Deutsches Elektronen-Synchrotron
DESY

Photon Science Detector group
FS-DS

Summer school report

LAMBDA background radiation image processing

Auther:
Francesca Pellegrino

Supervisors:
David Pennicard, Vahid Rahmani, Shah Nawaz

09 September 2021

Abstract

The development of next generation multi-mega pixel detectors produce data at such a high frequency that it is no longer possible to process them offline after the storing phase. Moreover in some experiments only 5-10% of the total images generated are qualitatively relevant. Therefore, the DESY's Photon Science Detector Group investigates optimal strategies for online data processing and reduction. This report describes the low preprocessing techniques developed on the LAMBDA background radiation images and how they can be performed to allow the detection and cataloguing of different images' features using segmentation.

Contents

1 Introduction

With the development of next generation multi-mega pixel detectors for DESY synchrotron and Free-Electron Laser experiments will be generating multi terabits/sec of data. This means it is no longer possible to use the classical approach of first storing the data for later offline processing. Moreover, during a traditional experiment workflow all images are saved to disk while often only 5-10% of them (lower than 0,1% in the worst cases) are qualitatively relevant [2]. Therefore, the Photon Science Detector Group (FS-DS Group) has started a large project to investigate optimal strategies for online data processing and reduction. A special emphasis is on investigating the relevance of machine learning techniques. The summer school project was carried out by two students, each focusing on a particular aim. This report shows the low preprocessing techniques developed on the LAMBDA (Large Area Medipix3 Based Detector Array, X-ray area detector [3]) background radiation images. LAMBDA is a hit-counting detector which means if a pixel is working properly then its value should correspond to the number of particles that hit it during the acquisition time. It will be explained how preprocessing techniques can be performed to allow the detection and cataloguing of different features present in the images using segmentation. The images of background radiation utilized are taken with different times per image: 100ms, 10s and 300s. The Python code is available at https://github.com/FrancescaPellegrino/DESY-SUMMER-STUDENT-PROGRAMME-2021

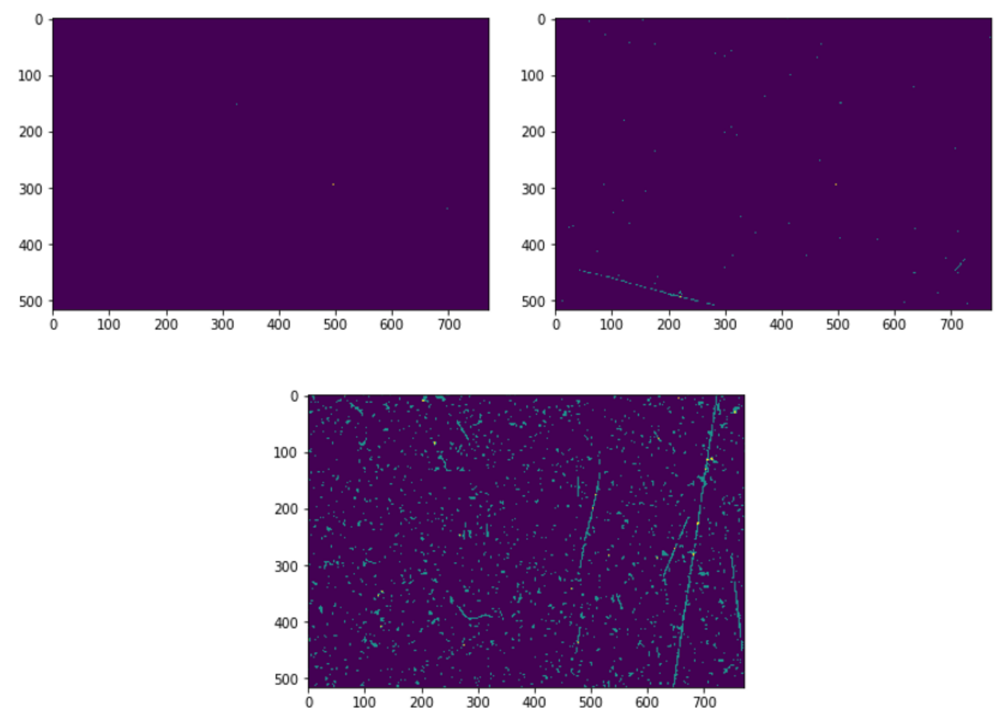


Figure 1: examples of LAMBDA images (100ms on up left, 10s on up right, 300s on left)

2 Background knowledge

Image processing is a set of methods to perform operations on an image to manipulate it or extract some information. Features recognition is allowed by image manipulations which improve image quality reducing noise and simplify data access. In order to execute any process it is necessary to insert the image inside a coordinate system which divides it into many little sections called pixels. Each of them is defined by its position and intensity. Many techniques demand a small matrix of pixels called *structuring element* or *kernel* to probe the image. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels. Considering an image it said that a structuring element "fits" if there is a positive value correspondence for all the pixels, it said it "hits" if there is an intersection and so at least one pixel correspondence, finally it said it "misses" if there is no correspondence. From the result of these comparisons each technique operates in a different way. Furthermore, the results of these techniques are influenced both by the size and shape of the kernel.

