Photonic Science

Scientific Detector Systems

PSL-Viewer Software

User guide

CONTENTS

APPENDIX

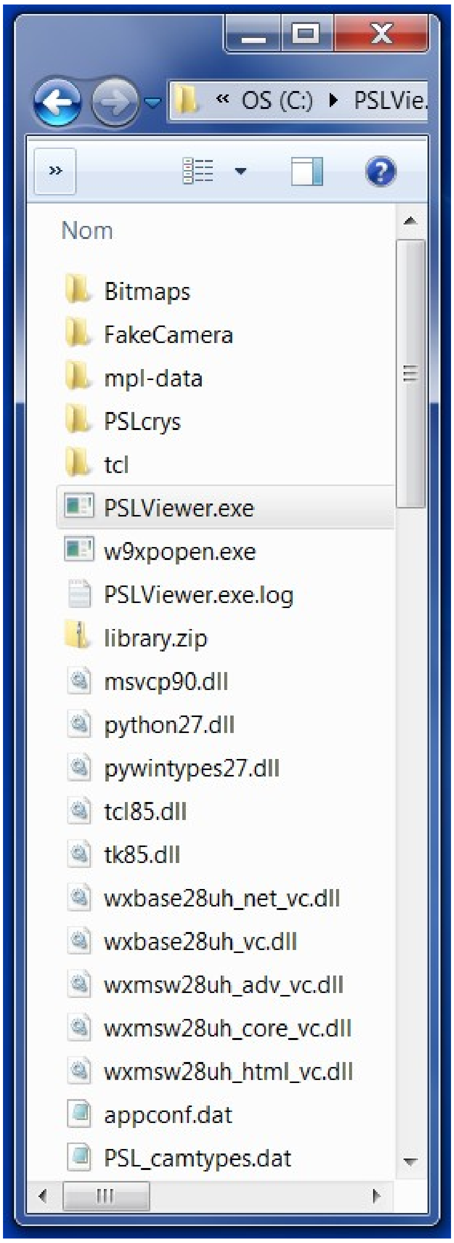
I. Installation and camera detection

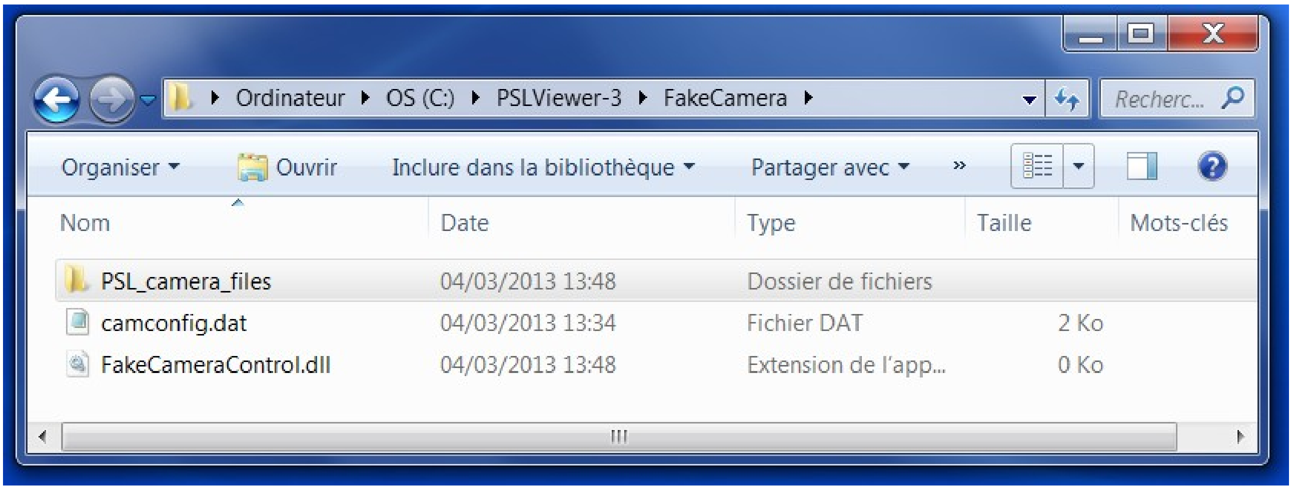
a) Software installation

Copy the software main folder *"PSLViewer"* on your master drive (e.g. *C:\PSLViewer*)
Start the software with a double click on *"PSLViewer.exe"* (e.g. *C:\PSLViewer\PSLViewer.exe*)

b) Camera folder and configuration files

Each camera you want to control with PSLViewer must have its own camera folder.
The camera folder must be in the main software folder (e.g. *C:\PSLViewer\FakeCamera*).
The camera folder name must use as prefix the camera type name (e.g. if you have a VHR camera, the camera folder name must be *"VHR"* or *"VHR-1"* or *"VHR_whatever"*).
The file *"PSL_camtypes.dat"* describes the camera type names allowed.





The camera folder must contains a unique camera DLL and the *"PSL_camera_files"* folder.
The DLL depends on the frame grabber type you are using and may be updated according to your frame grabber version.

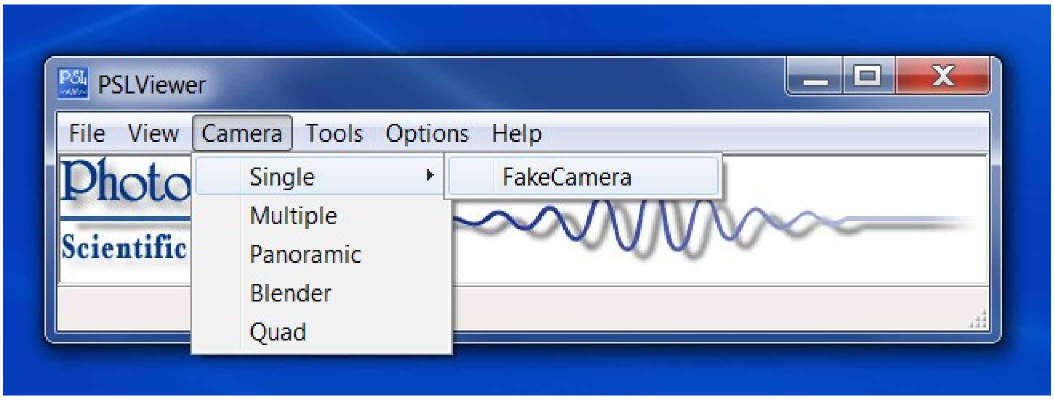
The *"PSL_camera_files"* folder, contains all the camera setup and correction files.
These files are specifically designed for your camera and generated during the calibration process performed in our laboratory. They are provided on a CD delivered with the camera.
If you lose or corrupt these files, please contact us and we will send you the original copy.

The camera folder also contains the *"camconfig.dat"* file which stores the camera configuration of the last software session.

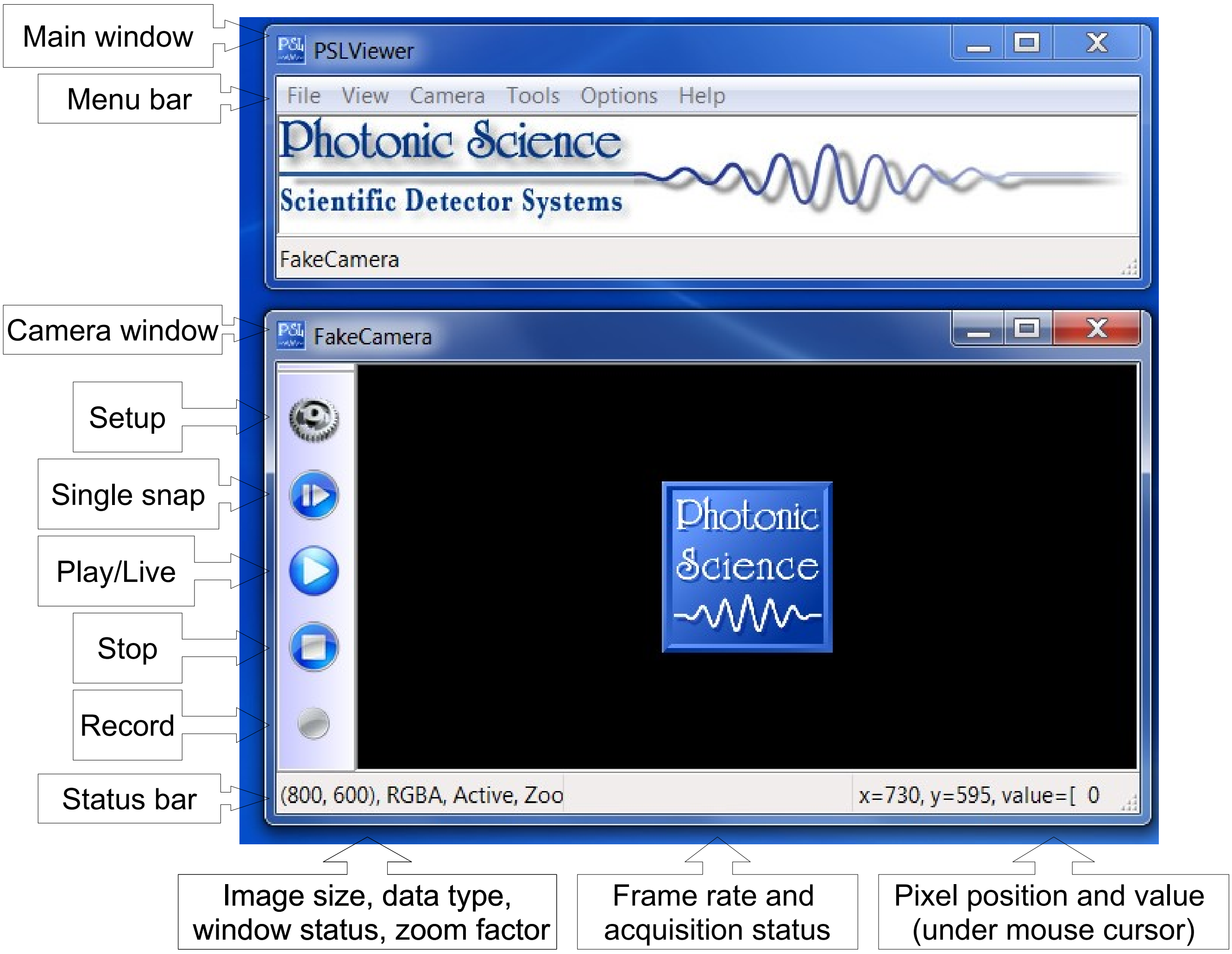
c) Open a camera viewer

Start the software with a double click on *"PSLViewer.exe"* (e.g. *C:\PSLViewer\PSLViewer.exe*)
The main software window appears on the screen.
This main window allow you to manage multiple cameras and image files.

