# NanEye Viewer

Revision History:

| *Version* | *Date* | *Modifications* | *Author* |
|---|---|---|---|
| 0.0.01 | 19-10-11 | Document creation | João Santos |
| 0.0.02 | 13-12-11 | Added the new features | João Santos |
| 0.0.03 | 02-01-12 | Changed interface and added features | João Santos |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Table of Contents

# Index of Tables

# Index of Figures

# 1 Introduction

The NanEye Viewer allows to receive images from the NanEye Sensor using the EFM 001 Usb device. It allows the user to record videos, take snapshots and perform image processing: *Color Pre-Gain*; *Color Reconstruction*; *Color Adjustment, Apply LUT, Apply Gamma Correction, Adjust Brightness* and *Automatic Exposure Control*.

Moreover, the user can load recorded videos in the "*Load Awvideo File"*, and perform all the image processing described above.

This application also allows to convert the recorded videos to *Avi files* and to individual *Png* images.

# 2 Nan Eye Viewer

The main window of the Nan Eye Viewer is in Figure 1. In this document will be described all the features of this application.

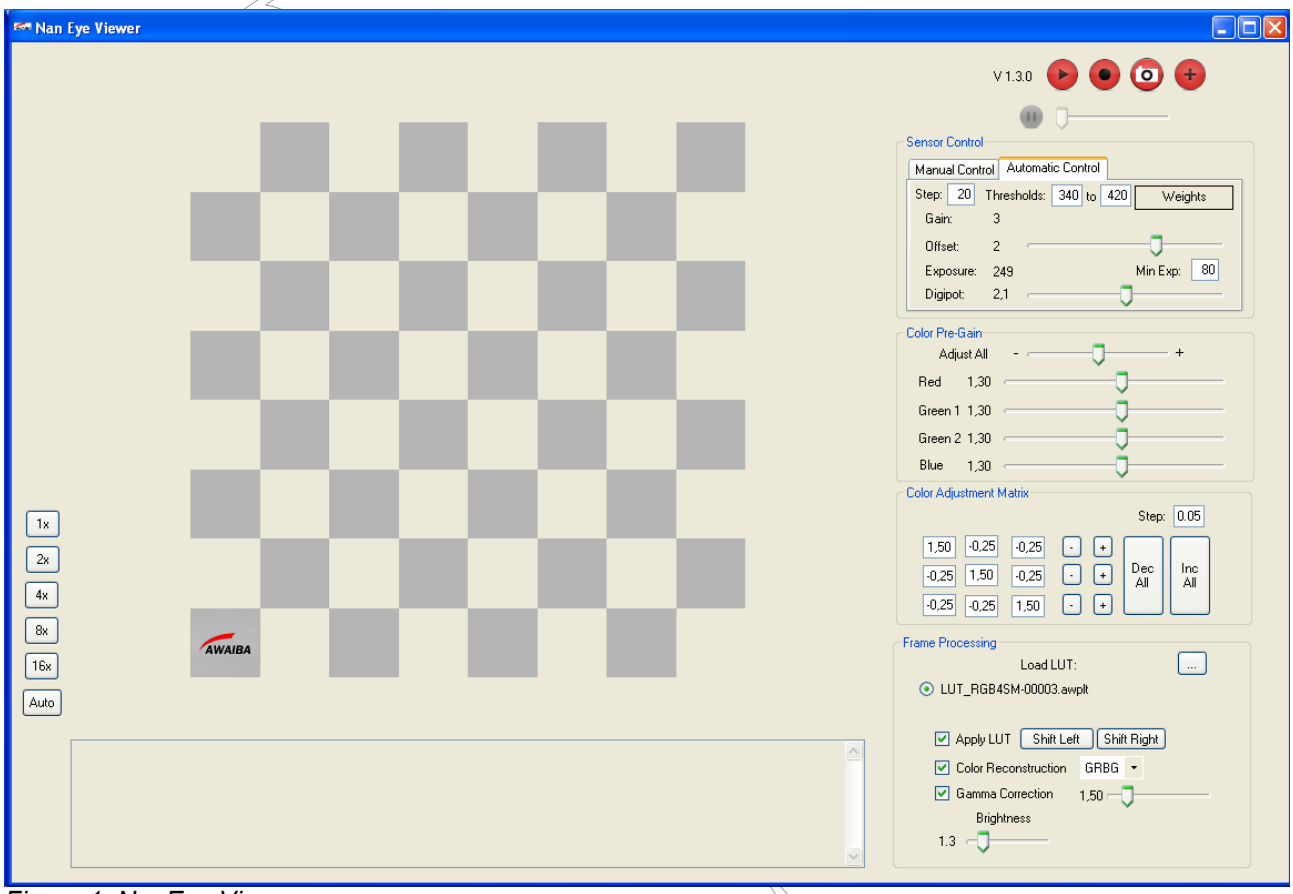


*Figure 1: NanEye Viewer*

## 2.1 Nan Eye Viewer – Load Configuration

Figure 2 presents the first interface of the application. In this window the user is able to choose which configuration to start the NanEye Viewer: *Load Last Configuration*, *Load Default Configuration*, *Load Configuration from File* and *Load Nan Eye Video Converter*.