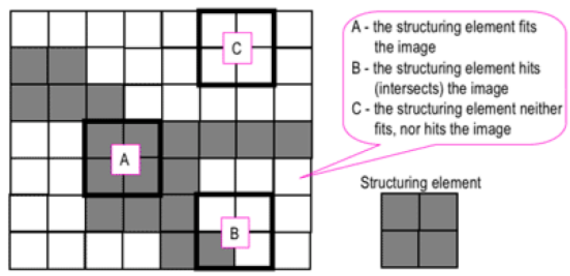


Figure 2: Probing of an image with a structuring element [4]

A common practice is to have odd dimensions of the structuring matrix and the origin defined as the centre of the matrix. The figure 3 helps to better understand how the structuring element defines the neighbourhood of the pixel of interest, which is circled. The output value changes depending on the technique applied.

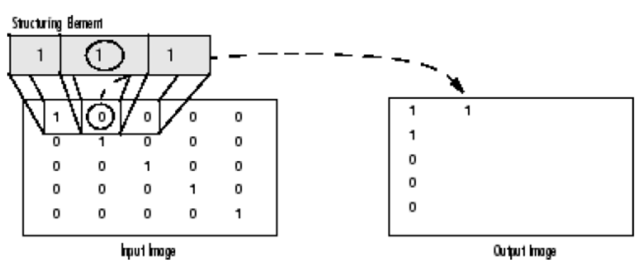


Figure 3: Image processing of a binary image [4]

List below some of the fundamental operations shown in the case of binary images (0 as black, 1 as white) for easier comprehension. These are easily expandable to grey-scale or coloured images.

3 Morphological operations

In an image morphological operations act on the shape and structure of features contained in it.

The **dilation** adds a layer of pixels to both the inner and outer boundaries of regions. This makes objects more visible, small holes are filled and gaps between different regions become smaller. The value of the output pixel is the maximum value of all pixels in the neighbourhood. The dilation operator takes two pieces of data as inputs: the first is the image which is to be dilated, the second is the structuring element. It is this kernel that determines the precise effect of the dilation on the input image. A larger kernel produces a more extreme dilation effect, although usually very similar effects can be achieved by repeated dilation using a smaller but similarly shaped structuring element. With larger structuring elements, it is quite common to use an approximately disk shaped kernel, as opposed to a square one. Dilation can be made directional by using less symmetrical structuring elements. e.g. a structuring element that is 10 pixels wide and 1 pixel high will dilate in a horizontal direction only. Similarly, a 3×3 square structuring element with the origin in the middle of the top row rather than the center, will dilate the bottom of a region more strongly than the top. Dilation can also be used for edge detection by taking the dilation of an image and then subtracting away the original image, thus highlighting just those new pixels at the edges of objects that were added by the dilation. [6].

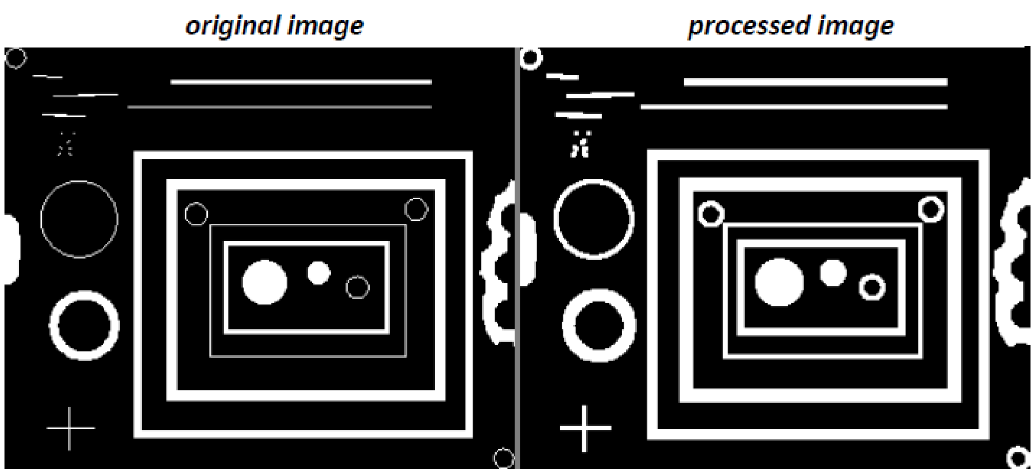


Figure 4: Dilation [4]