To select and open a camera, go in the main window menu bar and select: "*Camera* => *Single* => *FakeCamera*" (instead of *FakeCamera* select the name corresponding to your camera)



After selecting your camera, the software starts the camera initialization process.
If the camera answer properly you must see the camera window appearing on the screen.
The window title corresponds to your camera folder name.
All the image acquisitions and camera configuration actions will be performed from this window.



The acquisition control buttons are on the left side of the camera window:

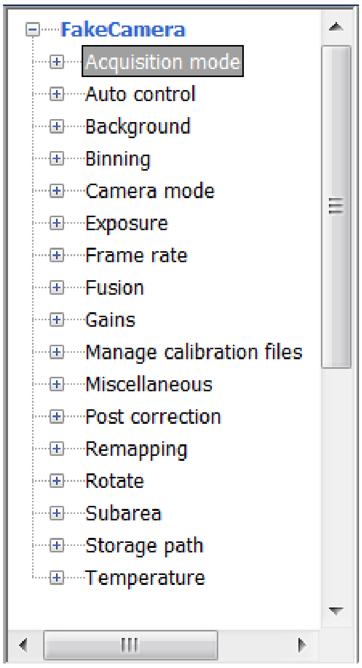
The *"**Setup**"* button allow you to open/close the camera options tree. From the setup panel you can modify the camera configuration and the acquisition parameters.

The *"**Snap**"* button starts a single image acquisition.
The *"**Play**"* button starts a live acquisition.
The *"**Stop**"* button stops a running live acquisition or a single snap acquisition.
The *"**Record**"* button enable/disable the automatic image recording on disk (turns red when ON).
(see the setup section to see how to change the saving path)

The *"status bar"* (left field) displays the current image size, data type, window status and zoom factor.
The *"status bar"* (middle field) displays the current acquisition status and the frame rate.
The *"status bar"* (right field) displays the current pixel position and value (under mouse cursor).

II. Image acquisition

a) Setup control and options tree



The camera option tree allow you to modify the camera configuration and the acquisition parameters. The number of options available in the tree may change regarding the camera model you are running. Refer to the camera manual for specific options which are not describe in this manual.

Acquisition mode:

"*FrameNumber*" is the number of image to acquire. If "*FrameNumber*" = 10 and if you press the "*Snap*" button then a sequence of 10 images will be acquired. If you press "*Play*" instead then it will start a infinite loop of a 10 images sequence acquisitions.

You can choose between 3 different acquisition modes:

- "*Standard*": usual mode, images are displayed one after the other.
- "*Average*": display the Average of "*FrameNumber*" images.
- "*Summation*": display the Sum of "*FrameNumber*" images (switch to float data type)
- "*Accumulation*": display the Sum of "*FrameNumber*" images minimizing the noise accumulation (switch to float data type)

The "*ShowSteps*" tick box allow you display the intermediate steps of the modes. For example, if you average 10 frames together and if you turn off the "*ShowSteps*" tick box then 10 images will be acquired but not shown, only the final averaged image is displayed. Else if you turn on the "*ShowSteps*" tick box, you will see image1 then (image1+image2)/2 then (image1+image2+image3)/3 … until (image1+ … + image10)/10.

Auto control:
This section show you all the options which can be automatically adapted by the camera such as the exposure or gains.
-*AutoBinning*: the camera automatically adapts the binning to the light level.
-*AutoLevel*: the camera automatically adapts the exposure and gain to the light level.
-*BestFit*: center the camera display range to the PEEK value.
-*BinningFilter*: apply binning to the image and rescale it to the unbinned size. If enabled it overrides the AutoBinning option.

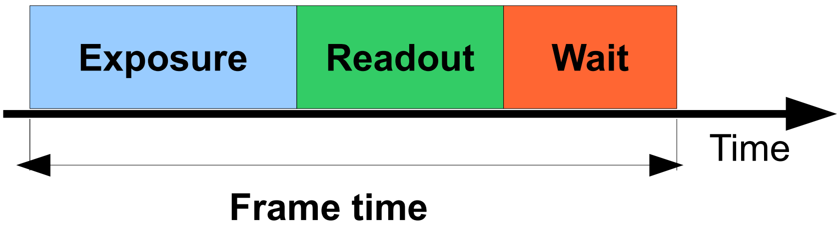
Background:
Enable\Disable the automatic background subtraction.
If turned on, a background image is subtracted after each image acquisition,
Before doing the subtraction, the "Pedestal" value is added to the acquired image to avoid data clipping of possible negative values while subtracting.
You can save a new "Background" file. Select the number of frames you want to average together and press the "*Save*" button. DO NOT perform this action if the camera is already running (single snap or live)!
If the "Software method" option is ticked, the background subtraction is done by the software (using the file "*background.tif*" from "*Camera_Folder*"). Useful while running Multi-cameras.
If the "Software method" option is NOT ticked, the specific camera background subtraction method is used (using the file "*background.flf*" from "*Camera_Folder\\PSL_camera_files*").

Binning:
-*Hardware*: On-chip binning is the process of summing the signal from neighbouring pixels prior to the readout of this signal from the CCD. It has the effect of increasing the signal amplitude (by a factor equal to the number of pixels being binned), whilst making little change to the random noise amplitude. The signal-to-noise ratio of a signal therefore improves by the binning factor. Binning reduces resolution by the binning factor in both X and Y directions. Binning in the Y-direction also reduces the time taken to read out the CCD, very approximately by the Y-binning factor.
For example, when compared to a 1392 x 1040 unbinned image, an image captured with 3 (X) x 2 (Y) binning would have a resolution of 464 x 520, a signal amplitude six times higher, and would read out in about half the time.
-*Software*: The Software Binning is very similar to the On-chip Binning except that the sum of the pixel signals is not limited by the CCD saturation level. On the other hand, because the signal from the binned pixels is not combined before the read out, the noise level is greater than the same binning but in hardware. The “SoftBin” algorithm is slower than the On-chip binning.

Camera mode:
-*ClockSpeedMode*: select the camera internal frequency.
-*ColorMode*: select the camera output data format (Mono-8bit, Mono-16bit,RGB-24bit …).
-*ContrastMode*: select between the different built-in auto-contrast algorithm.
-*FanMode*: Enable/Disable the camera fan.
-*GainMode*: select the level of the camera sensitivity.
-*TriggerMode*: select the way to trigger the camera to perform an image acquisition.
The camera includes an external trigger feature that allows the camera’s exposure to be synchronised with an external event. Two forms of this feature are available: a hardware trigger, and a software trigger. The hardware trigger input is an optional feature, whereas the software trigger facility is always present.
See your camera documentation for more details on available trigger modes.
-*VirtualShutterMode*: select the level of the virtual shutter correction.

Exposure:
The integration or exposure period of the CCD is user-selectable via software over a very large range. The integration period is defined by a 14 bit number (range 1 to 16384) together with a unit (microseconds, milliseconds, or seconds). Very fine control of integration period is therefore available to the user.
Note that the driver written by Photonic Science always uses the finest unit available for any particular selected integration period, e.g. an integration period of 17 seconds is transmitted to the camera as period = 17, unit = seconds, whereas an integration period of 16 seconds is transmitted to the camera as period = 16000, unit = milliseconds.

Frame rate restriction:
This option allow you to slow down the acquisition frame rate in order to give more time to the system to process third-party threads. This can be useful if your actual frame rate is not stable enough. The value is a time in milliseconds which will correspond to the minimum time between two consecutive image acquisitions. This option is transparent if you set a time shorter than the minimum time between two image acquisitions. For example, if your camera maximum native frame rate is 40fps, so the time between two images is 1/40 = 0,025s = 25ms. Then if you set the "*FrameRate*" below 25ms it is transparent.
And if you set the "*FrameRate*" to 30ms (5ms above 25ms) then the software will wait 5ms after the image acquisition. During this 5ms sleep, the system have time to process concurrent threads and it may helps to stabilize the camera to computer data transfer timing.



Fusion:
This option extend the dynamic range of the camera by 16 times. An *"Average"* value of between 1 and 16 can be selected. A setting of 16 offers the least noise at higher grey levels but at the expense of increased overall acquisition time. A setting of 1 minimises acquisition time but has higher noise at the higher grey levels.
If available on your camera, you can also enable the *"Low noise"* option to reduce the noise.

Gains:
-*IntensifierGain*: select the intensification level of the camera intensifier (100 or 4095 steps).

-*VideoGain*: Video Gain is user selectable via the software driver in 100 steps. The transfer function is approximately logarithmic. Gain amplifies the signal amplitude but does not change the fundamental signal-to-noise ratio.

Manage calibration files:
This option allow you to create new correction files like the "FlatField" file. Select the number of frames you want to average together and press the "*Save*" button.
Read carefully any message from the software about the camera configuration requirements.
DO NOT perform this action if the camera is already running (single snap or live)!

