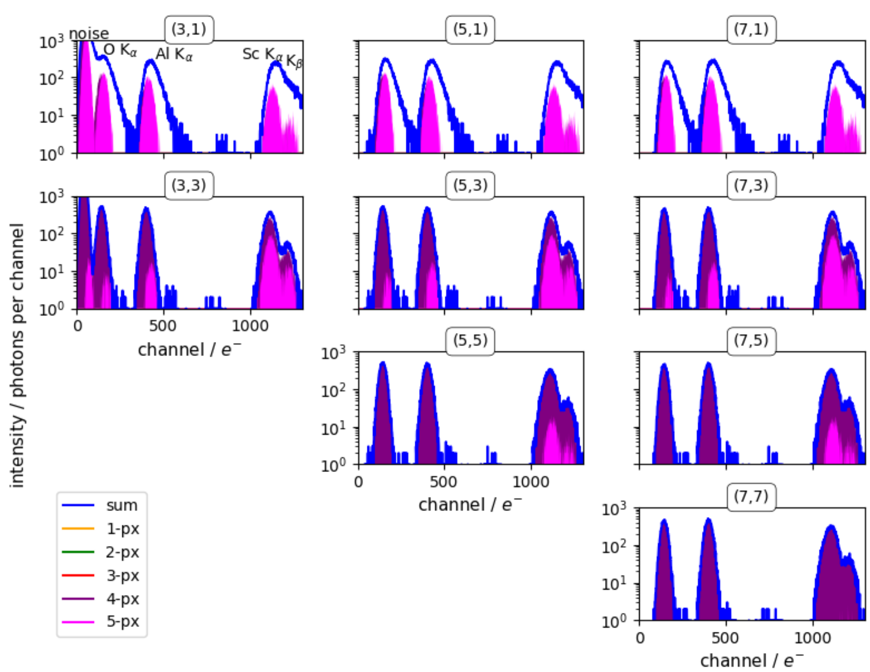

Figure S4: As S2 and S3 but for the *4px-Area-Clustering* algorithm. The influence of a high second noise threshold leading to incomplete charge collection is less drastic, since at least 4 pixels are used for the evaluation.

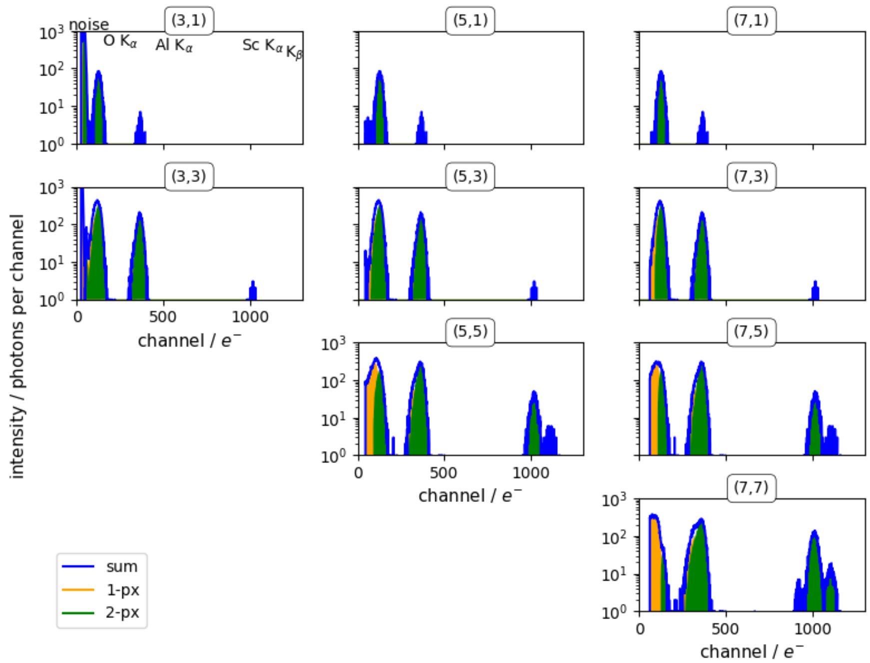

Figure S5: Here the results of the *ASCA* algorithm are shown. The second noise threshold strongly influences the height of the fluorescence peaks, since the event patterns become distorted, if noisy pixels are assigned as valid. Thus, for low T_2, many photon event patterns are likely not recognized as photon event and rejected. For high T_2 the low-energy peaks become asymmetric due to incomplete charge collection. Note that only patterns with less than 2 contributing pixels are valid.