The **erosion** acts oppositely to the dilation, removing a layer of pixels from both the inner and outer boundaries of regions. Holes and gaps between different regions become larger and small details are eliminated. The value of the output pixel is the minimum value of all pixels in the neighbourhood. The erosion operator takes two pieces of data as inputs: the first is the image which is to be eroded, the second is the structuring element. It is this kernel that determines the precise effect of the dilation on the input image. Erosion is the dual of dilation, i.e. eroding foreground pixels is equivalent to dilating the background pixels. A larger structuring element produces a more extreme erosion effect, although usually very similar effects can be achieved by repeated erosion using a smaller similarly shaped structuring element. With larger structuring elements, it is quite common to use an approximately disk shaped structuring element, as opposed to a square one. Like dilation, also erosion can be made directional by using less symmetrical structuring elements. For example, a structuring element that is 10 pixels wide and 1 pixel high will erode in a horizontal direction only. Similarly, a 3×3 square structuring element with the origin in the middle of the top row rather than the center, will erode the bottom of a region more severely than the top. Erosion can also be used for edge detection by taking the erosion of an image and then subtracting it away from the original image, thus highlighting just those pixels at the edges of objects that were removed by the erosion[7]

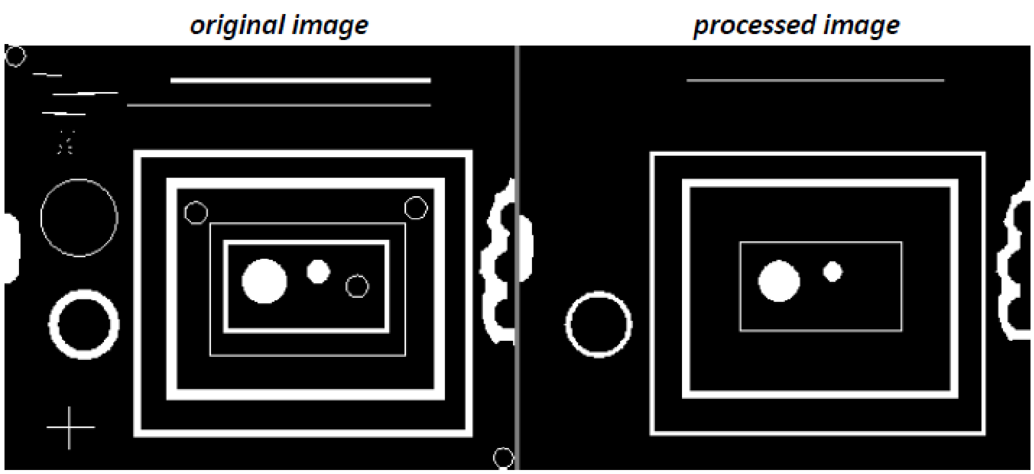


Figure 5: Erosion [4]

Many morphological operations are built as a composition of simpler techniques.

The **opening** is an erosion followed by a dilation using the same structuring element for both operations. Opening acts taking the structuring element and sliding it around inside each foreground region, without changing its orientation. All pixels which can be covered by the structuring element with the structuring element being entirely within the foreground region will be preserved. However, all foreground pixels which cannot be reached by the structuring element without parts of it moving out of the foreground region will be eroded away. After the opening has been carried out, the new boundaries of foreground regions will all be such that the structuring element fits inside them. This technique is useful for removing thin objects while preserving the shape and size of larger objects in the image. The opening operator requires two inputs: an image to be opened, and a structuring element [8]. Opening is an idempotent operation: once an image has been opened, subsequent openings applied to the same structuring element have no further effect on that image [4]. The basic effect of an opening is somewhat like erosion in that it tends to remove some of the foreground pixels from the edges of regions of foreground pixels. However it is less destructive than erosion in general. As with other morphological operators, the exact operation is determined by a structuring element. Unlike erosion and dilation, the position of the origin of the structuring element does not really matter for opening, the result is independent of it [8].

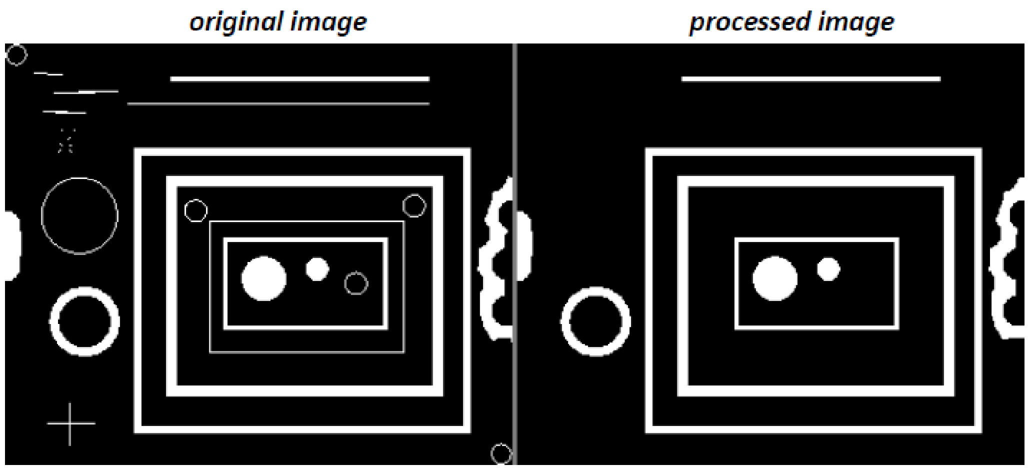


Figure 6: Opening [4]

