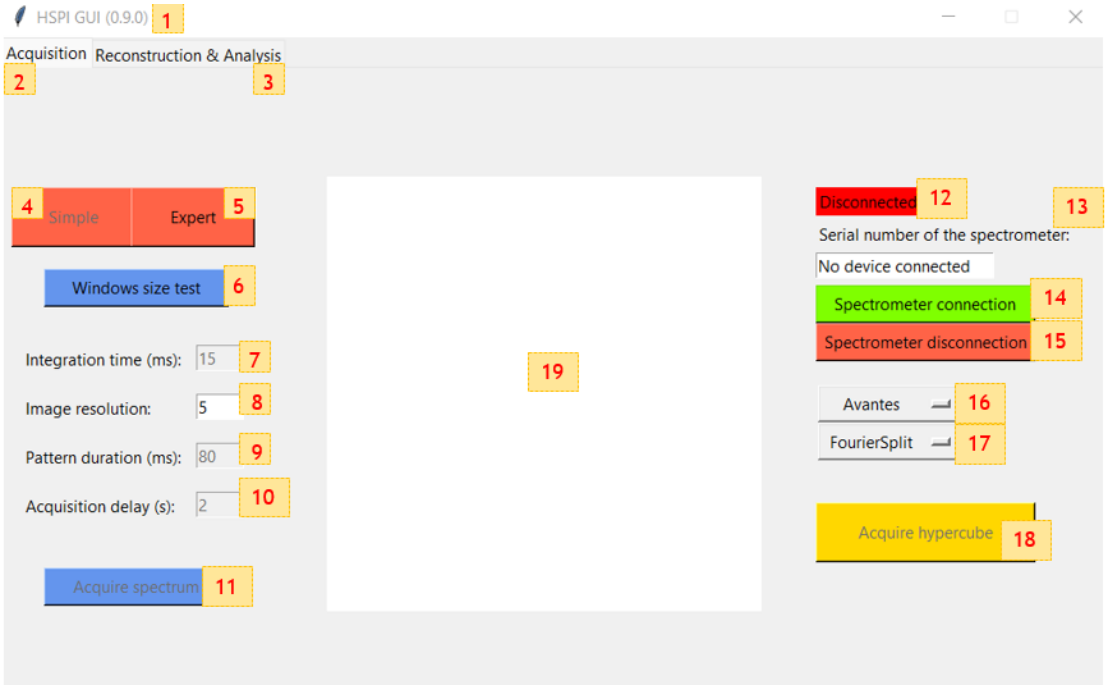
ONE-PIX GUI (1.0.0) user manual :

This GUI allows a user wishing to use ONE-PIX technology to simply make acquisitions and reconstruct them. It consists of two widgets. One to perform hypercube measurements and the other to reconstruct and analyse the results.

- Running Python script « ONEPIX_app.py » creates the graphical interface

Acquisition widget :



First of all, it is always necessary to indicate the type of spectrometer[16] used, the pattern basis [17] to be projected, the spatial resolution [8] of the desired hypercube to be reconstructed and to connect the spectrometer by pressing the spectrometer connection button [14].

The Acquisition widget has two operating modes : a simple mode [4] and an expert mode [5].

The **simple mode** allows measurements to be made automatically without worrying about measurement parameters by pressing the **Acquire Hypercube** button [18].