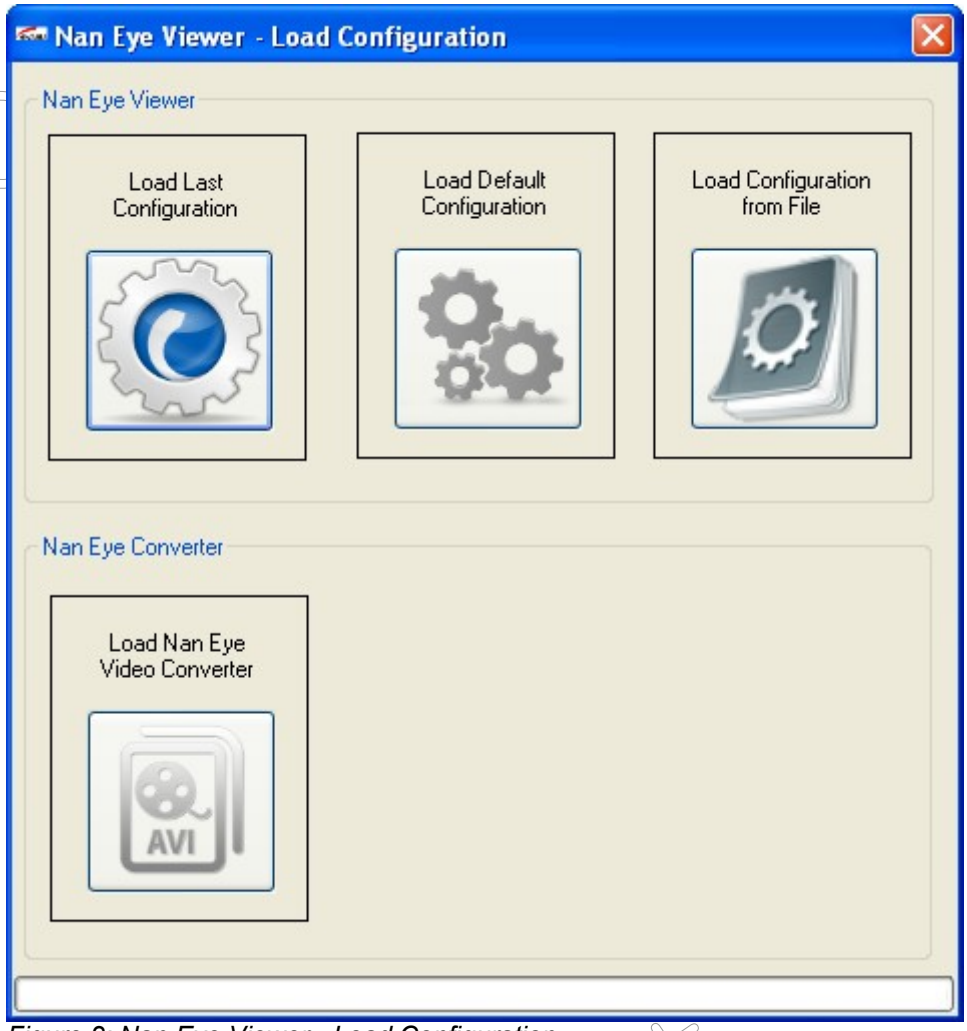


*Figure 2: Nan Eye Viewer - Load Configuration*

### 2.1.1 Load Last Configuration

In this option, the NanEye Viewer with load the last configuration that has been used in the viewer.

### 2.1.2 Load Default Configuration

The viewer loads its default configuration.

### 2.1.3 Load Configuration from File

The user is able to save personalized configurations, and in this option the user can load them.

### 2.1.4 Load Nan Eye Video Converter

This option allows to open the Nan Eye Video Converter feature, to load *awvideos* and save them to *Avi* files. This feature will be explained later on this document in section 2.2.4.

## 2.2 Interface

### 2.2.1 Image Captured, Zoom and Log Textbox

In Figure 3 it is represented where the sensor's image appear, the zoom options, and the log text box.