The **closing** is opening performed in reverse. It is a dilation followed by an erosion, which fills holes keeping the initial region sizes. As opening, closing is idempotent [4]. Closing is similar in some ways to dilation in that it tends to enlarge the boundaries of foreground regions in an image (and shrink background color holes in such regions), but it is less destructive of the original boundary shape. As with other morphological operators, the exact operation is determined by a structuring element. Closing acts taking the structuring element and sliding it around outside each foreground region, without changing its orientation. For any background boundary point, if the structuring element can be made to touch that point, without any part of the element being inside a foreground region, then that point remains background. If this is not possible, then the pixel is set to foreground. After the closing has been carried out the background region will be such that the structuring element can be made to cover any point in the background without any part of it also covering a foreground point. To achieve the effect of a closing with a larger structuring element, it is possible to perform multiple dilations followed by the same number of erosions. Closing can sometimes be used to selectively fill in particular background regions of an image. Whether or not this can be done depends upon whether a suitable structuring element can be found that fits well inside regions that are to be preserved, but doesn't fit inside regions that are to be removed. Unlike erosion and dilation, the position of the origin of the structuring element does not really matter for closing, the result is independent of it [9].

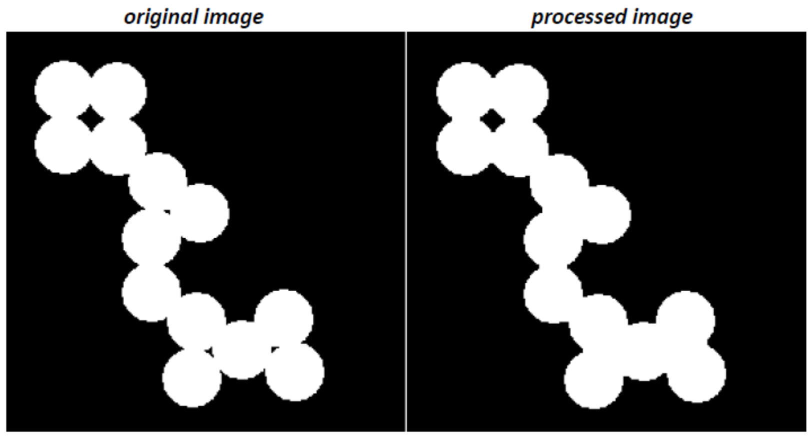


Figure 7: Closing [4]

4 Noise and Edges

Frequently the images taken from detectors or other experiments suffer from the presence of **noise**, which can be caused by intrinsic or extrinsic factors. Image noise is an undesirable random variation of brightness or colour information in pixels that obscures the desired information [3]. Denoising or at least reducing the noise in an image allows for a clearer analysis of its contents and an easier manipulation for subsequent image processing.

There are some different image filters that can provide noise reduction. A first example is the median filter which replaces each pixel intensity with the median of the kernel neighbouring ones. A Gaussian filter take a weighted average of the pixel values but the kernel weights are mainly assigned to the centre and less in the periphery, exactly as described by a Gaussian distribution. Finally, the functioning of a bilateral filter is similar to the Gaussian one. In this case the weight depends on both the geometric distance between pixels and their photometric similarity. It is useful to choose the image manipulation approach based on the main strength of each filter type. For example, the median filter does a better job than other functions of ignoring extreme values whereas the bilateral filter takes photometric similarity into account to reduce the blurring of edges.

The **edge** detection is led by a group of mathematical methods that try to identify changes in gradient of pixels' intensity. The Canny operator works in a multi-stage process. First of all the image is smoothed by Gaussian filter. Then a simple 2-D first derivative operator is applied to the smoothed image to highlight regions of the image with high first spatial derivatives. Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as non-maximal suppression. The tracking process exhibits hysteresis controlled by two thresholds: T1 and T2, with T1 > T2. Tracking can only begin at a point on a ridge higher than T1. Tracking then continues in both directions out from that point until the height of the ridge falls below T2. This hysteresis helps to ensure that noisy edges are not broken up into multiple edge fragments. The effect of the Canny operator is determined by three parameters: the width of the Gaussian kernel used in the smoothing phase and the upper and lower thresholds used by the tracker. Increasing the width of the Gaussian kernel reduces the detector's sensitivity to noise, at the expense of losing some of the finer detail in the image. Moreover, although the Canny edge detector allows us the find the intensity discontinuities in an image, it is not guaranteed that these discontinuities correspond to actual edges of the object [10].

