WiDyVISION 1.3

REFERENCE GUIDE

<table>
<thead>
<tr>
<th>Author</th>
<th>Date</th>
<th>Object</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sophie AMBERT</td>
<td>16/06/14</td>
<td>Creation</td>
<td>1.0</td>
</tr>
<tr>
<td>Sophie AMBERT</td>
<td>23/07/14</td>
<td>Modifications</td>
<td>1.1</td>
</tr>
<tr>
<td>Sophie AMBERT</td>
<td>22/12/14</td>
<td>Modifications</td>
<td>1.2</td>
</tr>
<tr>
<td>Sophie AMBERT</td>
<td>29/01/15</td>
<td>Modifications</td>
<td>1.3</td>
</tr>
</tbody>
</table>

WidyVISION 1.3

REFERENCE GUIDE

<table>
<thead>
<tr>
<th>Author</th>
<th>Date</th>
<th>Object</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sophie AMBERT</td>
<td>16/06/14</td>
<td>Creation</td>
<td>1.0</td>
</tr>
<tr>
<td>Sophie AMBERT</td>
<td>23/07/14</td>
<td>Modifications</td>
<td>1.1</td>
</tr>
<tr>
<td>Sophie AMBERT</td>
<td>22/12/14</td>
<td>Modifications</td>
<td>1.2</td>
</tr>
<tr>
<td>Sophie AMBERT</td>
<td>29/01/15</td>
<td>Modifications</td>
<td>1.3</td>
</tr>
</tbody>
</table>
# Table of Contents

1. INTRODUCTION ........................................................................................................... 3
2. DRIVER INSTALLATION ................................................................................................. 4
3. DISCOVER THE WIDYVISION ..................................................................................... 5
   3.1) Connect the camera ................................................................................................. 5
   3.2) Main GUI ................................................................................................................ 6
       3.2.1) Toolbar Menu .................................................................................................. 7
       3.2.2) Camera Controls ............................................................................................ 9
   3.3) AGC CONTROLS ..................................................................................................... 11
       3.3.1) AGC Controls ............................................................................................... 12
       3.3.2) Gain and Offset Controls ............................................................................. 12
   3.4) NON UNIFORMITY CORRECTION ......................................................................... 13
   3.5) RECORDING ........................................................................................................... 17
   3.6) IMAGE PROCESSING ............................................................................................... 18
       3.6.1) Colour Maps ................................................................................................. 18
       3.6.2) Filters ............................................................................................................. 20
       3.6.3) Gamma Correction ....................................................................................... 20
   3.7) ROI, HISTOGRAM AND CAMERA INFORMATION ........................................... 23
       3.7.1) ROI ................................................................................................................ 23
       3.7.2) Histogram ...................................................................................................... Erreur ! Signet non défini.
       3.7.3) Camera Information ...................................................................................... Erreur ! Signet non défini.
4. ANNEXE: READ PTW C++ ......................................................................................... 27
5. MATLAB SCRIPT FOR PTW ...................................................................................... 29
1. INTRODUCTION

WiDyVISION is the new updated software provided with our InGaAS cameras (320U, 640U, and 640US). It replaces the previous software, WiDyView.

It includes all necessary tools to discover our cameras and offers a user-friendly interface. It is only working on Windows system.

This software is not subject to any demand and the code source is not provided.
2. DRIVER INSTALLATION

Before launching the application and get the acquisition output, you need to install the necessary driver.

You will find a Driver folder on the package of the software. The folder contains drivers for multiple Windows platform. Please follow these steps to install correct driver:

1. Connect the camera board to the PC USB port. A Dialog Box inviting the user to install a new device driver should then pop up. In the event this screen does not show up you can have access to the new device in the system device manager in the system options.
2. Then Right Click on the unknown device to install the driver and follow the installation procedure as follow:
   a. Select Update Driver Software ...
   b. Select “Browse my computer for driver software”
   5. Browse the correct platform driver from “…\Driver”.

Now verify on your device manager that you have a Cypress USB device on the list

![Figure 1: Driver Installation](image-url)
3. DISCOVER THE WIDYVISION

3.1) Connect the camera

When you execute the application a first window will open to select your camera:

![Camera Selection Dialog](image.png)

Figure 2: Camera Selection Dialog

3 SWIR cameras are available:

- WiDy SWIR 320U
- WiDy SWIR 640U
- WiDy SWIR 640U-S

The list on the top of the dialog box shows all connected cameras, click on one row to select the camera then configure your camera parameters using the list box below.

Select the right camera if you want to upload the good timing configurations for the sensor.
3.2) Main GUI

Figure 3: Main Interface
The main interface is divided on 2 windows: a Camera Stream window when you will have the living output of the camera and the control interface with the commands and post-processing functions. Press the green icon start to begin the acquisition.