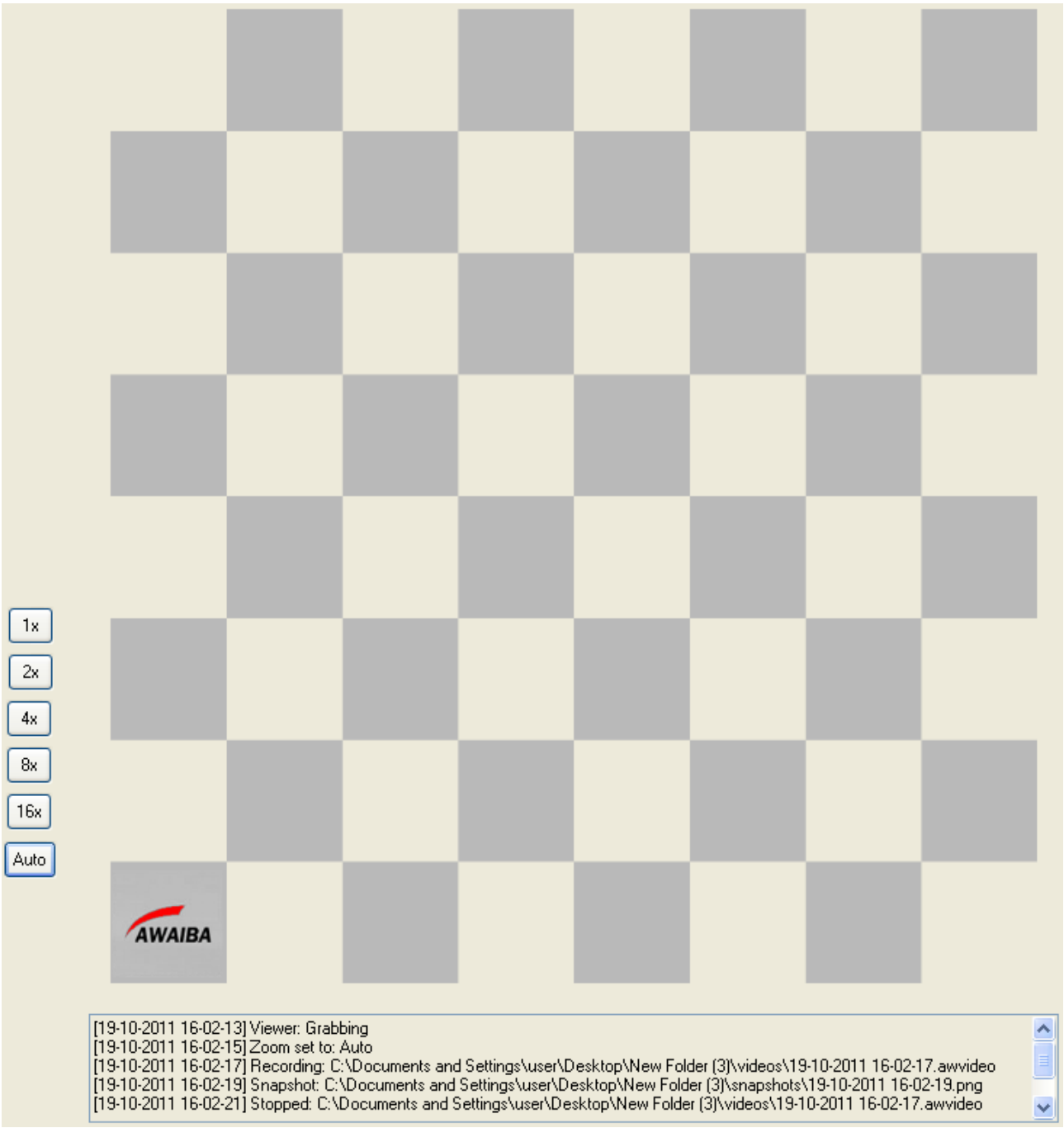


*Figure 3: NanEye Viewer - Image, Zoom and Log*

The user can choose between six different zoom options: 1x, 2x, 4x, 8x, 16x and the auto option. All the zoom options are calculated from the initial 250x250 sensor's image. The *auto zoom* allows the image to fit the viewer's size.

The log text box saves all the user's changes, and can be used as a historic. It also saves a log file in the NanEye Viewer's execution directory called: *Diagnostics Awaiba Nan Eye Viewer Log File.txt*. This file stores everything that was made using the NanEye Viewer. An example of this file is in Figure 4.