The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input image. Natural edges in images often lead to lines in the output image that are several pixels wide due to the smoothing effect of the Sobel operator. Some thinning may be desirable to counter this. Failing that, some sort of hysteresis ridge tracking could be used as in the Canny operator [10].

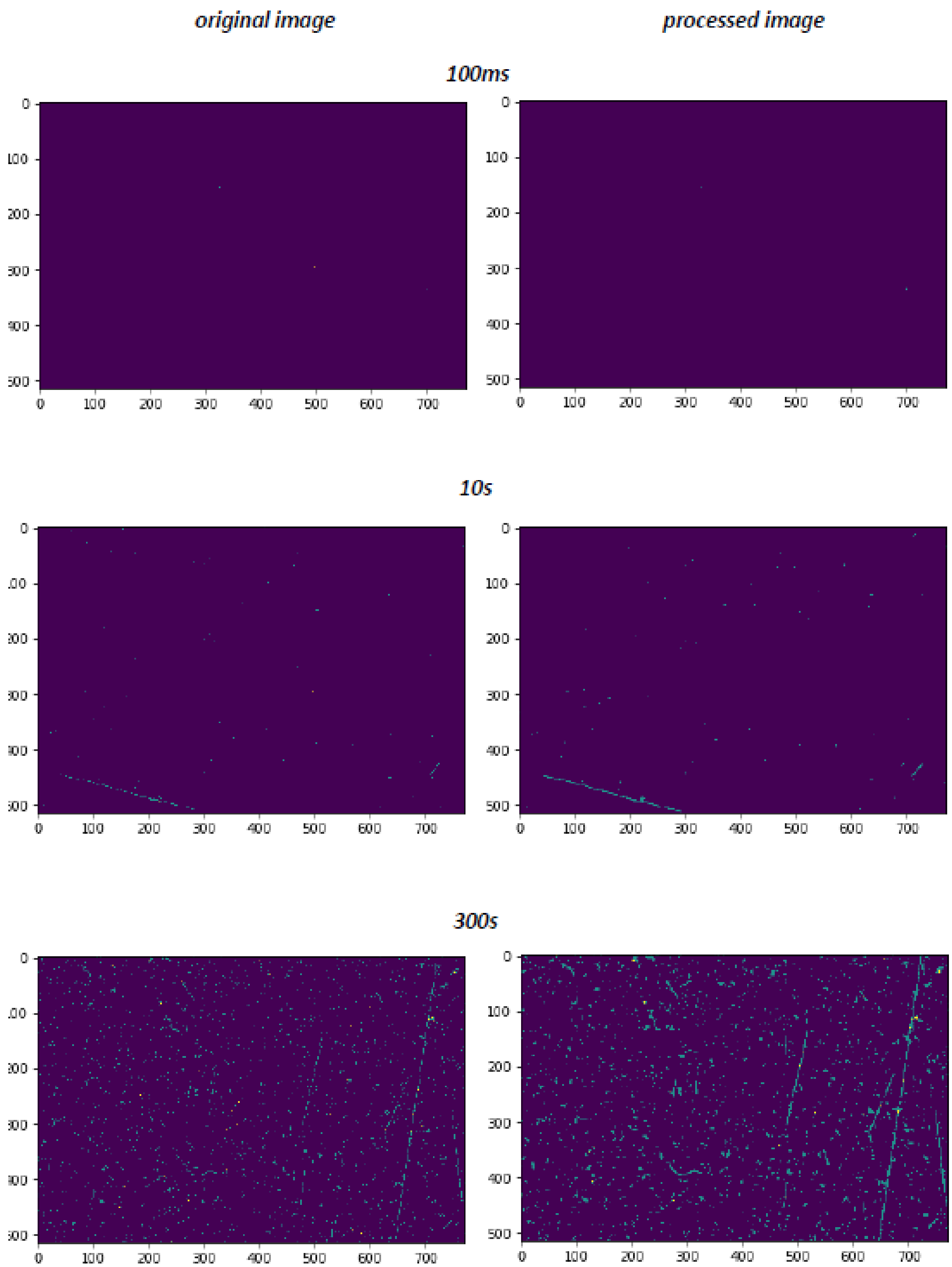
5 Application on LAMBDA background radiation images

5.1 Pre-processing

As discussed so far, preprocessing an image before its analysis allows for an easier access to its contents and feature extraction. The entirety of this summer project is based on the background radiation images taken from the LAMBDA detector. Different combination of techniques were tested and we selected a group of three of them for a satisfying result:

1. Dilation (with 2x2 kernel)
2. Closing (with 2x2 kernel)
3. Median blur (with 3x3 kernel)

The point at issue of this preprocess is the connections of non continuous lines to detect them as a feature. For this reason the first dilation facilitates the respective approach of segments' extremities. The following closing allows to fill holes keeping the region sizes. Finally, a median blur acts especially on tiny spots which return to dimensions similar to the initial ones. Some example follow.


original image
processed image
100ms
10s
300s

5.2 Feature detection

Once preprocessed images are ready for features extraction. Considering a physical interpretation it is possible to distinguish three different categories of image contents:

- straight lines related to high-energy particles
- small clusters related to X-ray background
- bad pixels (yellow in our images) related to no response or noise

For a more specific analysis it started leading a point and line detection. The first recognizes and highlights pixels with a specific value or range value.

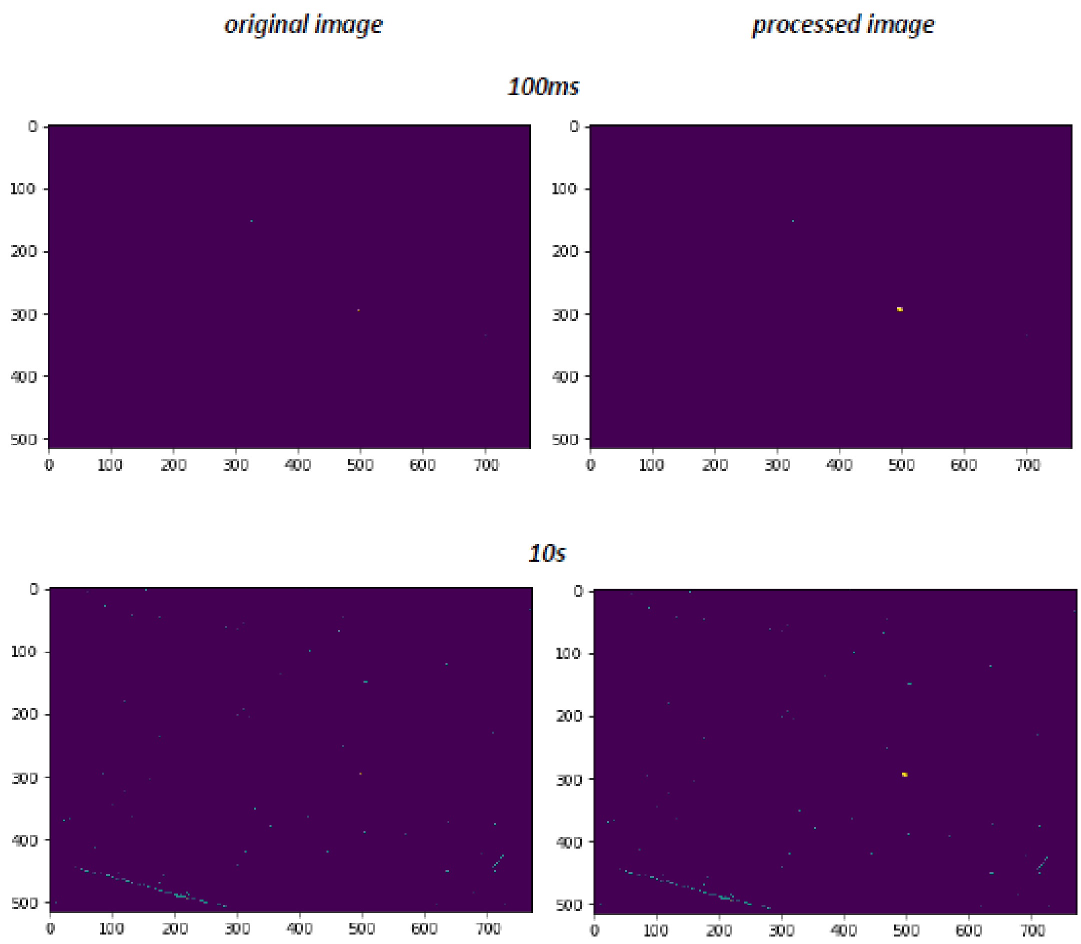


Figure 8: point detection LAMBDA images comparison (previous page 100ms [pixel's value 12]; 10s [pixel's value 50])

For line detection it uses the Hough Transform. In this manner each line is defined by its distance by the axis origin and the angle of inclination. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise. The Hough transform can be used to identify the parameters of a curve which best fits a set of given edge points. The work of the Hough transform is to determine both what the features are (i.e. to detect the features for which it has a parametric (or other) description) and how many of them exist in the image [11].