Miscellaneous:
In this section you will find all camera functionalities which are very specific and rarely used.
See your camera documentation for more details.

Post correction:

-*BrightCorner*
At long exposure times the dark current in this camera is spatially non-uniform, with a higher value along the top edge of the image. Bright edge subtraction removes this dark current together with its non-uniformity, thereby giving an image with practically zero dark current.

-*BrightPixel*
Enable this correction to reduce the number of bright pixels (individual high dark current pixels). The CCD sensor in this camera is cooled to reduce dark current. This cooling results in the dark current in the majority of pixels to be negligible, allowing very long on-chip exposures. There are however a small proportion of isolated individual pixels with higher dark current, which can appear as bright pixels at longer exposure times. "Bright pixel" automatically subtracts these bright pixels from the image as part of the acquisition process.

-*DarkCurrent*
Removes the camera dark current noise as part of the acquisition process.

-*FlatField*
Wide aperture lenses and fibre-optics can introduce shading across an image, resulting in reduced intensity at the edges of the image compared to the centre and in small scale patterning from fibre optic structure. Flat field correction restores the intensity in the shaded areas and removes any patterning to compensate for such effects, and produces an image of very high spatial uniformity. When enabled, this function divides the acquired image by a stored flat field image, on a pixel-by-pixel basis, as part of the acquisition process. This correction supports binning, sub-area, and gain operations. For best results, always enable "*Offset*" correction when using flat field correction.
Note: flat field correction can be enabled only if the file "*flat.flf*" has been previously installed and loaded correctly during camera initialisation. This file has to be in the "*PSL_camera_files*" folder corresponding to your camera (such file is provided with the camera).

-*Offset*:
Enable this correction to automatically carry out a "dark subtraction" as part of the acquisition process. To do this the driver loads a stored dark image of the appropriate binning setting and subtracts it from the acquired image before returning the image. This process sets the *"Pedestal"* (the average value of the image in darkness) to 100 ADU over the whole image in all binning modes. This correction supports binning and sub-area operations.
Note: offset correction can be enabled only if the files *"darkbin*x*.flf"* have been previously installed and loaded correctly during camera initialisation. Those files have to be in the *"PSL_camera_files"* folder corresponding to your camera (a range of such files are provided with the camera). If you would like to operate the camera with auto offset correction in a binning mode for which a stored dark file is currently unavailable, please contact Photonic Science.

Remapping:
Enable this option to remove the effect of fibre-optic distortion by automatically resampling the raw image. Enable *"Smooth"* to use sub-pixel interpolation of the original image, which eliminates any aliasing effects introduced by the resampling. Enable *"Clip"* to exclude edge irregularities caused by the remapping process, and to present a cleaner remapped image, particularly when sub-area operation is employed.
Note: image remapping can be enabled only if the file *"distortion.map"* has been previously installed and loaded correctly during camera initialisation. This file has to be in the *"PSL_camera_files"* folder corresponding to your camera.

Rotate:
Apply one or a combination of the 3 elementary rotations.
-"*Fliplr*": Left/Right flip.
-"*Flipud*": Up/Down flip.
-"*Rotate*": 90 degrees anti-clockwise rotation.

Subarea:
Only the image data for this rectangular sub-area being transferred to the PC. This feature can be used in conjunction with binning, to provide the user with considerable flexibility in image location, size, frame rate, and resolution.
The time taken to read out the CCD will be reduced, in proportion to the number of whole TV lines that are above and below the sub-area selected. For example, if the sub-area (200,200) - (700, 700) were selected, then readout would occur in about half the time of a full 1392 x 1040 image, and the image returned to the PC would have resolution 500 x 500. If in addition binning of 2 x 5 were selected, the sub-area selected would be unchanged but the image resolution would become 250 x 100.
('*L*', '*T*', '*R*', '*B*') = (Left, Top, Right, Bottom) labels refer to the rectangle box which defines the sub-area.

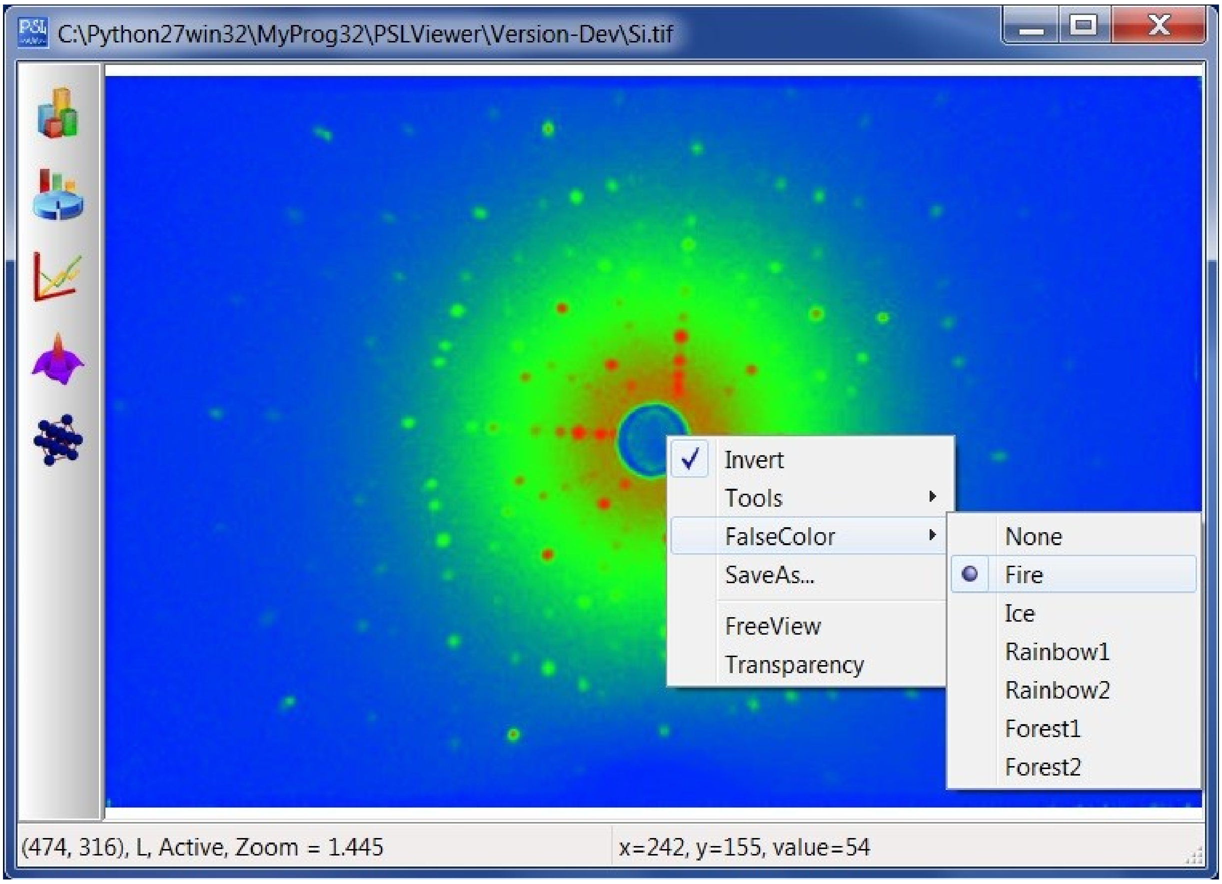
Storage path:
When the "Record" button is turned on, all new image will be automatically saved at the end of the acquisition process. In this section you can define the saving path.
The path is built like this: "*Directory\NameNumber.Format*"
By default the *Directory* parameter is equal to the path to the CameraFolder.
The *Name* parameter is equal to '*frame_*' and default *Format is* '*tif*'.
The *Number* starts with the value define in the "*Offset*" field and is automatically incremented.

Press the "*Counts*" '*reset*' button to reset the image counter.
!!! WARNING: files with the same file name are overwritten !!!

You can write comments in the "*Tag*" field in order to add these lines to the DESCRIPTION TAG (TIFF file only).

Temperature:
In this section you can read the current camera temperature by pressing the button.
The "*Target*" field let you define a target temperature for the camera cooling.

b) Image manipulation



Resize the image/camera window: grab the right bottom corner to resize the window.
The image display size is automatically adjusted in order to fit the window.
The current magnification is displayed in the "StatusBar" (left field) ("Zoom=1.108")

Drag the image: press and hold left click on the image and move the mouse to drag.

Zoom in/out: use the mouse wheel to zoom in a specific image area.

Context menu: press right click on the image to open the contextual menu.
From this menu you can access the *Tools*, *Invert* or select a *FalseColor* lookup table.
Use *SaveAs* to store the image as {TIFF, JPEG, PNG, BMP, GIF, ICO, PNM, PCX, FLF} file.
You can also set the image window in *FreeView* mode and define the window *Transparency*.

c) Tools and measurements

The tool bar is on the left side of the image/camera window.
Press one of the buttons to enable/disable the associated tool.

