Laboratory ultrasound scanner displayer

Object:	Design brief of the laboratory ultrasound scanner displayer			
Reference:	DB_laboratory_tool_v2_en			
Revision:	v1	2018/02/28	JED	Creation
	v2	2018/05/30	JED	Modifying image

Written by	Checked by	Validated by
Name: Jérôme Dubois Function:	Name: Function:	Name: Function:
Date: Signature:	Date: Signature:	Date: Signature:

Table of contents

Table of tables

Table of figures

List of abbreviations

sps: sampling per seconds

ADC: analog to digital converter

GUI: graphical user interface

Reference and applicable documents

Reference documents

Here we list all the documents that are need to build the software.

Reference	Title	Date	Owner
RD1	Communication_Protocole.odt	01/03/2018	echOpen
RD2	Signal processing (in preparation)		
RD3	Image processing (in preparation)		

Table 1: List of reference documents

Applicable documents

Here we list all the documents relative to the standards.

Ref	Reference	Year	Title	Owner

Table 2: List of applicable documents

Contacts

The main contributer of this tool and contact point is

echOpen
Mr Dubois Jérôme
Main contact
jerome@echopen.org

1 Introduction

1.1 Presentation of the project

echOpen is an open and collaborative project and community, led by a multidisciplinary core of experts and senior professionals with the aim of designing a functional low-cost (affordable) and open source echo-stethoscope (ultrasound scanner) connected to a smart-phone, allowing the radical transformation of diagnostic orientation in hospitals, general medicine and medically under-served areas. This initiate is aimed for health professionals from southern and northern countries.

In this consultation document, we describe only the electronic card design and manufacturing needed for this ultrasound scanner, in order to obtain a first integrated device within a time scale of 6 month.

1.2 Ultrasound scanner functioning

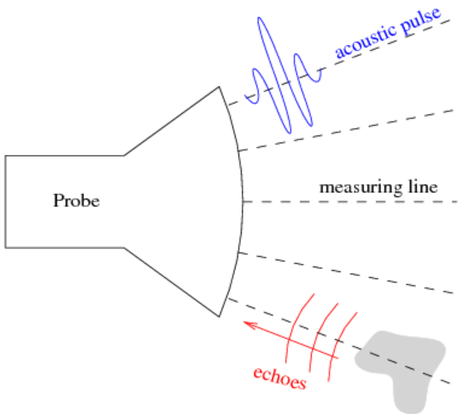

Figure 1.1: Ultrasound scanner principle

An ultrasound scanner is a medical imaging device that makes images of the inside of the body with ultrasound wave in the mega-Hertz range, typically between 1 and 10 MHz. The accuracy of the image increases with the frequency, as the depth of penetration decreases.

The image is composed of a cross-section of the body in front of the probe. To make the image, the probe of the ultrasound scanner sends a short acoustic pulse into the body. This pulse stays enclosed in a collimated beam, so the measurement is made on a line. The ultrasound pulse is generated by one (or many) transducer(s). The measurement of the echo-signal is also made by this (these) transducer(s).

At each interface (between different organs) and each time the pulse encounters a small scatterer (muscle fibers for example), echoes are scattered back to the probe and measured by it. By making different measuring lines on a plan, we can rebuild an image. The ultrasound image is the envelope of the echo-signal measured by the probe displayed in gray scale.

This document detail the functioning and all the tools needed in the software that will determine and display the ultrasound scanner image.

1.3 The echo-stethoscope

An echo-stethoscope is a simple low-cost ultrasound scanner, not as efficient as the portable ultrasound devices in the hospitals but accurate enough to assess some pathologies. It's a diagnosis-helping device, which will be part of the clinical examination.

To save costs, our echo-stethoscope won't have any screen, the image will be displayed on a smart-phone. Moreover, we will use mono-element transducers instead of the multi-element transducers classically used in current ultrasound scanners so the electronic is simpler. In order to have an universal ultrasound scanner, we will use three different transducers at 3.5, 5 and 7.5 MHz.

The image created is a 60° circular sector composed of 64 up to 128 measuring lines.

The circular sector will be achieved by a given mechanism that will change the direction of the acoustic wave homogeneously in the sector.

1.4 Classifications

The echo-stethoscope will be a medical device of Class Iia. This software is laboratory software that will only be used in the echOpen laboratory, so it doesn't need any medical classification.