![Camera Stream Window](image)

**Figure 4: Camera Stream Window**

### 3.2.1) Toolbar Menu

![Toolbar Menu](image)

**Figure 5: Toolbar Menu**
<table>
<thead>
<tr>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="START" /></td>
<td>Start the acquisition</td>
</tr>
<tr>
<td><img src="image" alt="STOP" /></td>
<td>Freeze the acquisition</td>
</tr>
<tr>
<td><img src="image" alt="CAMERA CONNECT" /></td>
<td>Select a camera</td>
</tr>
<tr>
<td><img src="image" alt="SNAPSHOT" /></td>
<td>Save an image to JPEG, PNG or BMP</td>
</tr>
<tr>
<td><img src="image" alt="BAD PIXELS TOOL" /></td>
<td>Open the bad pixels calculation dialog (see part 3.)</td>
</tr>
<tr>
<td><img src="image" alt="SELECT ROI" /></td>
<td>Allow you to display statistical information on a ROI (tab 5)</td>
</tr>
</tbody>
</table>
3.2.2) Camera Controls

![Camera Controls Tab](image)

**Figure 6: Camera Controls Tab**

<table>
<thead>
<tr>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
</table>
| Mode               | ➢ For the SWIR 320U and 640U the mode is Rolling  
                      ➢ For the SWIR 640U-S the mode is Global Shutter |
| Exposure Time      | ➢ The combo box is locked on rolling mode  
                      ➢ On Global shutter, you can choose your exposure time |
| Trigger Mode       | 2 trigger modes are available (see below for a description) |
| Pixel Clock        | Set the Pixel Clock (the control is disabled and fix at 25MHz)) |
| Frame Rate controls| FPS controls  
                      |
| Send command       | When you change the FPS make sure that the commands have been send to the camera by pressing this button |
TRIGGER SETTINGS:

2 trigger modes are available:

- Trigger from ext. to camera: the camera is waiting for an external signal. The trigger signal must be injected through the SMC connector. The trigger signal must be in the voltage range of 0-3.3V. The camera will be synchronized with every pulse rising edge. The time between 2 pulses rising edge must be > 15ms. Each pulse generated during the 20ms delay will be ignored until the end of the readout sequence.

- Trigger from camera to ext.: the camera is sending a trigger output signal during the live acquisition. A pulse (0-3.3V) is generated every frame and can be used for triggering external devices.
3.3) AGC CONTROLS

![AGC Controls Tab](image)

**Figure 7: AGC Controls Tab**

<table>
<thead>
<tr>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGC Controls</td>
<td>Set on automatic or manual controls of the image low/high limits. The slider controls are available on manual mode</td>
</tr>
<tr>
<td>Gain and Offset Controls</td>
<td>Set the gain and offset levels to change the displaying instead on using the high and low limits sliders</td>
</tr>
</tbody>
</table>
3.3.1) AGC Controls

Be sure to know what you are doing when changing these settings.

3.3.2) Gain and Offset Controls

The settings of the high limit and low limit can be very sensitive. In fact, if you are moving to far the limits sliders you arrive quickly to saturation.
In this case, it is better to use the gain and offset sliders which can add more precision to set the level of the displaying.

3.4. NON UNIFORMITY CORRECTION

![NUC Tab]

**NUC PROCEDURE:**

In this tab, functions are given to compute the NUC table and apply the correction on the output of the camera to remove dark current. By default, a calibration NUC file named **NUCFactory.yml** on the NUC folder is provided which contains the calculated data. Do not remove this file! By default if this file exits the check box activate NUC will be set to OK.

The panel gives you the procedure to compute the NUC; you can do it also by yourself by following this procedure. This will save 2 matrixes on the memory. The third step will compute and allow you to save the NUC data on the NUC folder directory (on the root of the executable file). Then you can click on the checkbox button to activate the NUC.

For the WiDy SWIR U-S, the correction is a bit little different, due to its double memory, we need to apply 2 different corrections. But the processing method is the same. Be careful, that every time you
change the exposure time you have to do calculate a new set of data for the NUC using the procedure. **By default, we have the NUCFactory.yml has been computed for 20ms.**

**DEFAULT SETTINGS FOR NUC:**

By default, the NUC is activated because the software goes to get the values of the matrix on the NUCFactory file; this is why you should not remove it.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate NUC</td>
<td>Activate the correction after loading or computing the correction data. By default, the NUC is activated by loading data from the NUCFactory.yml file</td>
</tr>
<tr>
<td>Activate Bad Pixels Correction</td>
<td>Load the bad pixel map calibration file named BPM.yml</td>
</tr>
<tr>
<td>Low Point</td>
<td>Calculate the dark matrix. To do it correctly cover the lens or pointed it to a dark uniform source and push the button to acquire 50 dark frames</td>
</tr>
<tr>
<td>High Point</td>
<td>Calculate the bright matrix. To do it correctly, expose to an illuminated uniform source and push the button to acquire 50 frames</td>
</tr>
<tr>
<td>Compute NUC</td>
<td>After calculated the 2 previous matrix, this will compute the gain map and the offset</td>
</tr>
<tr>
<td>Load BPM</td>
<td>Load a bad pixels map</td>
</tr>
<tr>
<td>Load NUC Data</td>
<td>Open a dialog to load an existing calibration file</td>
</tr>
</tbody>
</table>
For the second checkbox which concerns the Bad Pixels correction, by default the BPM.yml file contains the Bad Pixels Map. If you want to calculate these bad pixels by yourself, a tool is available on the toolbar menu as indicated on the part.

