CCD matrix analysis

Alessandro Stecchi (INFN-LNF)

Exercise

The purpose of this practical exercise is to determine the FWHM of the projection of the 3D beam spot intensity profile along both the X and Y axes.
Assumpions: the projection of the beam spot intensity profile, along both the X and Y axes, is gaussian.

Algorithm

Many different approaches, with varying levels of accuracy, are possible. Here, we present a very simple one that is quite effective from a didactic point of view.
From the 2D matrix obtained via CCD readout (fig. 1), it is clear that each row contains the luminosity values of the beam spot along that pixel slice.

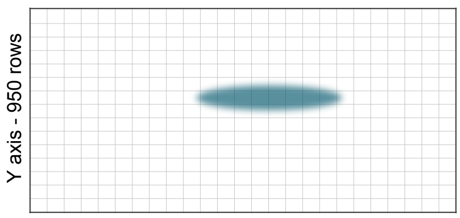

Fig. 1 - Pictorical representation of the CCD matrix with 950 rows and 1200 columns. The ellipse simulates a top view of the beam spot, with the numerical value of each pixel proportional to the beam spot's light intensity at that point.

In other words, each row represents a section of the 3D Gaussian beam spot intensity profile along the X-axis, and the highest of these sections gives the desired projection along that axis. The algorithm essentially involves a loop that iterates through the 2D matrix row-by-row, searching for the maximum value in each row (fig. 2). At each iteration, it is worth removing some noise by applying a low-pass filter to the array (this way, we avoid mistaking a spike of noise for the actual peak of the i-th Gaussian).

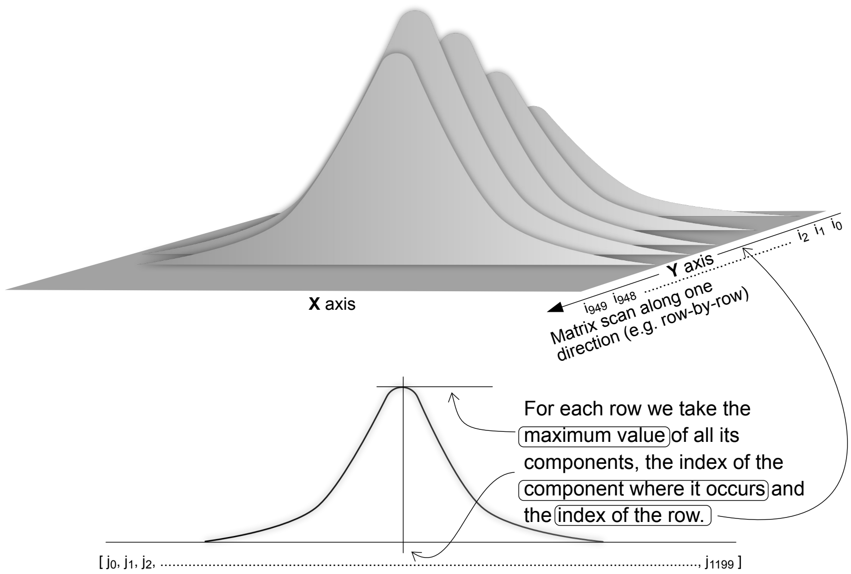

Fig. 2 - Iteration row-by-row of the 2D matrix

After scanning the matrix row-by-row, we obtain a table like the one shown in Fig. 3.

We seek the maximum of all maxes M_M and extract the corresponding j_M and i_M

Max value of the i-th row components	Index of the component where the max occurs [0-1199]	Row index [0-949]
Max_0	j_{Max0}	0
Max_1	j_{Max1}	1
Max_2	j_{Max0}	0
...	...	...
...	...	...
Max_{949}	j_{Max949}	949

Fig. 3 - The highlighted row corresponds to the highest Gaussian and gives the X (j_M) and Y (i_M) coordinated (in pixels units) of the beam spot center.

Referring to table in fig. 3, the row $\mathbf{i}_M$ is the one that contains the highest Gaussian curve, that corresponds to the highest section, parallel to the X-axis, of the 3D beam spot profile.
At this point we can perform a Gaussian fit[1] on the row $\mathbf{i}_M$ and work out its $\boldsymbol{\sigma}$, which is mathematically related to the FWHM.
Regarding the projection along the other axis, one option is to transpose the matrix and repeat the whole procedure. Alternatively, we can use the index $\mathbf{j}_M$ from the table in Fig. 3, which obviously corresponds to the *column* where the maximum value was found. Than, we can perform a Gaussian fit on that *column* and work out its $\boldsymbol{\sigma}$