2 Overall functioning

The purpose of this software is to display the ultrasound image made by the probe in real time. This software will communicate with the probe *via* Wi-Fi or *via* USB cable. In the one hand, the software receives continuously the image data from the probe to reconstruct the image. In the other hand, the operator may change at any time some settings to improve image quality or change image parameters.

2.1 Image characteristics

The ultrasound image made by the probe is presented on figure 2.1. The image is a 60° sector, composed of 64 up to 128 lines. The start depth of measurement R0 and end depth of measurement Rf mostly depend on the central frequency of the transducer but can also be modified by the operator.

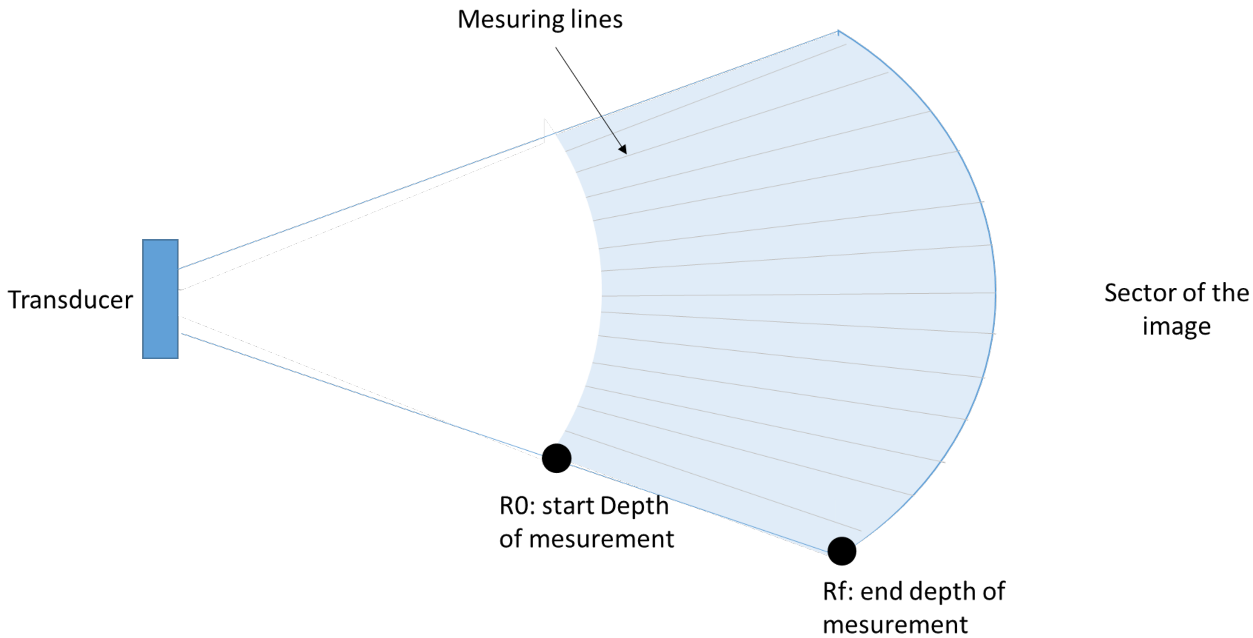

Figure 2.1: Ultrasound image parameters

The image is composed of N lines which are made by rotating the transducer (emitter/receiver) around a given center of rotation. There is an angle dphi between each line that depend on the number of lines.

The software shall be able to display this kind of image knowing that number of lines, R0 and Rf (and sector?) may change and the frame rate can be up to 15 frame per second.

2.2 Software architecture

The architecture of this software is described in figure 2.2, it is composed of 7 principal modules:

- Connexion management: module that checks the connexion between the probe and the display software.
- Settings management: module that manages the settings between the probe and the display module.
- Feedback: module that manages the feedback from the probe.
- Data streaming reception: module that receives the data from the probe sends it the the signal processing module.
- Signal processing: module that applies the signal processing to each measurement line and sends it to image processing module.
- Image processing: module that determines the image.
- Image display: main window of the software where the image is displayed.