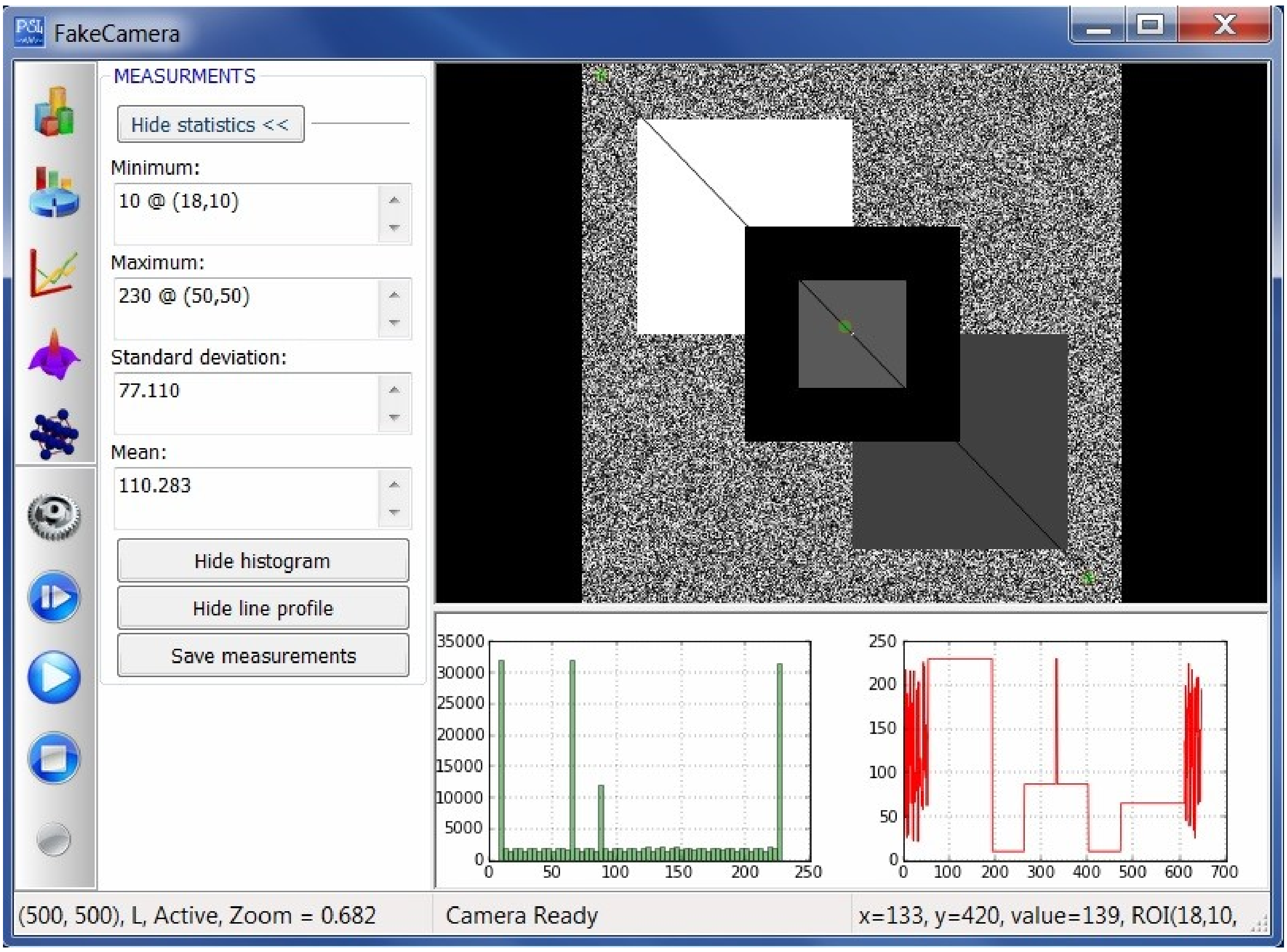
Display: display range, image filters.

Measurements: statistics, histogram, line-profile.

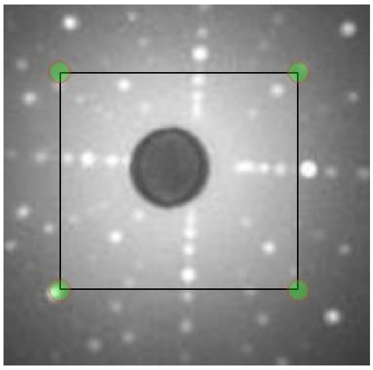
Mathematics: operations, parameters, curve plot.

Shape detection: search peaks, find objects, match pattern.

Laue orientation: crystal, instrument, orientation, refinement, find HKL.

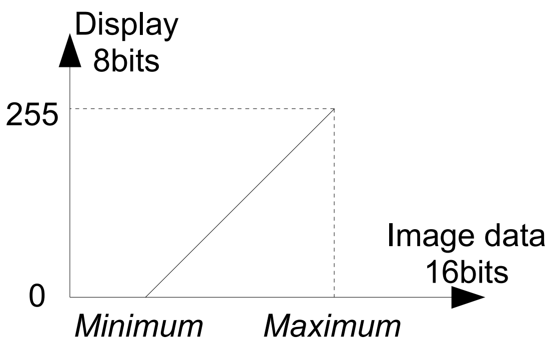


Draw a measurement selection:
Once you have selected a tool, press and hold right click on the image and move the mouse to draw a line or a rectangle (depending on the active tool).
The line or the rectangle selection can be re-adjusted using the green dots.
You can translate the line selection with the middle green dot.
You can translate the rectangle selection by holding left click on the selection. By default, if no area selection is drawn, the measurement is performed on all the image.



Display range:
When the camera image data depth is superior to 8bits, the data has to be scaled to an 8bits display range to be displayed on screen.

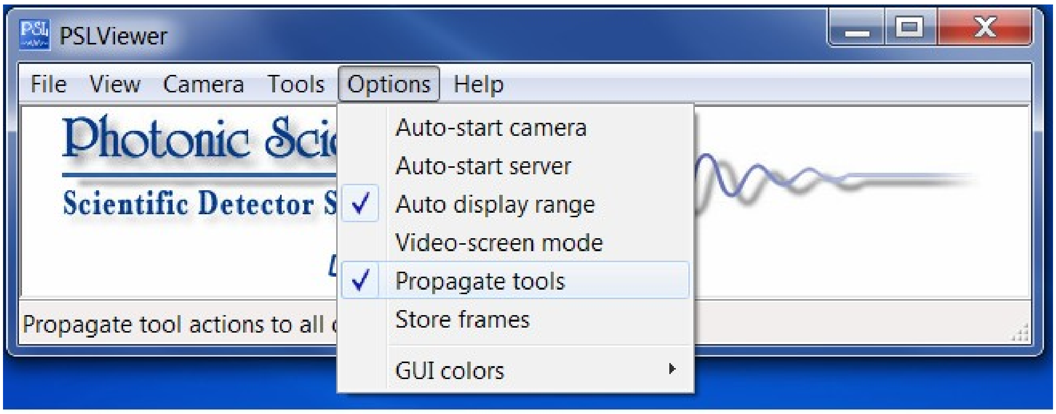
The display range tool uses a linear scaling function. The *Minimum* and *Maximum* value can be defined by the user or automatically adjusted by the software. All pixels below the minimum are set to 0 (black) and all pixels above the maximum are set to 255 (white).



Speed-up the min/max search by setting the *Sub-sampling* factor to a greater value (0=disable).

Avoid bright and dark pixels by setting the *Gaussian filtering* factor to a greater value (0=disable).

Measurements on multiple images:

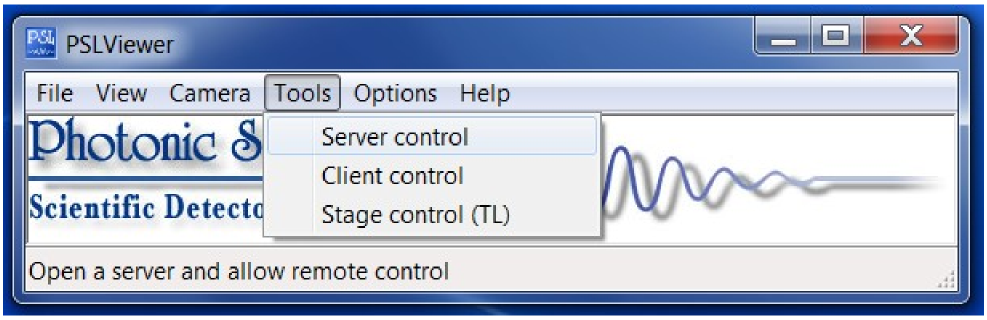


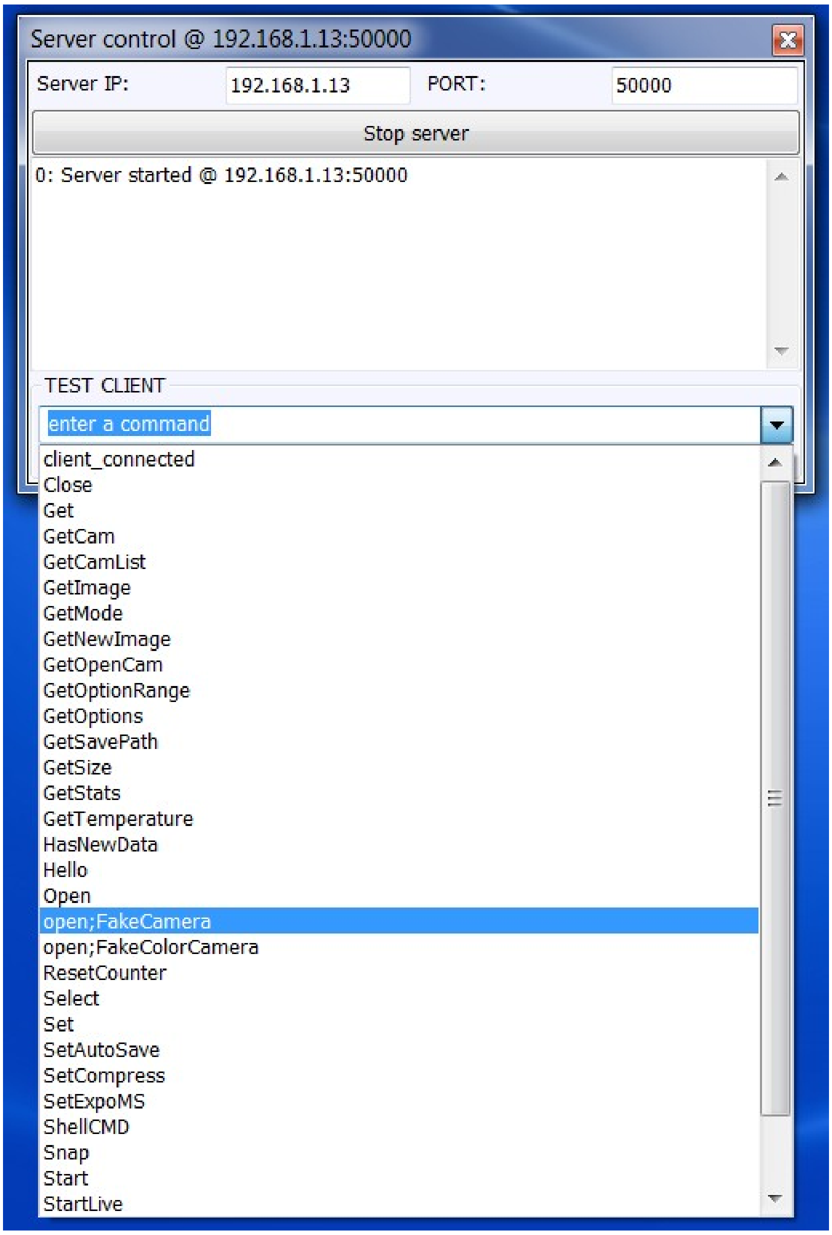
In the case you have multiple images/cameras opened in the software, you can ask to draw the same selection on all images when doing it on one of the images.
To do so, go in the software menu bar and select => "*Options*" => "*Propagate tools*".