To calculate the bad pixels, you will need to load 2 videos (a video on dark and another one on an illuminated scene) on PTW format (see later for more precision about this format) that you can by using the functions on the recording tab. This will save a BPM.yml file after calculations. Thanks to the button “Load BPM” on the main interface of the WiDyVISION, you can load your own Bad Pixels Map.
Procedure to detect the bad pixels:

- Cover the lens of the camera and record a PTW video to store several dark frames. Load the video.
- Expose the camera lens to a uniform and powerful light source and record a PTW video to store several illuminated frames.
- The Bad Pixel Maps will appear on the widget below, you can adjust the tolerance of the responsively by moving the slider to considerer more or less enough pixels as bad.
- Press compute BPM to save the BPM.yml which contains your bad pixels map.
3.5. RECORDING

![Recording Tab](image)

**Figure 10: Recording Tab**

<table>
<thead>
<tr>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record AVI</td>
<td>Record video on AVI format</td>
</tr>
<tr>
<td>Record PTW</td>
<td>Record video on PTW format (RAW data)</td>
</tr>
<tr>
<td>Convert PTW to PNG (available on June 2015, upgrade to a video viewer)</td>
<td>Convert your PTW video on PNG images (“Frame_Number”)</td>
</tr>
</tbody>
</table>

An OpenCV code and Matlab script is also provided to read PTW videos (see annexes).
3.6. IMAGE PROCESSING

Figure 11: Image Processing Controls

3.6.1) Colour Maps

<table>
<thead>
<tr>
<th>Colour Maps</th>
<th>JET</th>
<th>RAINBOW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1" alt="Jet Image" /></td>
<td><img src="image2" alt="Rainbow Image" /></td>
</tr>
</tbody>
</table>
3.6.2) Filters

<table>
<thead>
<tr>
<th>Filters</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invert</td>
<td><img src="image" alt="Invert" /></td>
</tr>
<tr>
<td>Canny</td>
<td><img src="image" alt="Canny" /></td>
</tr>
<tr>
<td>Laplace</td>
<td><img src="image" alt="Laplace" /></td>
</tr>
</tbody>
</table>

3.6.3) Gamma Correction

You can adjust the contrast and the gamma of the scene using the gamma tools provided on the last tab of the interface. The gamma curve is described as an “S-Curve” as below:
You can modify 2 parameters: the gamma factor and the center of the curve. If the value is less than the center the LUT curve red is applied and if it is greater than the center value, the LUT curve in blue is applied.
3.6.4) Image Enhancement (Release on June 2015 only!!!)

You can activate or not the contrast enhancement with this option.

![Image before and after enhancement]

**IMAGE ENHANCEMENT FILTER**

<table>
<thead>
<tr>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color Maps</td>
<td>Apply one of the available color maps</td>
</tr>
<tr>
<td>Filters</td>
<td>Apply one of the 3 available filters</td>
</tr>
<tr>
<td>Zoom</td>
<td>Apply a Zoom x2</td>
</tr>
<tr>
<td>Gamma S Correction</td>
<td>Apply the Gamma S Correction</td>
</tr>
</tbody>
</table>
3.7) ROI, HISTOGRAM AND ANALYSIS FEATURES

3.7.1) ROI

To access the ROI functionality click on the button to activate the mouse event, by checking the button you can draw a rectangle on the “Camera Stream” window and select a ROI, this will open and display a histogram for the selected region.