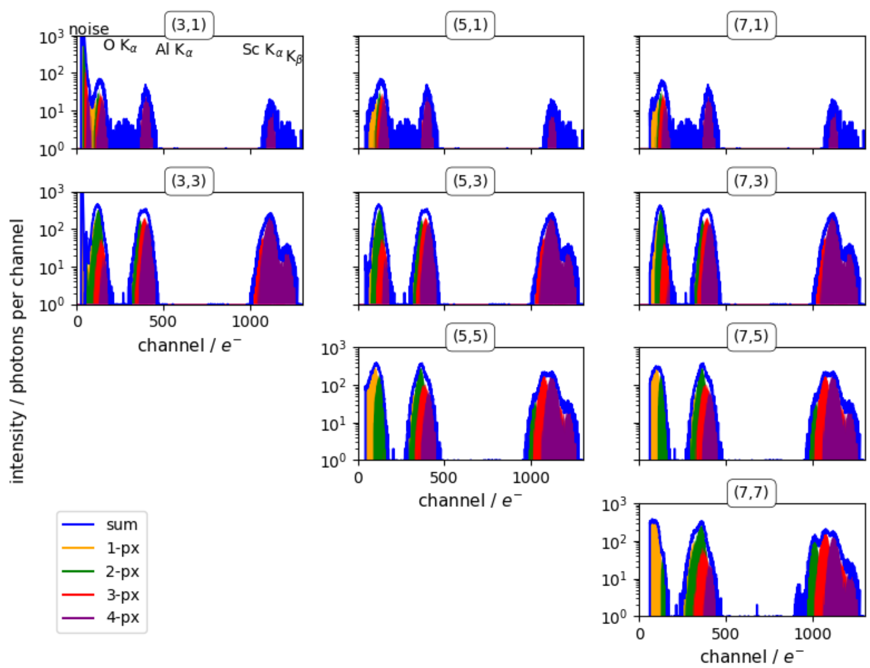

Figure S6: Similar to S5. Here the results for the *EPIC* algorithm are shown, which also evaluates photon events with charge splitting to up to 9 pixels. Too high values of T_2 distort the spectral shape due to incomplete charge collection, while a value too low leads to the rejection of the photon event pattern due to noisy pixels.

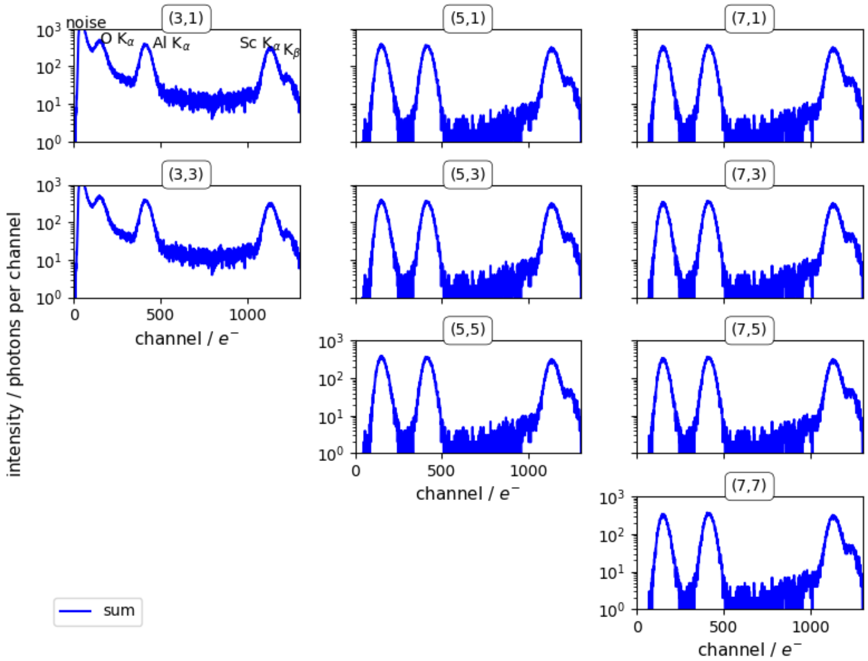

Figure S7: Here the results for the *G-Fit* algorithm are shown. As in S2, the second noise threshold is not used for the evaluation. However, the first noise threshold factor T_1 can significantly reduce the spectral background. Here, a value of $T_1 < 5$ leads to an increased background, but the peak shapes are hardly affected. Note that all pixels in the 5×5 box contribute to the fit results of the *G-Fit* algorithm, and thus to the spectrum. Therefore, the various *n*-px spectra do not exist.

Description on how to use the provided raw data:

This section contains information on how to use the raw data available in the supplementary materials zip-archive.

Due to the large amount of data, not all simulated CCD frames could be provided in the supplementary material. However, in "/simulated_data/ccd_data/" the data set with σ_{dark} = 9 e^- and σ_{cc} = 0.3 *p* is stored. It consists of 100 16-bit tif-images, which can be opened either by using modules like Christoph Gohlkes tifffile.py (https://www.lfd.uci.edu/~gohlke/code/tifffile.py.html), the python-package “imageio” or the free-ware software “ImageJ”. Also, the python script used to create the simulated data is provided (“simulate_SPEs.py”). Here the important settings can be changed as is shown in Table 1.

Furthermore, all energy-dispersive spectra, as evaluated with the various PEE algorithms and threshold settings, can be found in "/simulated_data/spectra/". The subfolders' names show what property was varied and the subsubfolders’ name what PEE algorithm was used for data evaluation. The file name includes what ccd noise and charge-cloud size settings were used in the simulation, e.g. "sim_sigmaCC_0.30_sigmaNoise_9" means a charge-cloud size of 0.3 *p* (rms) and a ccd noise of 9 ADU. As stated in the main text, the two applied threshold factors are always T_1=T_2=3. Only in the case, where the thresholds are varied, the applied values are given in the second last and third last number in the file name. For example in "sim_sigmaCC_0.30_sigmaNoise_9_3-1-5" the first threshold factor is 3 and the second one is 1. For reasons of completeness, the number 5 in the end of the file name indicates the length of the quadratic area used for the G-Fit algorithm.

For further information please feel free to contact the corresponding author.

Table 1: List of variables to adapt the properties of the simulated CCD frames in simulate_SPEs.py of the zip-archive provided as supplementary material.

Line	Python variable	Description
71	-	ccd extensions in pixels
78	pixelh	horizontal pixel size in µm
79	pixelv	vertical pixel size in µm
168	no_frames	number of ccd frames to simulate
169	no_photons	number of photons to simulate per frame
195	path_all	directory to save the simulated data
196	s_CC_default	default value for σ_{cc} if σ_{dark} is varied
197	s_N_default	default value for σ_{dark} if σ_{cc} is varied
203	sigmaCC	varied values for σ_{cc}
204	sigmaNoise	varied values for σ_{dark}