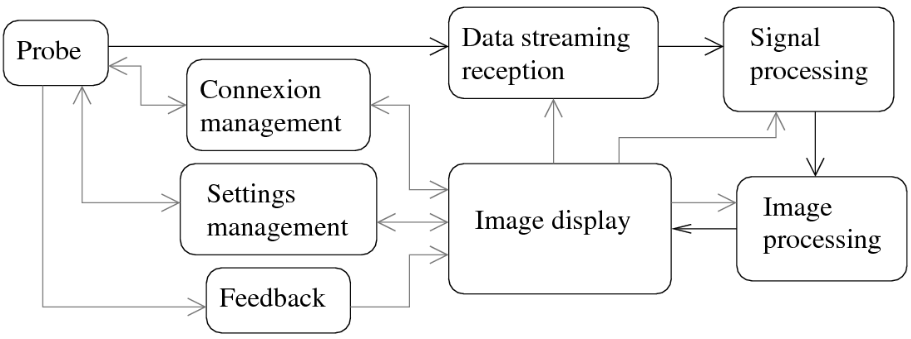

Figure 2.2: Architecture of the software

The black lines represent the flux of the data from the probe to the displayed image. The gray lines represent the flux of settings and some informations such as feedback from the probe.

3 Main window

On the main window, the ultrasound image is displayed in real time.

3.1 Interface

The interface of this window may look what is presented in figure 3.1.

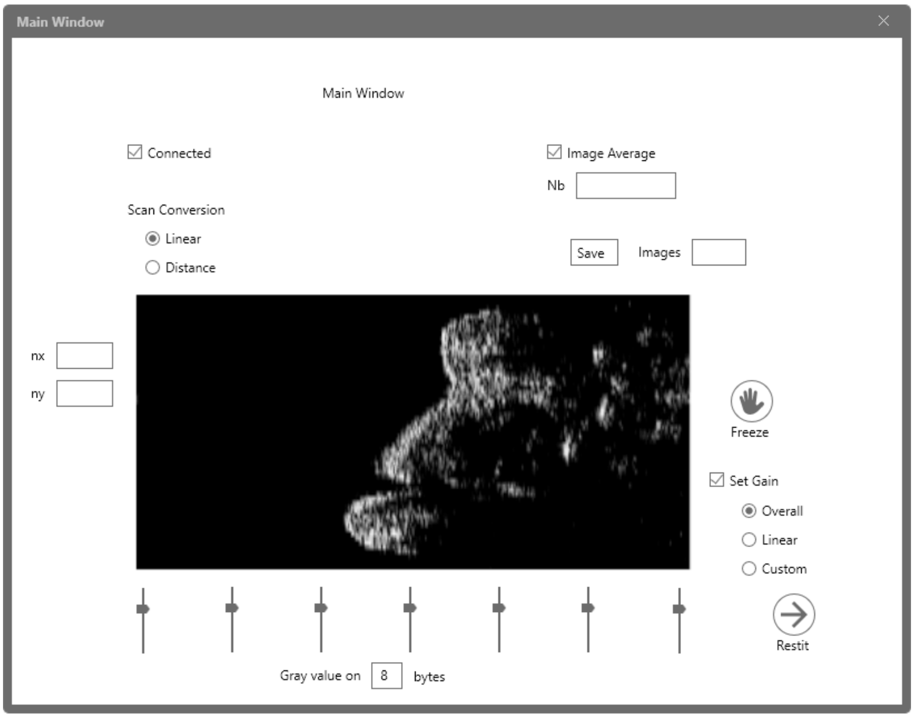

Figure 3.1: Main window interface.

On this window we show the connection status (connected or disconnected), the probe battery (not shown here) and the ultrasound image.

3.2 Settings

The different settings accessible in this window are described in this section.

3.2.1 Scan conversion

The ultrasound image can be displayed with or without scan conversion, this can be chosen with a check box. The scan conversion can be made with linear or distance interpolation, the selection is made with radio button.

When scan conversion is made, attention should be paid to the image must keep the geometry and ratio (there must not had distortion).

A flip option can be added if the image is in the wrong direction along y.

3.2.2 Image size

The default size of the image is nx = 256 pixels and ny = 256 pixels. The size can be change at any time (warning when resizing the image, some calculation must be maid an can take some time, image processing must be stop while these calculus are not finished). Validation of resizing must be made by pushing a button, not each time a value is changed.

When scan conversion is not activate, the default size is number of point per line, included re-sampling rate (following x) and number of line (following y).

3.2.3 Gain management

In this window, we shall be able to tune the gain to improve image quality in real time. There is three way to change the gain:

- overall gain: the gain is same everywhere in the image,
- linear gain: the gain vary continuously from R0 to Rf,
- custom gain: the gain can vary non-monotonically from R0 to Rf.