![ROI Histogram](image)

Figure 12: ROI Histogram

3.7.2) HISTOGRAM

You can choose to check and uncheck the displaying, consider disabling this function if you are not using it, this will reduce the consummation of the calculation for the software.
3.7.3) Analysis features

On the analysis tab you can proceed to some measurement for a certain number of frames:

You can choose between median, mean and standard deviation variables. First enter a number of frames and then push the button plot to obtain the statistic curve.
4. DEVELOPMENT TOOLS

The **SWIRSDK** is the example to build an application using Visual Studio 2013, OpenCV 2.4.9 and Cypress FX3 SDK to implement a user-interface with our SWIR cameras. It contains all the necessary functions to integrate our cameras on your own system.
5. ANNEXE: READ PTW C++

```cpp
#include <iostream>
#include <fstream>
#include "windows.h"
#include "Mmsystem.h"
#include "cv.h"
#include "highgui.h"

using namespace std;
using namespace cv;

// Simple Program for reading images in a .ptw files and
// and displaying it using OpenCV library.

// Initialization of the different buffers to be used.
unsigned short *bufferImage;
char *buffer;
ifstream DataBase;

int LgthFileMainHeader = 3476;
int LgthImHeader = 1016;
int NombrPixellImage=0;
int NombrImage=0;
unsigned short NombrColImage=0;
unsigned short NombrRowImage=0;
Mat CVImBuffer;
Mat CVIm;
double MaxVal;
double MinVal;

void main()
{

    //Open the file "Data.ptw" for reading
    DataBase.open("C:\Documents and Settings\cbouvier\Bureau\testvid.ptw", ios::in | ios::binary);
    DataBase.seekg(0,ios::beg);
    //Recover the number of pixels in images:
    DataBase.seekg(23,ios::beg);
    DataBase.read((char *)&NombrPixellImage,4);
    //Recover the total number of images:
    DataBase.seekg(27,ios::beg);
    DataBase.read((char *)&NombrImage,4);
    //Recover width of images:
    DataBase.seekg(377,ios::beg);
    DataBase.read((char *)&NombrColImage,2);
    //Recover height of images:
    DataBase.seekg(379,ios::beg);
    DataBase.read((char *)&NombrRowImage,2);

    // We Initiate the image buffer size. Images are stored in the database
```
// are stored under 16 bits per pixel format.
buffer = (char *)malloc( NombrPixelImage*2*sizeof(char) );

// Matrix allocation for displaying images with OpenCV.
CVImBuffer.create(NombrRowImage,NombrColImage,CV_8U);
CVIm.create(NombrRowImage,NombrColImage,CV_32F);

DataBase.seekg(3476,ios::beg);
if(DataBase.is_open())
{
    for (int i=0; i<NombrImage; i++)
    {
        // The file pointer is moved to the beginning of the image pixels data block.
        DataBase.seekg(1016,ios::cur);

        // We read the data block corresponding to the image pixels in row order.
        DataBase.read(buffer,884736);

        // A pointer is initiated to fill a OpenCV array and we fill the array with the image pixels.
        float* Mj = CVIm.ptr<float>(0);
        for (int jji=0; jji<(int)NombrRowImage * (int)NombrColImage; jji++)
        {
            bufferImage = (unsigned short *)&buffer[2*jji];
            Mj[jji] = saturate_cast<float>(*bufferImage);
        }

        // The image is rescaled to be displayed on a conventional display.
        minMaxLoc(CVIm, &MinVal, &MaxVal);
        CVIm = (CVIm - MinVal)/(MaxVal-MinVal);
        CVIm = 255.0 * CVIm;
        CVIm.convertTo(CVImBuffer,CV_8U);
        //imwrite("Imtest.bmp",CVIm);
        // We draw the image on the display.
        imshow("Video Stream",CVImBuffer);
        cout << "Image =" << i << endl;
    waitKey();
    }
}
else
{
    cout << "Unable to open file";
}
}
6. MATLAB SCRIPT FOR PTW

```matlab
clear all
close all
clc

[filename  pathname] = uigetfile('*.ptw*', 'Choose a picture ptw file');
fid = fopen([pathname  filename],'r');

LgthFileMainHeader = 3476;
LgthImHeader = 1016;

%Seek the number pixel in an image:
fseek(fid, 23, 'bof');
NbPixelImage = fread(fid,1,'uint32');
%Seek the total number of image:
fseek(fid, 27, 'bof');
Nbimage = fread(fid,1,'uint32');
%Seek the height of the images
fseek(fid, 377, 'bof');
NbColImage = fread(fid,1,'uint16');
%Seek the width of the images
fseek(fid, 379, 'bof');
NbRowImage = fread(fid,1,'uint16');

%Set the file point on the header of the first image
fseek(fid, 3476, 'bof');
A = zeros(NbRowImage,NbColImage,Nbimage,'uint16');

h = waitbar(0,[filename  ' database importation : ' num2str(0)  '/'
num2str(Nbimage)  ]);

for  i=1:Nbimage

    waitbar(i/Nbimage,h,[filename  ' database importation : ' num2str(i)  '/'
num2str(Nbimage)  ]);  
    fread(fid,1016);%Jump the image header
    B = fread( fid , [NbColImage,NbRowImage] , 'uint16' );
    [M,N] = size(B);
    A(:,;,:i) = uint16( B( 1:NbColImage , 1:NbRowImage )' );

end
close(h)
fclose('all');
```
NIT, New Imaging Technologies and Native WDR, are trademarks of New Imaging Technologies.