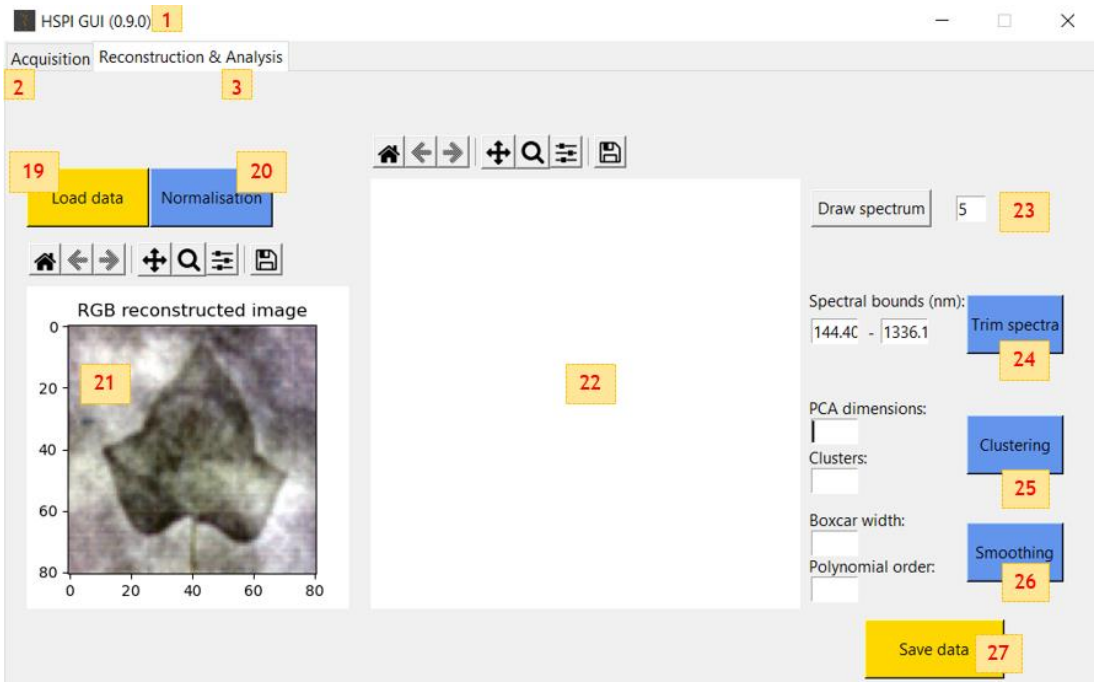
Once all the parameters have been determined, a window is displayed to show an estimate of the duration of such a measurement.

If this duration is too long, it is possible to reduce it by decreasing the initial spatial resolution requested or by changing the acquisition parameters by switching to the expert mode.

By clicking on the **expert mode**, the inputs are unlocked and it is thus possible to modify the measurement parameters manually.

- The **integration time** [7] is that of a unit measurement made by the spectrometer expressed in milliseconds. In order to manually determine a relevant integration time, it is possible to display any Fourier pattern by choosing its resolution with the **Windows size test** button [6]. In order to optimise the measurement in terms of signal-to-noise ratio, an integration time should be chosen so that the intensity of the maximum value of the measured spectrum is as close as possible to 40000 counts. However, if this involves too long an acquisition time, it is possible to make trade-offs between SNR and acquisition time.
- The **Acquire spectrum** [11] button is used to measure a spectrum that will be displayed in the graphics area.
- **Pattern duration** [9] is the display time of each pattern. This duration must be greater than the integration time and preferably greater than 80 ms. The spectrometer measures on running mode during the acquisitions so that each spectrum measured during the pattern duration can be averaged.
- Finally, the **acquisition delay [10]** is a delay before running an hypercube acquisition.
- Select the **Acquire hypercube** [18] button to start the acquisition.

Reconstruction & Analysis widget :



- By clicking on the **Load data** button [19], a message box ask if the data to load were already analysed using this app or not. Then a window opens to select the folder containing the hypercube data to be reconstructed for raw data or a «. npy » file containing a dictionnary including analysed data. Finally, a preview of the hypercube appears in RGB format in the graphical area [21].

- The **Normalisation** button [20] is used to perform a relative reflectance normalisation with respect to the average of a user-selected area of pixels in the image. a window opens and allows you to select the two opposite corners of an area of the hypercube. In order to normalise the hyperspectral data correctly, it is advisable to use calibrated reflectance surfaces.

- On the right side of the interface, it is possible to measure a user-defined number of spectra of the hypercube with the **Draw spectrum** button and the associated entry [23]. A window displaying the RGB image opens and allows the user to click on the pixels whose spectra they wish to view.

- **Clustering** button [25] allows to realize classification of the hypercube using kmeans algorithm on principal component of the principal components analysis. To learn more see https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

- **Smoothing** button [26] allows to smooth spectra of a whole hypercube using Savitzky-Golay filter. See more at : https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.savgol_filter.html

- Finally, it is possible to save hyperspectral data using button [27]. RGB images, clustered images and spectra can be directly saved using diskette icons of graphical areas [21] and [22] or from their matplotlib window.

Note : To go further in the normalisations and analyses of the measured hypercubes, it is necessary to use Python scripts and functions in command line.