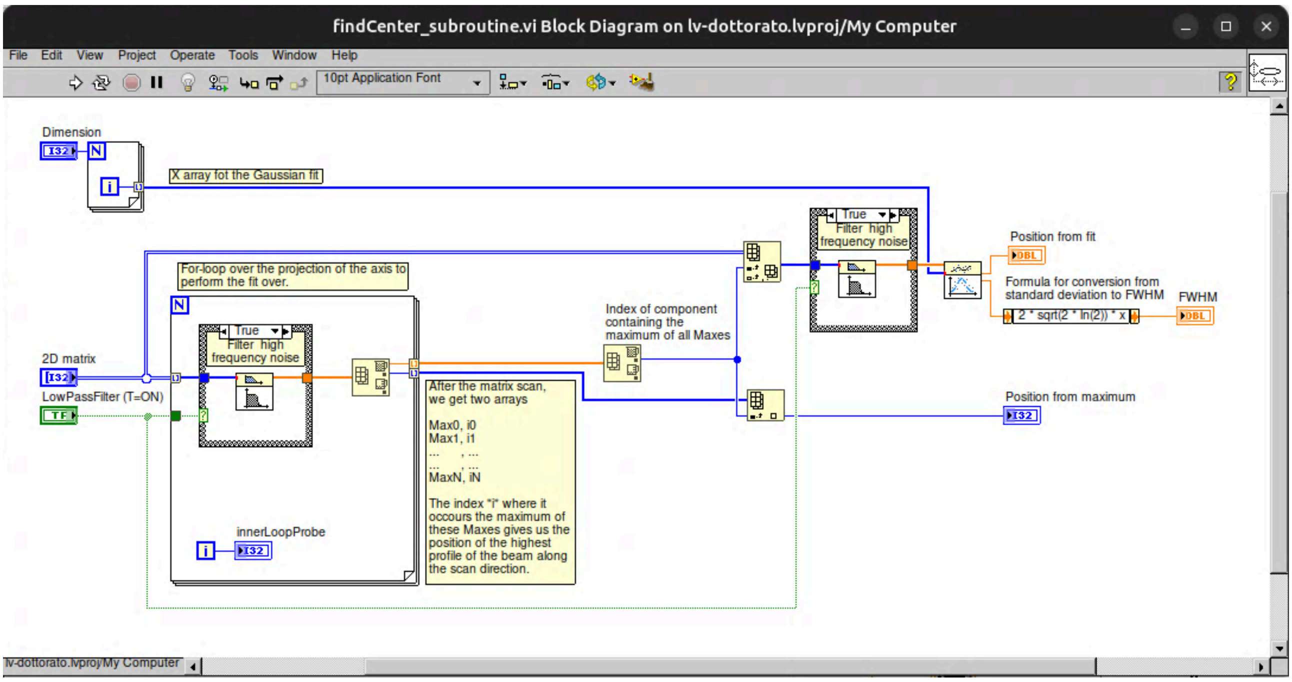

Fig. 4 - Algorithm in G (LabVIEW® graphical language) corresponding to the described procedure

[1] The Gaussian fit further reduces noise.

Data matrix format

For those wanting to develop their own algorithm (even in other languages like Python), a file (`BEAMPOS_good.dat`) is available on the INFN Agenda webpage in the April 15th slot (for both groups A and B), containing the data in the same format used during the laboratory session. The 950 x 1200 matrix is written as a sequence of bytes (fig. 5). The first four bytes constitutes a 32 bit integer that counts the number of rows (950) and the following four bytes another 32 bit integer that counts the number of columns (1200). Then, the subsequent bytes are to be interpreted in pairs as 16-bit integers, where the value represents the brightness level of the pixels.

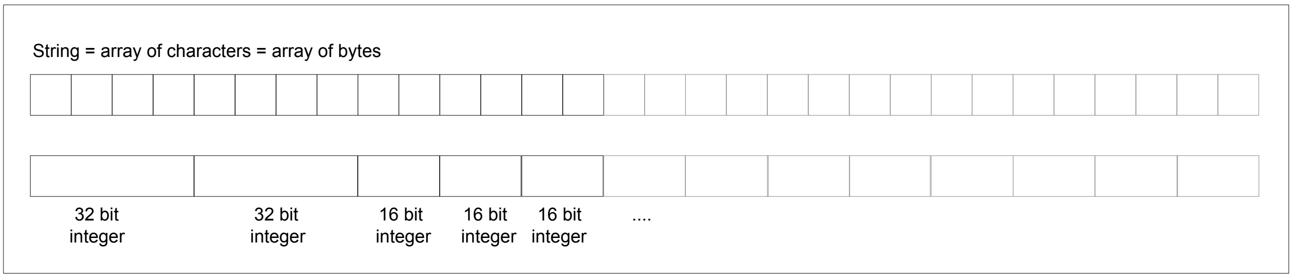

Fig. 5 - The matrix is written in the file as a single continuous ASCII string. As the figure illustrates, a string (a sequence of ASCII characters) is equivalent to a sequence of bytes that, in our case, must be interpreted according to the described pattern.

Setup

For the lab activity, we worked on workstations consisting of Intel® NUC Mini PCs with i5 quad-core processors, 500 GB SSD, 16 GB RAM, and a 27" monitor with 2K resolution. The programming environment used was LabVIEW®[2] v 2023 Q1 professional edition. We used a ready-made event-based main program template.

2 LabVIEW® is a graphical programming environment that allows an intuitive approach to programming and provides a fully integrated user interfaces. LabVIEW (formerly owned by Nationa Instruments) is a EMERSON product. See https://www.ni.com