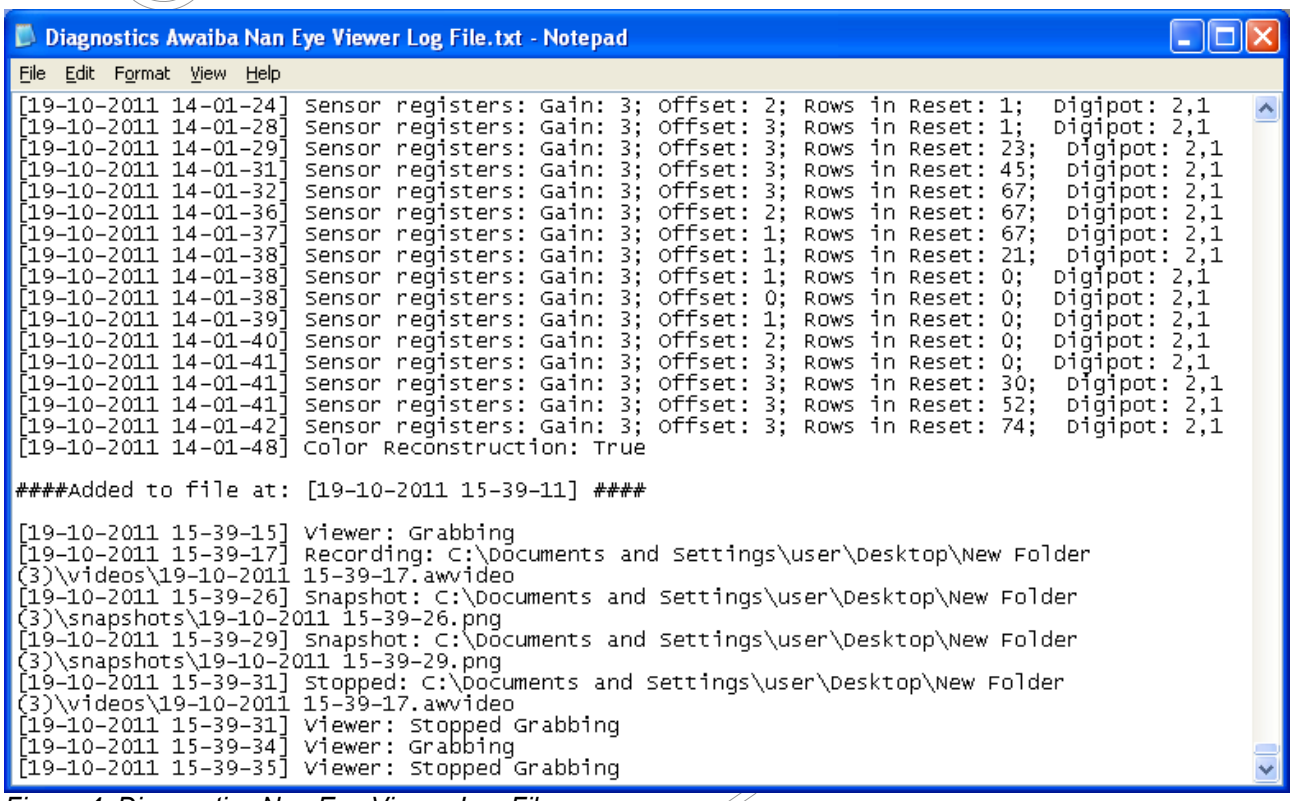


*Figure 4: Diagnostics Nan Eye Viewer Log File*

### 2.2.2 Buttons and More Options

Figure 5 shows the most important buttons of the interface:


*Figure 5: Buttons and More Options*

The *Play* buttons allows to start or stop the capture of images from the sensor.

The button aside this one, allows to record a video. This video is stored in the *Videos* folder, inside the folder chosen during the Nan Eye Viewer installation. The video's name is the date and hour of the recording.

The button that has a camera is used to take snapshots. This snapshots are in the *png* format and are stored in the *Snapshots* folder.

The *plus* button is used to show the user the other options that are available in the software, in the *More Option* window:

- The *Load Configuration* option will show the same interface as in Figure 2;
- The *Save Configuration* allows to save the current configuration in a file that is chosen by the user;
- The *Convert Video* shows the Nan Eye Video Conveter interface, that will be explained later on this document in section 2.2.4;
- The *Load Awvideo file* allows the user to load the recorded videos (with the awvideo format).

### 2.2.3 Image Processing

In Figure 6 are all the image processing features that the NanEye Viewer can perform:

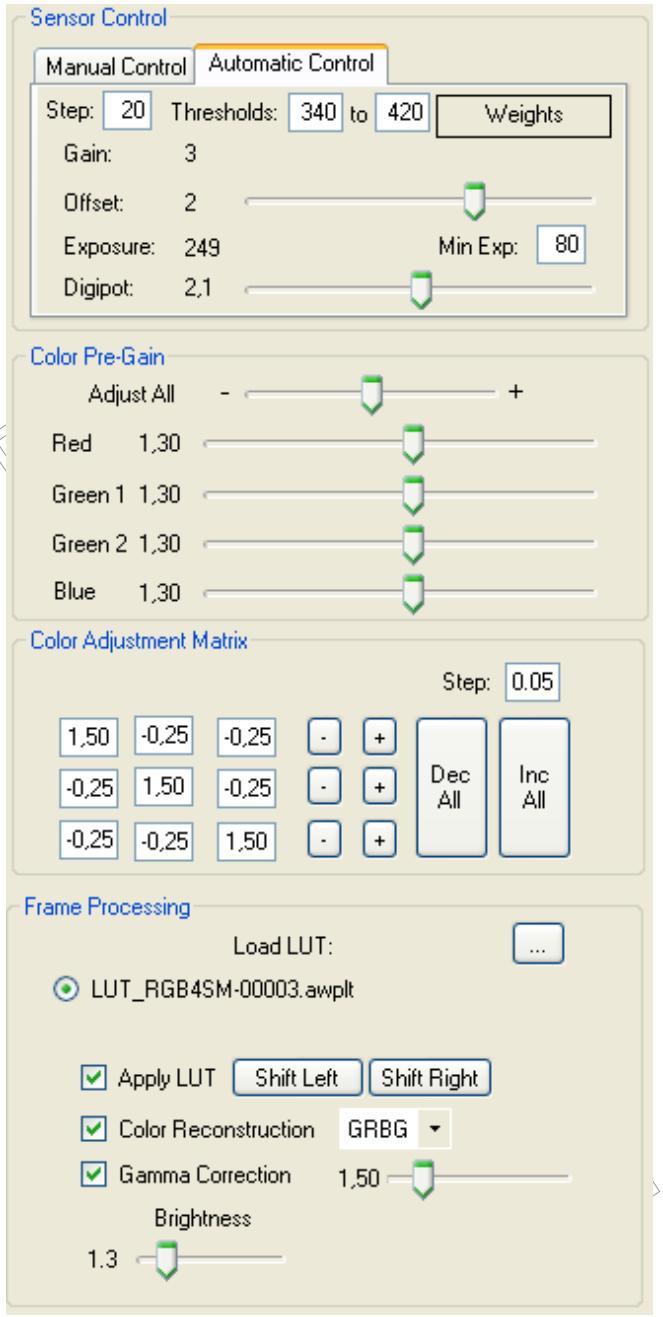


*Figure 6: Image Processing*

### 2.2.3.1 Color Pre-Gain

This color pre-gain is made directly to the raw image's pixels. The pixels are in the bayer pattern, as in Figure 7.

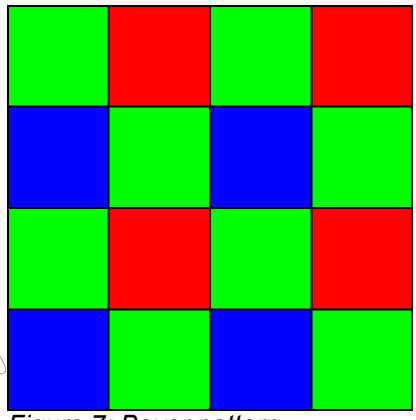

*Figure 7: Bayer pattern*

If you use the Figure 6 *color pre-gain* values, all the pixels will have its value changed by 1.3 from its pixel value. The slider *adjust all* allows to increase or decrease all the pixel's sliders, maintaining all the differences between them.