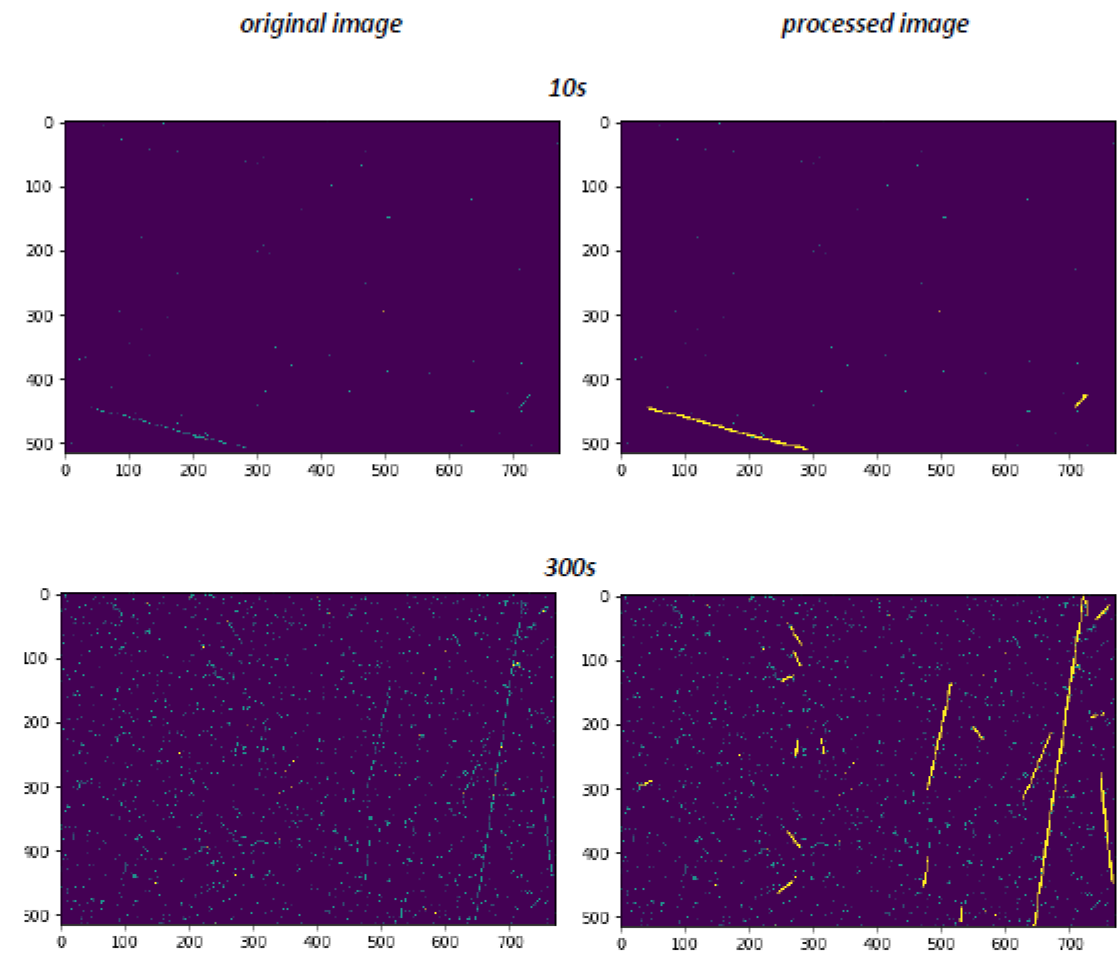


Figure 9: line detection LAMBDA images comparison (from top 10s, 300s)

5.3 Feature extraction: Labeling blob

Finally, the segmentation technique provides more general and effective results. The our goal of this phase is to detect connected components in any manner they appear, so it is better to say that we aim to operate a **labeling blob** technique. Bearing in mind the horizontal flow typical of a scan line algorithm, this methodological approach is based on an orderly analysis of each pixel, for example probing the image from top to bottom, from left to right.

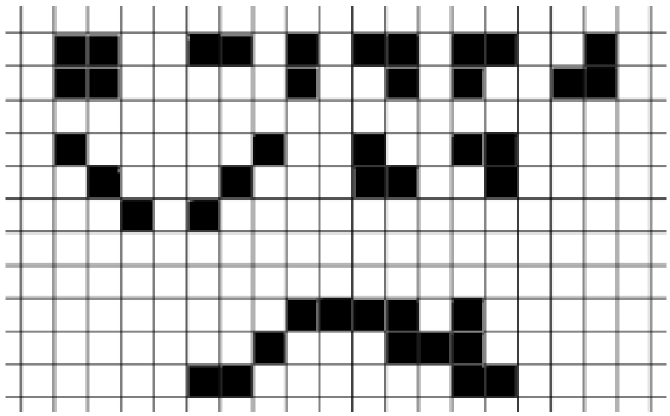


Figure 10: example of possible combination of connected components [1]

To each pixel are assigned some values:

- the pixel number
- (X, Y) coordinate of its position
- flag 0-1
- the group number to which the pixel belongs

The pixel number is unique for each pixel and it grows following the orderly probe. The flag is initially set to 0 which means "unprocessed" and after the pixel is processed the flag changes to 1. The group number is assigned on the basis of neighbour analysis. Indeed, for each pixel 4 neighbouring pixels are considered as shown in figure 13 and it is evaluated if they have no background value. In this way it is possible to recognize groups of pixels which are connected and assign them an equal group number. Regarding always four neighbour pixels in the specific position as the centre one, allows to optimise the probe covering all the image.