III. Remote control

a) Server control tool

The server control tool allow you to start a socket server which listen for incoming client requests from the network. The server reads and understands a list of keywords allowing a full control of the camera and data management.





Start the server:
Define the IP number and the PORT you want to use to run the server.
The default IP number correspond to the IP of the first network card found in the windows network list (type in a DOS shell/console => "*ipconfig*" or "*ipconfig /all*").
The default PORT value is 50000.

Start the server using the defined IP and PORT numbers by pressing the "***Start server***" button.
You can force the software to automatically start the server at launch (see section III.a => "*startserver=1*").

TEST CLIENT field:
This field allow you to simulate a remote client sending string keywords to the server using sockets.
Press the little arrow on the right of the field to see the list of available keywords.
Select a keyword and press enter to send the command to the server.
The server control log prints out the command receive by the server and the answer sent to the client.

b) Make your own CLIENT function

You can create your own client function to communicate with the server using the SOCKET library.
The socket type is : (*AF_INET, SOCK_STREAM)*

Below two examples in Python.
The first example "*SimpleTestClient*" is very simple and handles all commands except the image data transfer from server to client.
The second example "*TestClient*" show you how to receive the image data while sending the keywords "GetImage" or "GetNewImage".

IMPORTANT NOTE:
Each time you want to send a keyword, you have to open the socket, send the command and close the socket (i.e. one socket per command).

```python
ip = "168.194.102.101"
port = 50000
message = "One of the keywords"

def SimpleTestClient(ip, port, message):
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.connect((ip, port))
    answer = ""
    sock.send("%s\n"%message)
    answer = sock.recv(4096)
    sock.close()
    return answer
```

```python
def TestClient(ip, port, message):
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.connect((ip, port))
    answer = ""
    sock.send("%s\n"%message)

    if message.lower() in ["getimage","getnewimage"]:
        rep = sock.recv(1024)
        ans = rep.split(';')
        if len(ans)==4:
            mode,nx,ny,data_len = ans
        elif ans[0]=="I" and ans[1]=="16":
            mode,nx,ny,data_len = "I;16",ans[2],ans[3],ans[4]
        else:
            return False

        nx,ny,data_len = int(nx),int(ny),int(data_len)
        data = ""
        while 1:
            rep = sock.recv(10240)
            data = "".join([data,rep])
            if len(data)>=data_len:
                break

        answer = (mode,(nx,ny),data)

    else:
        answer = sock.recv(4096)

    sock.close()
    return answer
```

c) Keywords list

Below the list of string keyword you can send to the server. See following examples for usage.

KEYWORD	DESCRIPTION AND RETURNED STRING
Hello	get the *SoftwareName* and the *SoftwareVersion.*
GetCamList	list of camera names the software can open.
GetOpenCam	list of cameras already open in the software.
Select;*camera_name*	select the camera with the name *camera_name* among open cameras.
GetCam	get the name of the camera currently used by the server (active camera).
Open;*camera_name*	open the camera named *camera_name*.
Close	close the current camera.
Stop	stop a live acquisition and cancel the current exposition.
Start	start a live acquisition.
Snap	start a single image acquisition.
Abort	abort the image acquisition.
SetAutoSave;*enable*	save the last acquired image to disk using the saving path (*enable* = 0/1).
SetRecordPath;*dir*	set the saving path directory. (ex: *dir* = "*C:\\mydirectory\\test*")
SetRecordName;*name*	set the saving path file name. (ex: *name* = "*frame*")
SetRecordTag;*txt*	add text *txt* to the 'description' TIFF tag when saving image as TIFF file.
SetRecordFormat;*fmt*	set the saving image format. (ex: *fmt* = "*tif*" or "jpg" or "png" ...)
SetRecordNumber;*num*	set the counting number that extend the file name (auto-incremented).
GetSavePath	returns the full path used to save images on disk (see *SetAutoSave)*. path = "*RecordPath**RecordName_RecordNumber.RecordFormat*"
GetImage	return the image data as multiple packets. Blocks until the image is post-processed (see IMPORTANT NOTE below).
GetNewImage	same as *GetImage* but blocks until a new image is available. If called a second time without any new image acquisition, it will block until a Snap or Start command is sent and completed. See *HasNewData* command.
HasNewData	returns "*True*" if a new image is available, else returns "*False*". This flag is set to True after each new image acquisition.
DataReady	returns "*True*" if image is acquired and post-processed, else returns "*False*".
WaitForImage	block until the image is acquired and post-processed.
GetMode	returns the camera image mode ("*L*"=8bit mono, "*I;16*"=16bit mono).
GetSize	returns the current camera image size.
ResetCounter	reset the saved file counter to zero (see SetRecordNumber).
SetExpoMS;*value*	set the camera exposure time in milliseconds.
ShellCMD;*cmd*	pass the string "*cmd*" to the shell of the computer running the server.
GetOptionRange;*option_name*	get the authorized range for the possible values of a camera option.
GetOptions	returns the list of all camera options (see IMPORTANT NOTE below).

GetTemperature	returns the camera temperature (if available).
GetStats	returns the statistics tool measurements (if tool activated at the server level).
EnableStats;enable	enable/disable the statistics tool measurements (*enable* = 0/1).
Get*OptionName*	returns the value of the camera option *OptionName* (see IMPORTANT NOTE below).
Set*OptionName;value*	set the value of a camera option *OptionName* (see IMPORTANT NOTE below).
CreateGrid*;nx;ny;name*	creates a grid image named *name* to store *nx***ny* images.
FillGrid*;name;tilenum*	fill the tile number *tilenum* of the grid image named *name* with the last acquired image.
SaveGrid*;name;path*	Save the grid image *name* using the given *path*.
Loadbackground;*path*	Load the file *path* as a background image.

IMPORTANT NOTE:

To find the available camera options, use "GetOptions".
It returns all the camera options as a list of option names.
Ex: send("GetOptions") => ["Binning", "Exposure", "BrightPixel", ...]

Then use the command "Get*OptionName*" to get the current option value.
Ex: send("GetBinning") => (2, 2) (Current Binning is 2x2)

And use the command "Set*OptionName;value*" to set this option to a new value.
Ex: send("SetBinning;(3,3)") => True (Now Binning is 3x3)

The keywords GetImage or GetNewImage, must be handled in a special way.
After sending this keyword, you will receive the image data as a long string split into multiple packets. The first receive gives you information about the image data, then loop on multiple receives until you grab all the image data.

```
Ex: send("GetImage")  >>> socket.send("GetImage\n")
    first receive      >>> answer = socket.recv(1024)
    get the data info  >>> mode, width, height, data_length = answer.split(";")
    loop on receives   >>> while length(data)<data_length:   data=data.join( socket.recv(1024) )
```

See the Python code example of the "TestClient" function above (III.b).

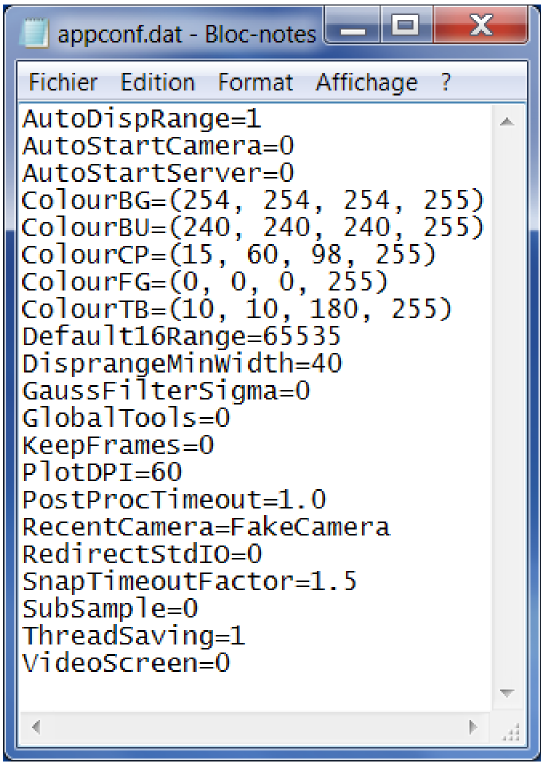
Example of an image acquisition sequence:

```
#Open the camera  (not required if the camera is already opened in the server application)
>TestClient("Open;FakeCamera")

#Define the image saving path
>TestClient("SetRecordPath;MyDir")
>TestClient("SetRecordName;MyImageName")
>TestClient("SetRecordTag;MyImageComments")

#Enable the image auto-saving
>TestClient("SetAutoSave;1")

#Snap an image
>TestClient("Snap")

#Start a Live
>TestClient("Start")

#Grab the Image data (block until the image is acquired and processed)
#Set the 'HasNewData' flag to False
>TestClient("GetImage")

#Grab the Image data (block until a new image is available)
#Set the 'HasNewData' flag to False
>TestClient("GetNewImage")

#Acquire multiple frames
>TestClient("Snap")
>TestClient("GetImage")
>TestClient("Snap")
>TestClient("GetImage")
> ….
or
>TestClient("Start")
>TestClient("GetNewImage")
>TestClient("GetNewImage")
> ….

#Perform a Scan/MultiImageGrid
#Create a Grid Image named 'testgrid' of size 2x2 (to store 4 images)
>TestClient("CreateGrid;2;2;testgrid")
#Acquire an image and Fill the tile number 0 of the Grid Image 'testgrid'
>TestClient("Snap")
>TestClient("FillGrid;testgrid;0")
#Fill next tiles
>TestClient("Snap")
>TestClient("FillGrid;testgrid;1")
>TestClient("Snap")
>TestClient("FillGrid;testgrid;2")
>TestClient("Snap")
>TestClient("FillGrid;testgrid;3")

#Close the camera
>TestClient("Close")
```