These three option shall be implemented as well as a reset button to readjust the gain to one everywhere in the image.

3.2.4 Freeze image

We shall add a freeze button on the interface, when cliquing on this button, the image freeze. (We may stop signal processing and image processing during freezing the image, or calculation still running but non refresh the image). When re-cliquing on this button, image display restart classically.

3.2.5 Save image

We shall add a option to save one or many images at any time. To simplify the process, name of the images can be the time stamp, we may enter the prefix of the image and increment the name of the images.

3.2.6 Image average

When activating this image, the display image is the average of N successive images. The interest of this option is that the intrinsic noise in the image will decrease, but the target of the image must move very slowly.

4 Communication with the probe

The connection between the probe and the display software can be made *via* Wi-Fi or *via* USB cable. In a first phase only Wi-Fi is considered, USB communication will be design later. The communication protocol between the probe and the display software is described in RD1.

The communication is handle by the three modules:

- connexion management,
- settings management,
- data streaming reception.

4.1 Connexion management

This module manage the connexion between the probe and the display software.

4.1.1 Interface

This module will have is on window which will look like what is presented on figure 4.1.

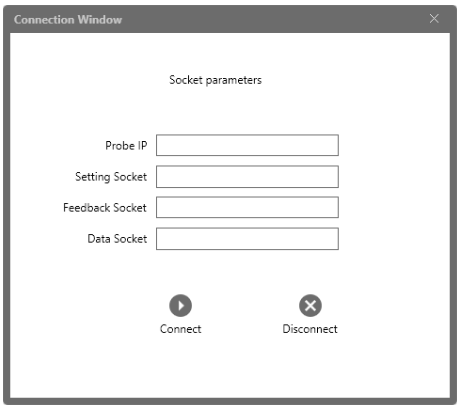

Figure 4.1: Connexion window.

Two buttons are put to connect or disconnect to the probe.

4.1.2 Settings

In this window will enter the IP address of the probe and the socket numbers of each socket. As describe in RD1, there is three sockets: settings socket, feedback socket and data socket. In this window we can connect or disconnect the display software to the probe. Following communication protocol (*cf*. RD1), we first connect to feedback socket to know the status of the probe, then connect to data socket to acquire the informations about the image and finally to settings socket.

In first time we may add the option to connect or not to feedback and settings sockets.

4.2 Settings management

This module manage the modification of the setting of the image made by the operator in display software. In this window, we display and can change all the settings of the probe.

4.2.1 Interface

This module has is own interface that will look like the one presented on figure 4.2.

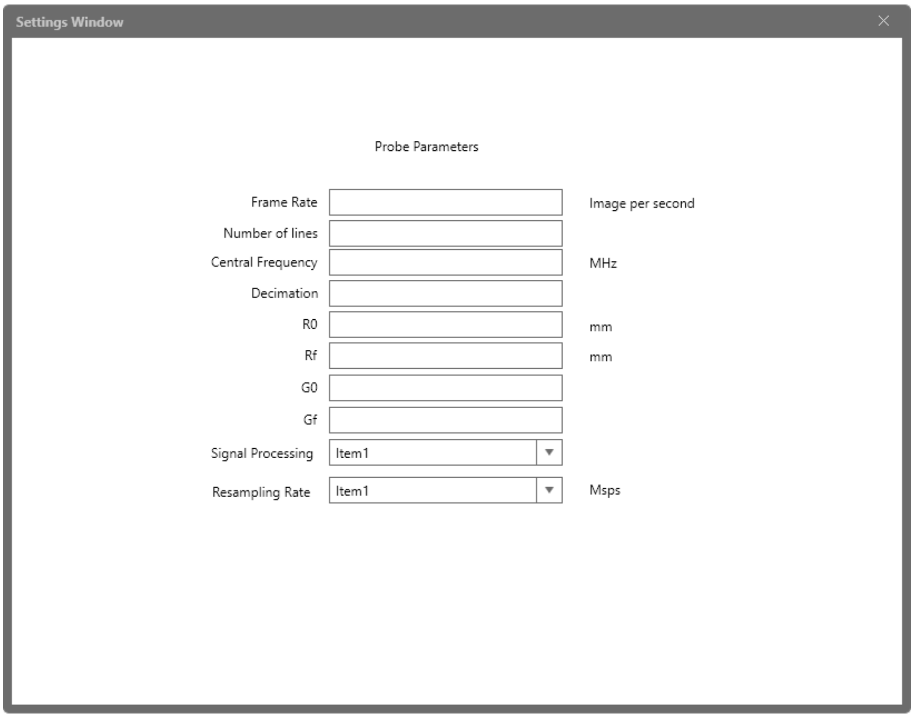

Figure 4.2: Settings window.