### 2.2.3.2 Color Reconstruction - Bi-linear Algorithm

After the pixel adjustment step, a color reconstruction algorithm allows to combine the pixels to form a colorful image.

This algorithm is used to pass from a gray image to a RGB image.
In Figure 7 we can see that each pixel has a different color filter, between Red, Green or Blue.

To transform a gray image to a colorful RGB image, it is needed to calculate the other pixel's components.

There is also the option to change pixel order from: *RGGB, GRBG, GBRG* and *BGGR*. This option is useful when some column or row are cut, and this allows to maintain the correct image's color.

#### Analyzing Red Filter pixel

In Figure 8 and Figure 9, the Red pixel has its component, but it is necessary to calculate the Green and Blue components. For that, it is calculated the mean of the surrounding Blue (Figure 8) and Green (Figure 9) pixel filters.

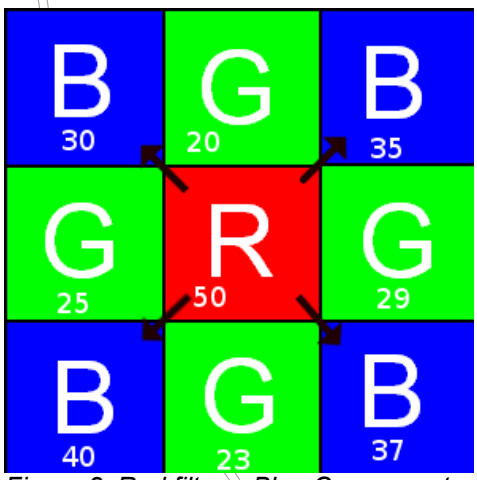


*Figure 8: Red filter – Blue Component calculation*

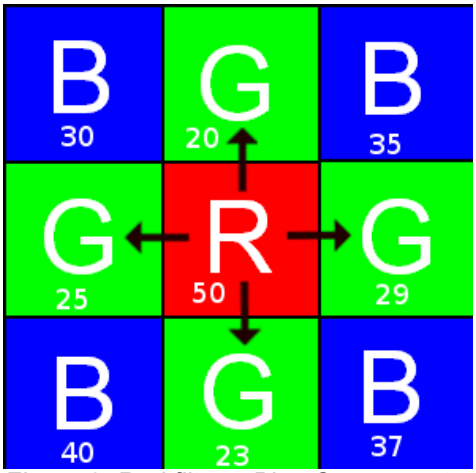


*Figure 9: Red filter – Blue Component calculation*

According to Figure 8 and Figure 9, the Blue filter and the Green filter value for the red pixel will be:

That way, that pixel passes from the gray scale value 50 to the RGB value of 50,24,35.

#### Analyzing Blue Filter pixel

When analysing a Blue filter pixel, it is calculated the Green and Red value in an analogues way (of the Red filter pixel), as we can see in Figure 10:

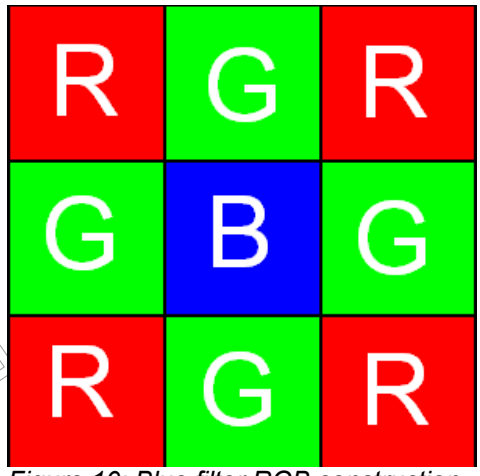


*Figure 10: Blue filter RGB construction*

**Analyzing Green Filter pixel**

As for the Green filter, in Figure 11 and 12 there is represented both the possible combinations to get the Red and Green values:

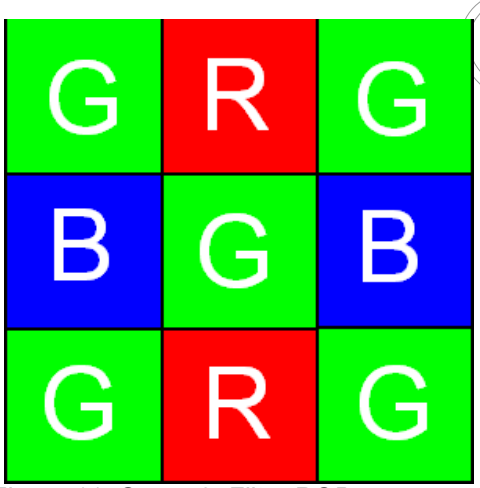


*Figure 11: Green 2 Filter RGB construction*

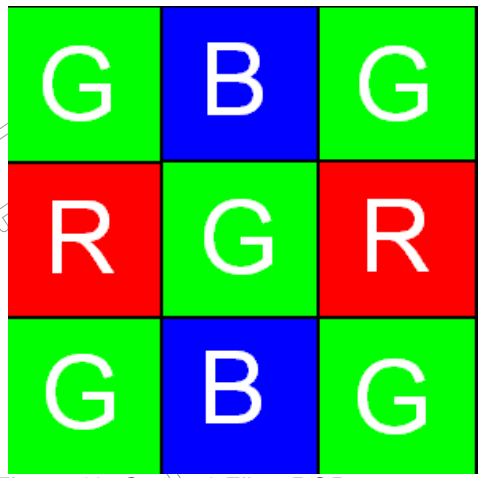


*Figure 12: Green 1 Filter RGB construction*

In Figure 11, the Red value will be the mean of the two Red filter pixels that are immediately above and below of the Green filter pixel that is being analysed. The Blue value, is the mean of the right and left Red filter pixel.