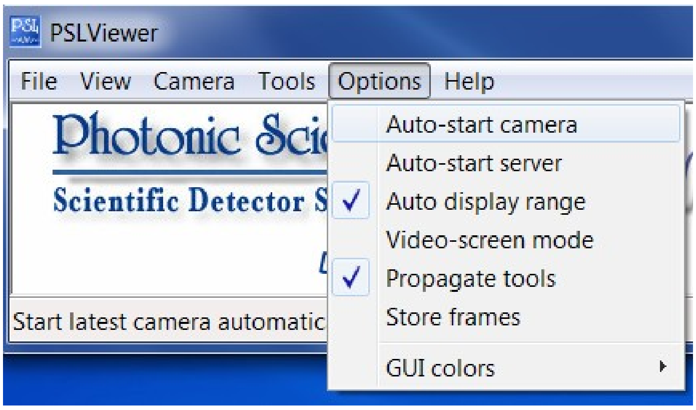
IV. Advanced setup

a) Software configuration file

The software behaviour can be adjusted using the file "appconf.dat" (main software folder). Some options can be changed via the software menu bar, in the "Options" menu.




```
AutoDispRange=1
AutoStartCamera=0
AutoStartServer=0
ColourBG=(254, 254, 254, 255)
ColourBU=(240, 240, 240, 255)
ColourCP=(15, 60, 98, 255)
ColourFG=(0, 0, 0, 255)
ColourTB=(10, 10, 180, 255)
Default16Range=65535
DisprangeMinWidth=40
GaussFilterSigma=0
GlobalTools=0
KeepFrames=0
PlotDPI=60
PostProcTimeout=1.0
RecentCamera=FakeCamera
RedirectStdIO=0
SnapTimeoutFactor=1.5
SubSample=0
ThreadSaving=1
VideoScreen=0
```



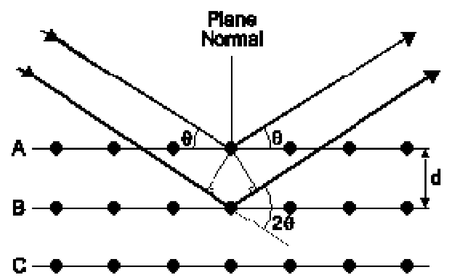
- ***AutoDispRange***: (1 or 0) Set the automatic display range as default behavior.
- ***AutoStartCamera***: (1 or 0) Start a live acquisition as soon as the camera is open.
- ***AutoStartServer***: (1 or 0) Auto-start the server tool when launching the software.
  The server will try the current first available IP number to start the server (port 50000).
  You can specify a specific IP number to start the server creating a file "*serverip.dat*" in the main software folder ("*PSLViewer\serverip.dat*") and adding a line like this: "*IP=132.186.1.14*". Use "#" to comment.
- ***ColourXX***: (r,g,b,a) Colour of the different element of the GUI.
- ***Default16Range***: (4095 or 65535) Default maximum display range for 16bit images.
- ***DisprangeMinWidth***: (0 to 255) Minimum display range width (not used).
- ***GaussianFilterSigma***: (0 to inf) Width of the gaussian filter used for auto-display range.
- ***GlobalTools (Propagate)***: (1 or 0) Propagate the tool area selection to other image windows.
- ***KeepFrames (Store frames)***: (1 or 0) Each image snap is stored in a different window.
- ***PlotDPI***: (20 to 140) Resolution of the graphic plots.
- ***PostProcTimeout***: (0.0 to inf) (second) Skip frame if post-process is longer than timeout.
- ***RecentCamera***: (camera_name) Store the name of the last camera used.
- ***RedirectStdIO***: (1 or 0) Error and warning messages are redirected to the "*log.dat*" file .
- ***SnapTimeoutFactor***: (1.0 to inf) Skip frame if camera does not return an image before *exposure**SnapT*imeoutFactor*.
- ***SubSample***: (0 to inf) The auto display range works on a sub-sampled image taking only 1pixel each *SubSample_value* pixels.
- ***ThreadSaving***: (1 or 0) On disk image saving is threaded. May improve performances while recording a live acquisition.
- ***VideoScreen***: (1 or 0) The camera output is displayed full screen and the GUI is hidden.

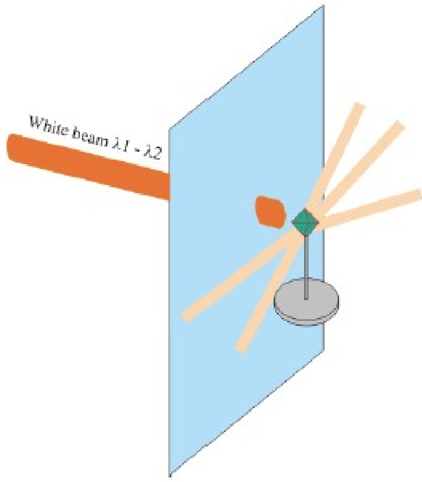
V. Laue orientation tool

a) Introduction

Laue experiment consist in a x-rays diffraction by a crystal. A polychromatic x-rays ($\lambda 1<\lambda<\lambda 2$) beam shine on a crystal, and the diffracted beams are digitized by the Laue camera (in the blue plane). When the beam arrives on the crystal, it is diffracted and diffraction peaks of appears, corresponding to the reciprocal lattice of the crystal. Indeed, each family of planes (hkl) correspond to a diffraction peak (the more atoms in these planes, the more intense the peaks will be), according to the Bragg s law:

$n*\lambda = 2*dhkl*\sin(\Theta)$ with n integer.

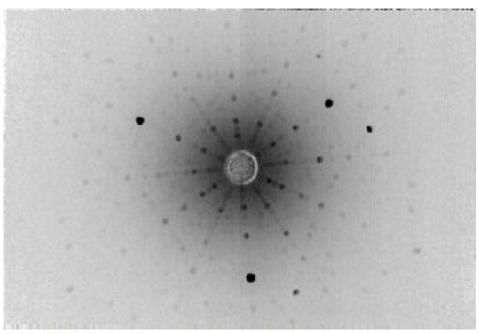




The Laue method allows to draw a map of the reciprocal lattice, (each spot corresponding to a diffraction peak and a group of <hkl> values). By using this map and the knowledge of the crystal parameters , it is possible to determine the orientation of the sample.

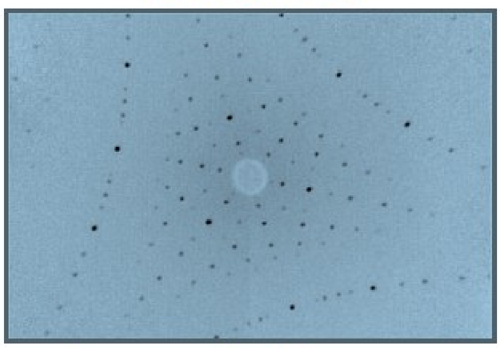
Three collimators are provided with the Laue camera: 0.7mm, 1mm and 1.4mm. The 1.4mm collimator will deliver a bigger beam onto the sample, resulting in enlarged diffraction spots. It is used for a quick overview. Should the background be too high, a 1mm collimator will reduce the beam size / background and deliver smaller and well focused diffraction spots. Most of users are using 1mm collimator as the beam size delivered by the source is typically 1.26mm diagonal. The 0.7mm collimator is generally used when the background originating from the sample is very high, it will cut the beam intensity by a factor 2 compared to the 1mm collimator, and 4 with respect to the 1.4mm collimator, hence extending exposure.