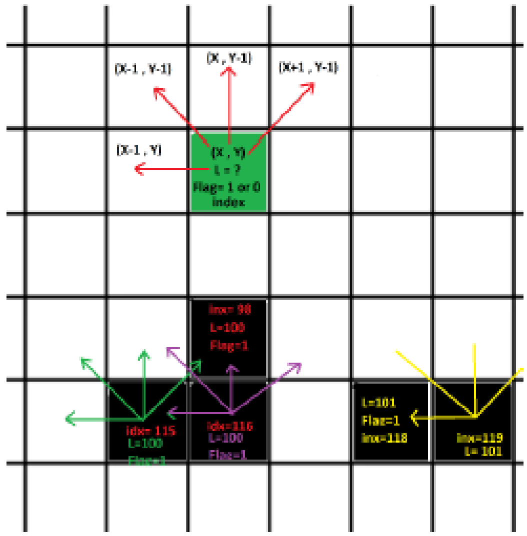


Figure 11: example of segmentation process [1]

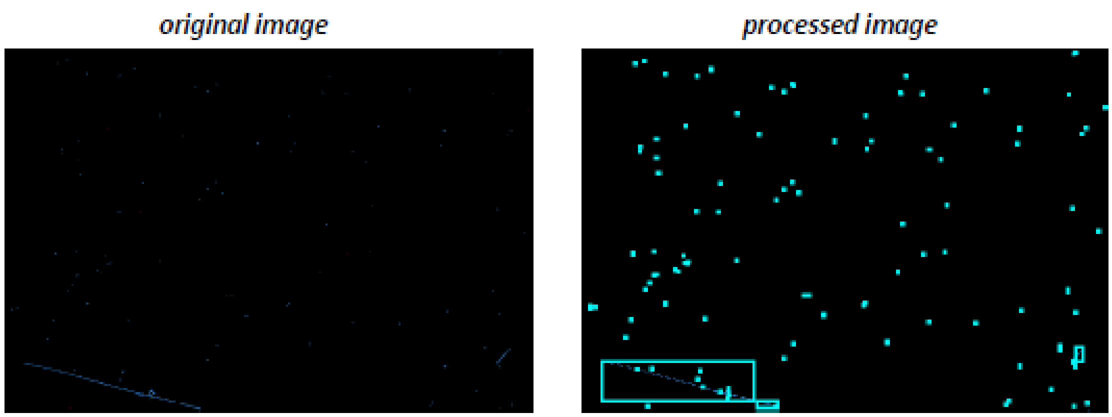


Figure 12: segmentation LAMBDA images comparison (10s) [1]

Once the entire scan is complete, information about the type of the feature considered and the number of pixels is easily extractable through the group number. In doing so the Hough Transform for the line detection becomes no longer necessary. A further possible operation is to use a threshold on the number of pixels inside each group to distinguish between straight lines and small clusters. An additional advantage of labeling blob it is easily organisable as a parameter free technique, depending only by the image's sizes.

6 Conclusion and Future work

This report is a brief contribution to the FS-DS Group project. The method defined until now can be the first step in speeding the data reduction process simplifying the feature extractions. One possible next improvement is continue the development of this process to allow the features categorisation and a wider distinction of different types of radiation.

7 References

[1] created and shared by Vahid Rahmani

[2] lab talk on "Data Reduction for Photon Science" by Vahid Rahmani and Shah Nawaz

[3] D. Pennicard et al., "The LAMBDA photon-counting pixel detector and high-Z sensor development", Journal of Instrumentation 9, C12026, 2014. doi:10.1088/1748-0221/9/12/C12026

[4] https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm (accessed September 07, 2021)

[5] https://se.mathworks.com/help/images/morphological-dilation-and-erosion.html (accessed September 07, 2021)

[6] https://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm (accessed September 07, 2021)

[7] https://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm (accessed September 07, 2021)

[8] https://homepages.inf.ed.ac.uk/rbf/HIPR2/open.htm (accessed September 07, 2021)

[9] https://homepages.inf.ed.ac.uk/rbf/HIPR2/close.htm (accessed September 07, 2021)

[10] https://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm (accessed September 07, 2021)

[11] https://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm (accessed September 07, 2021)