In the case of Figure 12, the Blue value will be the mean of the two Blue filter pixels that are immediately above and below of the Green filter pixel that is being analysed. The Red value, is the mean of the right and left Red filter pixel.

### 2.2.3.1 Color Correction Matrix

After the execution of the DeBayer algorithm, the images are in the RGB format, with 8 bits representing each color (red, green or blue).

The objective of this matrix is to saturate the images, changing the pixel's RGB values. For that, it multiplies the RGB pixel's values with the matrix that is in Figure 13. To allow the brightness to stay the same, the sum of each line should always be 1.

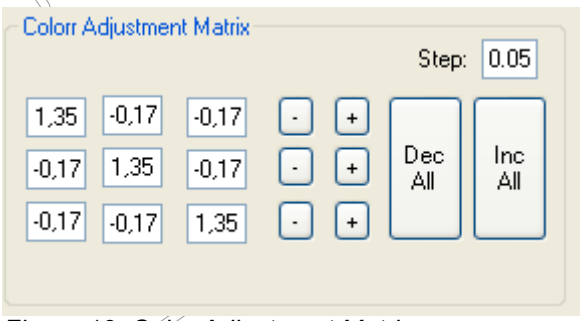


*Figure 13: Color Adjustment Matrix*

**Algorithm**

For example a pixel that before had the value 100, know is represented by three different values (R, G and B), for example 105,90,70. This values are achieved using the DeBayer Algorithm.
The color correction matrix is used to adjust that values, allowing to saturate the colors.

For example, the pixel 90,90 has the value 105,90,70 (for the red, green and blue component, respectively). By applying the matrix, that is in Figure 1, the new values will be:

Red: $105*1.35+90*(-0.175)+70*(-0.175)=114$

Green: $105*(-0.175)+90*1.35+70*(-0.175)=91$

Blue: $105*(-0.175)+90*(-0.175)+70*1.35=64$

In this example, the new values of the pixel 90,90 are: 114,91,64.

#### 2.2.3.2 Load LookUp Table

This option loads a *awplt* file that allows to have a linear pixel response.

#### 2.2.3.3 Adjust Brightness

This option allows to change the pixel value by the constant chosen by the user. If a pixel has the 100 value (and using the value in Figure 6), that pixel will have its value changed to 130.

#### 2.2.3.4 Gamma Correction

When choosing this option, there will be performed a *gamma correction* through all pixels with a gamma parameter that can be defined from 1 to 4.

#### 2.2.3.5 Sensor Control

This option allow to change the sensor's registers:

- *Gain*: value from 0 to 3.
- *Offset*::value from 0 to 3.
- *Exposure*: value from 250 to 0.
- *Digipot*.:value from 1.8 to 2.4.

**Automatic Exposure Control**

In this option, in Figure 14, the sensor's exposure and gain value are controlled automatically, to change its values according to the environment light.

The user defines the low and the high *Threshold* values. This values correspond to the mean of some pixels in the image. If this thresholds values are defined too close, it could lead to an unstable image.

The *step* is how much the exposure value is changed, if the image is not inside the Thresholds values. The higher this value, the faster the exposure value will change to match the environment conditions. If this value is too high it can lead to unstable images because the exposure value can change too much, not allowing the image's pixels to fit in the Thresholds values.

The *Min Exp* is the minimum value from witch the gain is changed instead of changing the exposure value.

The *Weights* parameters allows to give more, or less importance to an image region. The user can choose from three difference options. The number of green squares correspond to the importance of the region.

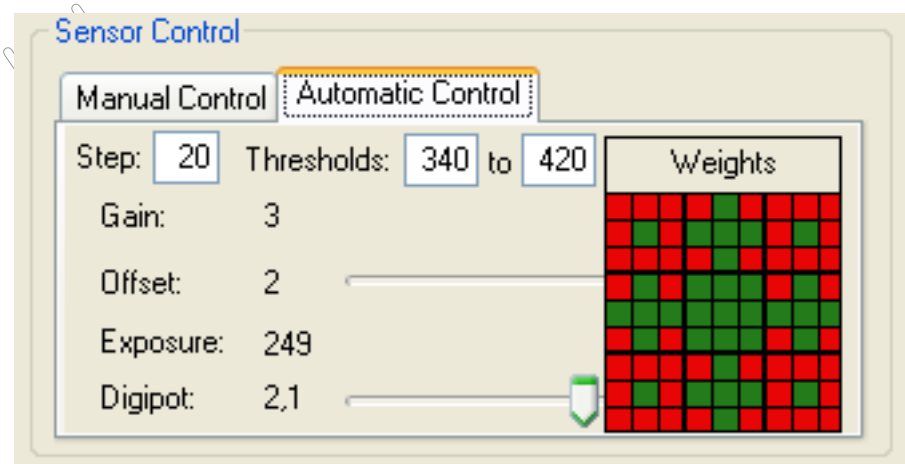


*Figure 14: Sensor - automatic exposure control*

### 2.2.4 Nan Eye Video Converter

In Figure 15 is the *NanEye Video Converter* interface, where the user chooses the different files and image processing settings.