1.4mm collimator



Sapphire sample, 100sec exposure

1mm collimator



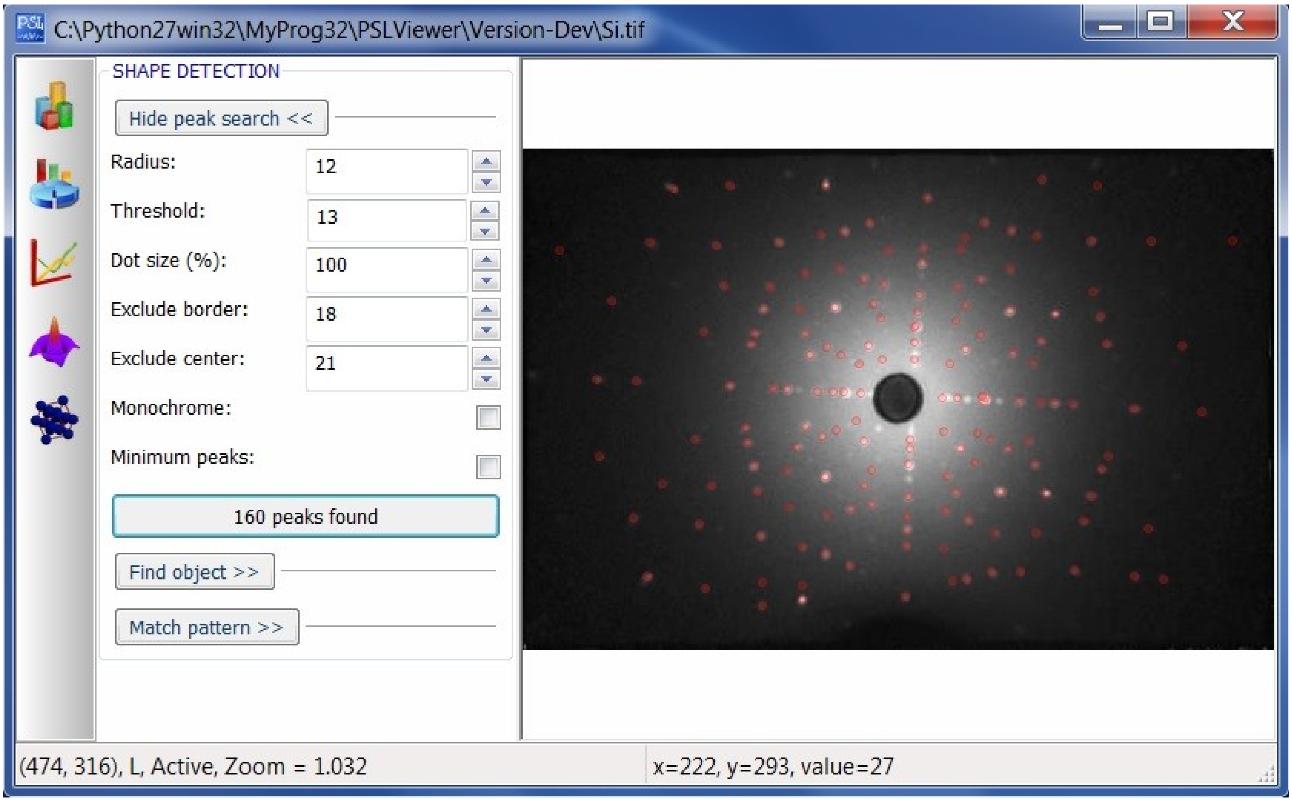
Tungsten sample, 300sec exposure

Depending on the type of source target used: W, Mo or Cu, and type of samples, exposure and kV/mA setting will vary. Typical exposure for Silicon is 2 to 3minutes using a Mo Long Fine Focus source at 35kV and 30mA with a 1mm collimator. Background fluorescence can be avoided by turning kV down to 20kV. A Tungsten source will deliver more power, hence requiring less kV, typically 25kV.

b) Experimental spot detection

The first step of the orientation process consists in the experimental spot detection.
Acquire or open an image of a Laue diffraction pattern, then in the tool bar, click on:

The  "***Shape detection***" tool icon.



The tool panel appears on the left side of the image window. The section "Peak search" shows the different parameters required for the peak detection method.

- ***Radius***: the average radius of the spots you want to detect (in pixels)
- ***Threshold***: the minimum intensity value of the spots you want to detect (in counts)
- ***Dot size***: the size of the red detection dots drawn over the image.
- ***Exclude border***: define the width of an excluded rectangular region around the image borders .
- ***Exclude centre***: define the radius of an excluded circular region around the centre of the image.
- ***Monochrome***: forces the detection to be performed on a monochromatic version of the image.
- ***Minimum peaks***: search for minimum peaks instead of maximum.

The button below the parameters list, displays the number of peaks found or any error message relative to the peak detection. Pressing the button will show you the detected peaks into red circles.

Important notes:
If too many dots are detected in the image, you will need to adjust the parameters in order to reduce the number of solutions. With the appropriate parameters the red dots should cover most of your Laue experimental spots but be sure to avoid fake spots such as those introduced by the background noise or any artefact in the image. It is better to detect less real spots but no fake spot at all than covering all the real spots but with some fake spots among.

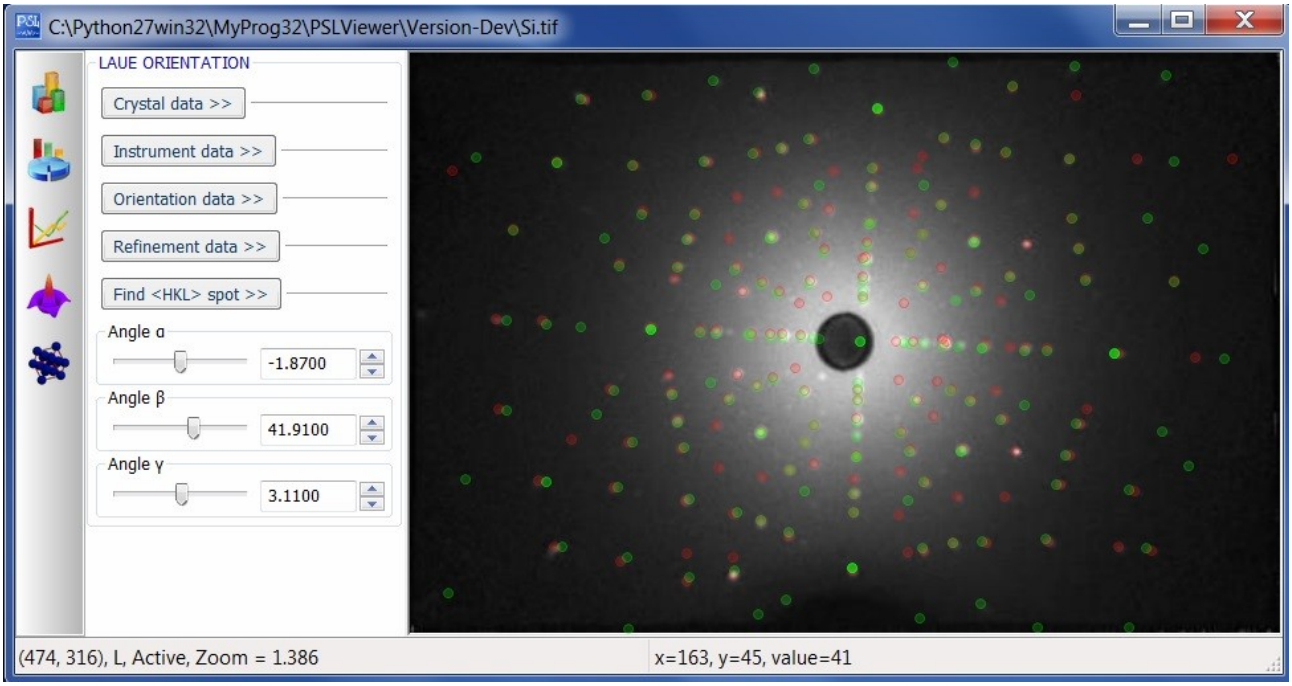
Once you have found the best detection parameters you can close the tool by clicking again on the "***Shape detection***" icon (in the tool bar). The last detection is stored in memory and will be displayed in the "***Laue Orient***" tool (see next step). Also the peak search parameters are stored between sessions

c) Theoretical Laue pattern generation

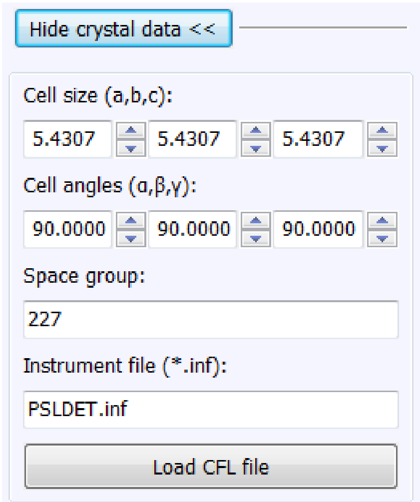
The second step of the orientation process consists to define the crystal type and the instrument (detector) characteristics. With this parameters the software will be able to draw a theoretical Laue pattern. The theoretical pattern is drawn on top of the image with green dots. The detected experimental spots are displayed with red dots (see section V.b).

To display both theoretical and detected spots, in the tool bar click on:

The  "***Laue Orient***" tool icon.



Crystal data:



=> define the unit cell lengths (angstrom)

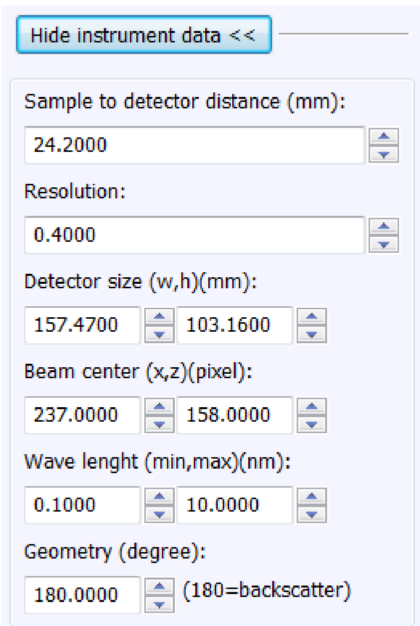
=> define the unit cell angles (degree)

=> define the crystal space group (number or intl)

=> name of the instrument file.
(“PSLDET.inf” as default)
(!usualy not to be modified!)

For more details about the crystallographic conventions and structure data see the appendix of this manual (“Space group table”) and visit the http://www.crystallography.net/result.php website.

Instrument data:



=> Distance between the detector plane and the crystal

=> Wavelength resolution, typically in the range [0.4:0.7]. Low value produce more theoretical spots.