When connecting to probe, the software receives the settings of the probe, it display that values in this interface when opening it. On this interface the settings may be non active by default and we can add a button to modify the settings. A send settings button shall be added.

4.2.2 Settings

The different settings that can be change by the display software are described in the communication protocol document. If a setting is changed, the information is sent to the probe following communication protocol. The probe then stop imaging and modify it's settings. If number of lines, R0 or Rf is changed, it will also change the image parameters of the display software. If decimation, central frequency, R0, Rf, ADC selection, signal processing or re-sampling rate are changed, it change parameters of the signal processing module of the display device.

It is not shown on figure 4.2, but we may have also to change fmin and fmax, these parameters are used if the data send from the probe are the FFT of the signal, fmin an fmax give the frequencies of the windowing of the data in the Fourier domain.

4.3 Data streaming reception

This module is a thread that received the data from the probe continuously. This module become active since the display software is connected to the data socket. As describe in the communication protocol document, the probe send data line by line, each line is encapsulated in one TCP packet, the first value of the packet being the number of the line.

The data are encrypted on 16 bit unsigned int. The data are stored in a buffer of size Nline*Npoint? When it receives a line, it put it in the column corresponding to the line. This module (class) must have two int variable, the first one referring to the line that it is actually storing and the second one referring to the last received line that has not been processed by the signal processing module yet.

For a sweeping movement system, the line 1 and Nline will be only measured one time?

5 Signal processing

In this section we describe the signal processing that can be applied to each measurement line, all signal processing algorithm will be describe soon in a reference document. The signal processing that can be applied depend on the signal processing previously made in the probe, so it depend on the signal processing option.

5.1 RAW data sent

If the raw data are sent, the display software must make an envelope detection. This can be made using FFT algorithm or IQ demodulation.

A pass-band filter can be optionally apply to each measuring line.

A re-sampling rate can also be applied before image processing.

5.2 FFT sent

To minimize the number of data to send *via* TCP a windowing from fmin to fmax can be applied. This option are slightly discuss in settings management section.

The pass band filter can correspond to the windowing made by the probe, but a small pass-band can be applied by the display software.

A re-sampling rate can be apply before image processing.

5.3 Envelope sent

The envelope detection can be made in the probe, in this case, the data sent are already the envelope. The display have no signal processing to apply except an optional re-sampling rate.

5.4 Interface

This module have its own window where one can change the signal processing settings, this window is presented on figure 5.4. This window can be used as a king of oscilloscope interface where we display a line and the envelope of this line. In this window we can select the line that is display, it can be used as a kind of TM mode imaging.