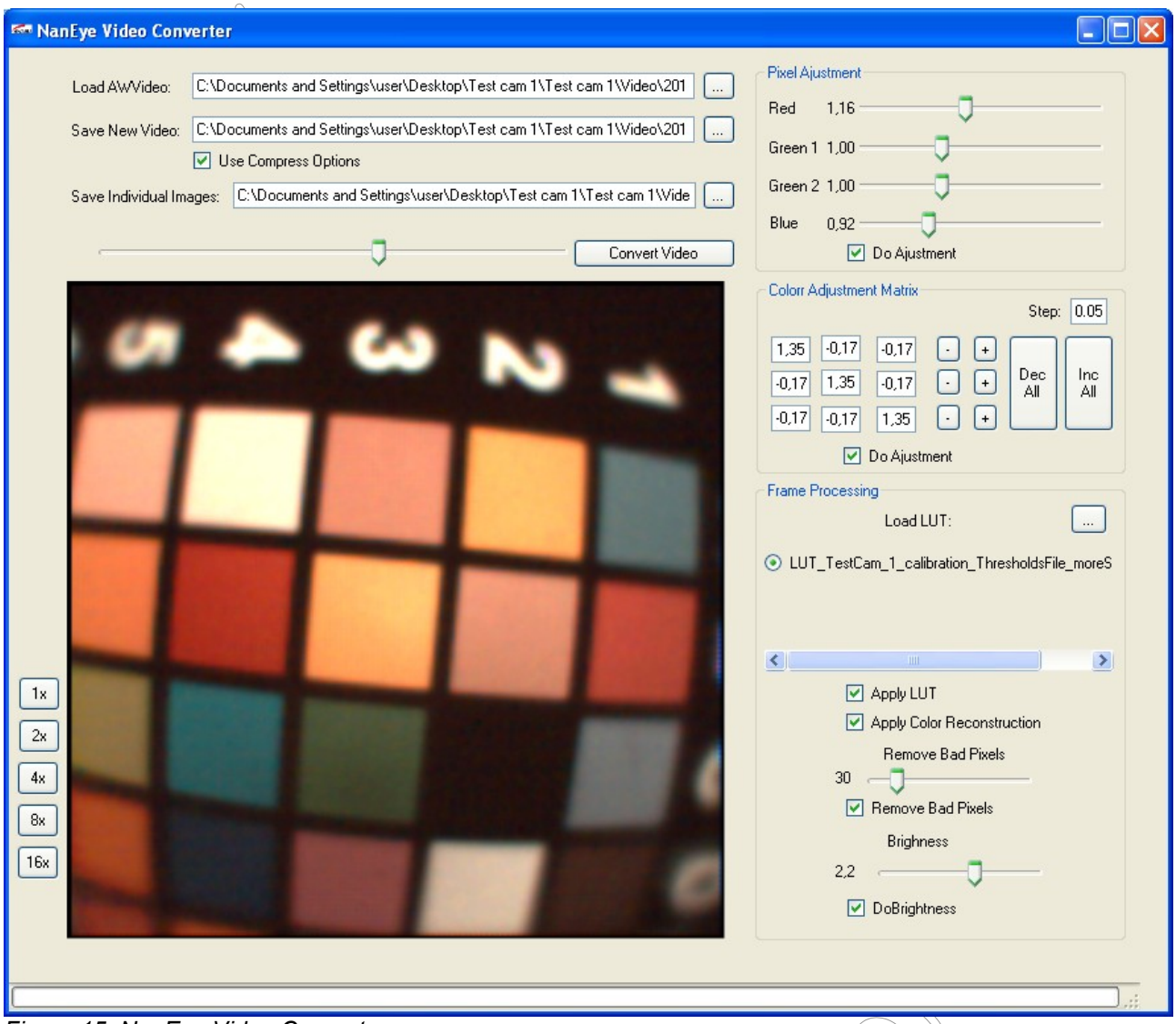


*Figure 15: NanEye Video Converter*

The user is able to scroll within the video, using the slider above the image. Moreover, it is also possible to zoom in the image.

#### 2.2.4.1 Files and Folders

Figure 16 presents all the files that are used to load and save the videos.

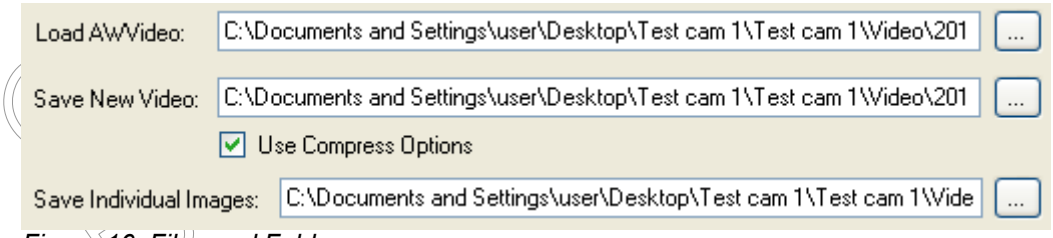


*Figure 16: Files and Folders*

The *Load AWVideo* is used to load a *AwVideo*. This video is created using the *Awaiba Nan Eye Viewer's* record option. The *Save New Video* allows the user to choose the *avi* file to convert the *AwVideo*. If *use compress options* is selected, it will appear an interface (in Figure 17) that allows to choose between the installed codecs.