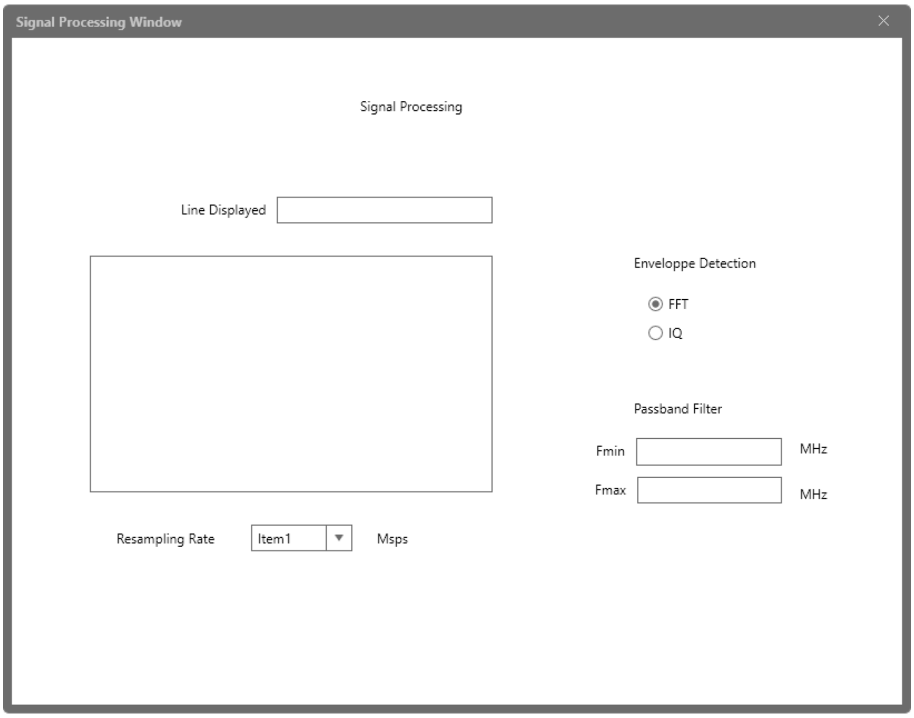

Figure 5.1: Signal processing interface.

5.5 High level processing

In this display software we also want to try some signal processing not often used in ultrasound imaging but that can improved the image quality. One possible algorithm of this king is the pseudo synthetic aperture algorithm.

6 Image processing

In this section we describe the image processing that can be applied to an image, all image processing algorithm will be describe soon in a reference document.

6.1 Gray conversion

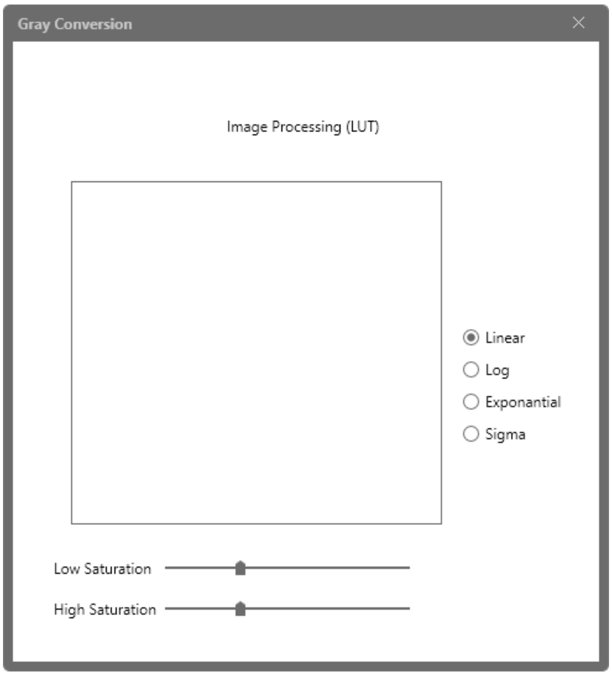

Figure 6.1: Grey conversion interface.

In ultrasound scanner one important step is to apply the right gray conversion. This conversion consist in convert the value of the envelope signal into gray level. In this display software we want to have access to this conversion table where we can change the low and high saturation and conversion curve. The interface of this module will look like what is presented on figure 6.1.

Possibly, a two stage ramp can be added.

6.2 Scan conversion

The image is acquire line by line by rotating a transducer around a given center of rotation. So we can not display the buffer as it is a rectangular image. We have to interpolate value of each point on each line that are on polar coordinate to a Cartesian coordinate (cf. RD 3), this is named scan conversion. This scan conversion can be made following two different interpolation, a “distance” interpolation or a linear interpolation. The operator may choose to not do scan conversion or to use or the other interpolation.

6.3 Pixel average

To improve the image quality by reducing the noise in the image, we can apply a pixel average on each pixel. The value of each pixel is recalculate by adding to its value, the value of each neighbor pixels. The tuning of this average processes is set in an interface such as the one presented on figure 6.2.

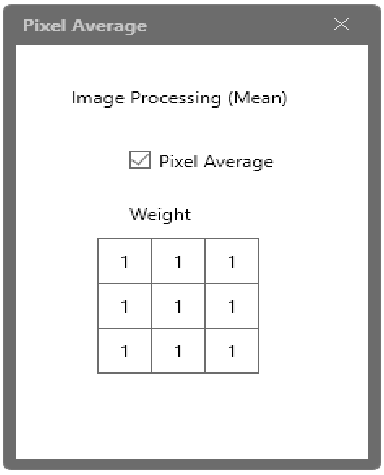

Figure 6.2: Pixel average interface.