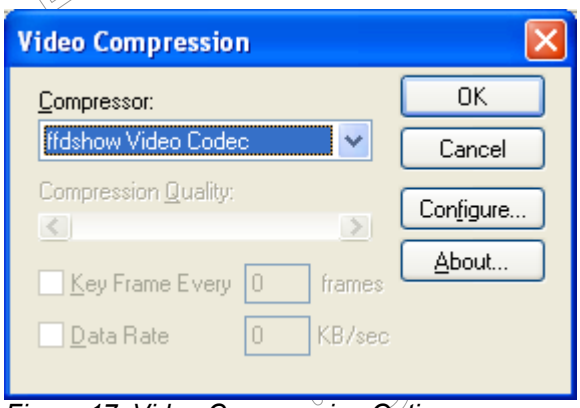


*Figure 17: Video Compression Options*

If the compressor is not configured, it will throw an error. To configure the compressor is necessary to use the *configure* button, appearing the Figure 18, that allows to choose the video encoding configuration.

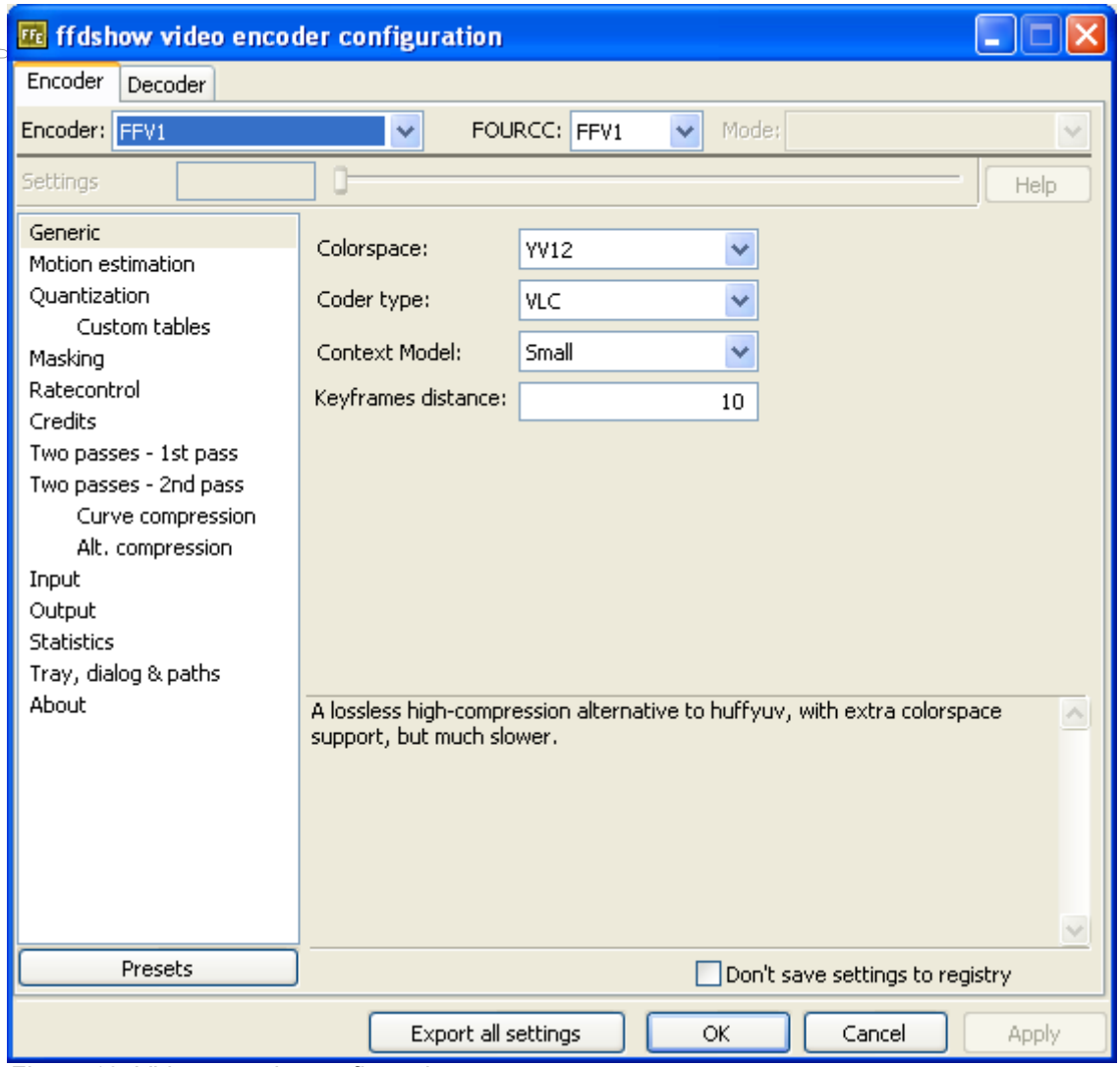


*Figure 18: Video encoder configuration*

The *Save Individual Images* option allows to choose where to save all the video frames, in the *Png* format.

It is possible to save both a new *Avi* video and *individual images* at the same time as it is possible to just save either one of them.

### 2.2.4.2 Log File

When an *Avi file* is created, it is also created a log file that stores how the conversion was made, storing all the files and folders locations, as well as the different image processing choices. An

example of that file is in Figure 19. This file is saved in the same folder of the *avi* video and in the individual image's folder.


*Figure 19: Log File*