=> The width and height of the detector plane

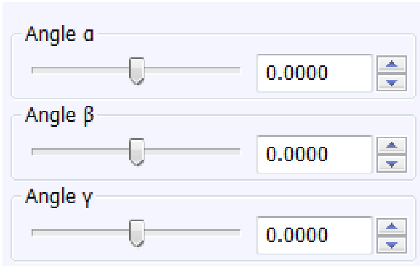
=> The beam position which should correspond to the centre of your image, so half the image size (in pixels)

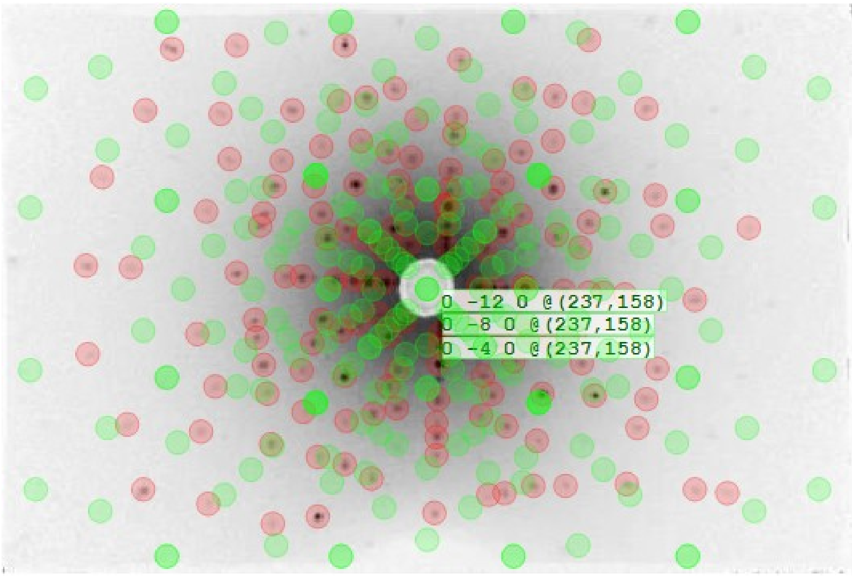
=> Simulated wavelengths; [0.1:10] is usually enough.

=> The angle between the beam axis and the detector normal axis. The Laue Backscatter configuration correspond to an angle of 180°.

The “***Sample to detector distance***” and “***Detector size***” parameters are critical and must be measured accurately. An error about one millimetre could prevent the Orientation Algorithm to find a solution in a reasonable time.
It is very recommended to do a calibration of these parameters with a sample for which you know the orientation already.

Once you have defined your Crystal and the Instrument, you can use the "***Orienation angles***" sliders to modified the theoretical Laue pattern (green dots). The three angles correspond to the three possible rotation of the cell unit around its cell axes.

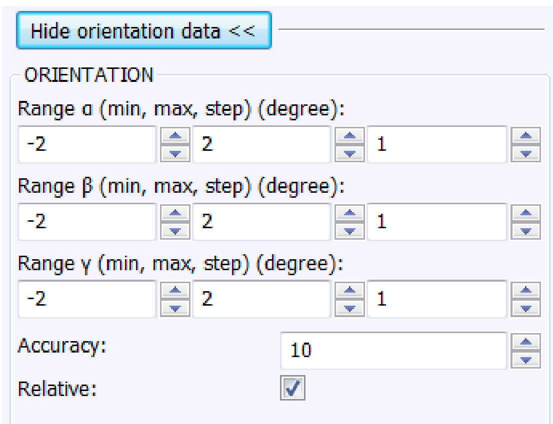




If you drag the mouse cursor over the a green dot, a label will display the <hkl> values corresponding to this spot (may have more than one solution).

d) Orientation algorithm and refinement

Orientation data:



=> Angle α: (from, to , step size)

=> Angle β: (from, to , step size)

=> Angle γ: (from, to , step size)

=> Between [10:15]; 10 is the best
=> Search around current angle values

The Orientation Algorithm will produce theoretical Laue pattern for all the angle combinations in the defined range.
If the "***Relative***" option is OFF you must enter the absolute value for the min and max of the search range for each angle (α,β,γ). Else if the option is ON, the range will be define around the current (α,β,γ) angle values (see "***Orienation angles***" sliders).

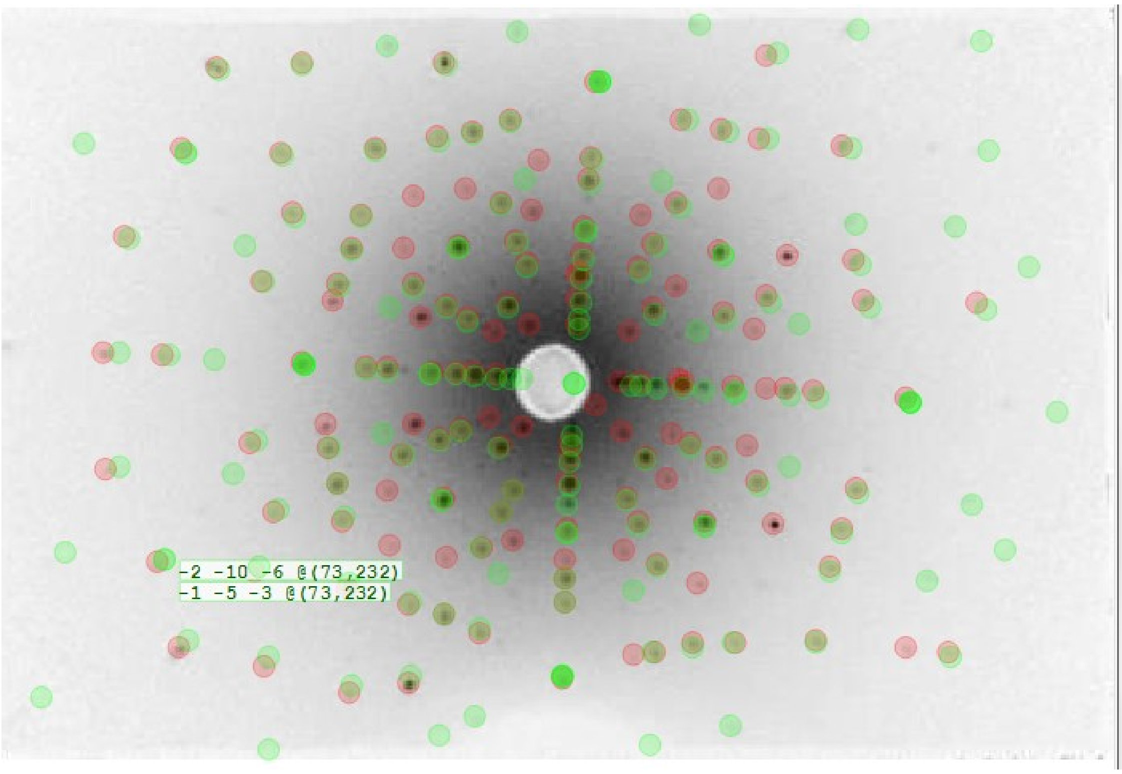
Warning: the time to process all solutions could be very long (hours) if you set a too wide range for the search angles. You can adjust the step size to reduce the number of steps while scanning on angles but increasing the step size also reduce your chances to find the solution.
Best is to keep the step value equal to 1 degree and make a "Relative" search in a +/- 2 degrees range (see the picture of the orientation panel above).

Press 

to start the orientation process.

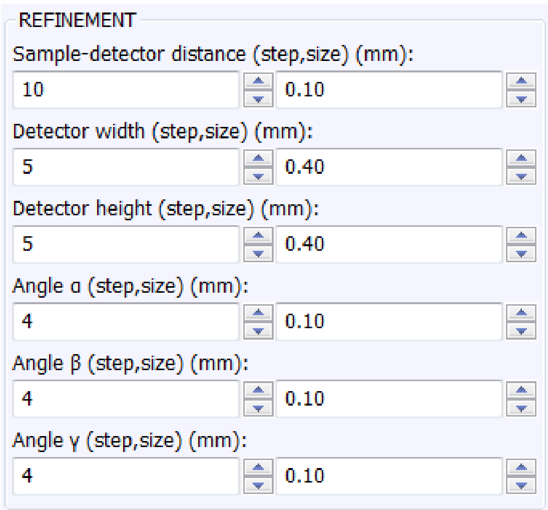
Press again to interrupt.

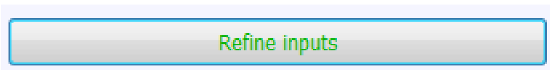
Wait until it finishes; the solution (green dots) is displayed at the end of the process.
If both red an green patterns do not match, make sure your Crystal data are correct and adjust more accurately your Instrument parameters.



Refinement data:
Once you get close to a solution, you can try to refine some of the parameters used in the orientation process. The data refinement consist to recompute the current solution but slightly modifying one or more parameters and check if the overlay between the theoretical and the experimental patterns can be improved.
For each parameter select the number of step and the step size. The search will be performed around the current parameter value.
E.g: if current distance=20, step=10 and stepsize=0.1 then the tested solution will be in the range [(20 – 5*0.1) : (20+5*0.1)]

Keep other parameter fixed by setting the step value to zero.



Press 

to start the refinement process.

Press again to interrupt.

APPENDIX

Table of space groups in 3 dimensions

#	Crystal system	Point group		Space groups (international short symbol)
		Intl	Schönflies	
1	Triclinic (2)	1	C_1 Chiral	P1
2		1	C_i	P1
3–5	Monoclinic (13)	2	C_2 Chiral	P2, $P2_1$, C2
6–9		m	C_s	Pm, Pc, Cm, Cc
10–15		2/m	C_{2h}	P2/m, $P2_1/m$, C2/m, P2/c, $P2_1/c$, C2/c
16–24	Orthorhombic (59)	222	D_2 Chiral	P222, $P222_1$, $P2_12_12$, $P2_12_12_1$, $C222_1$, C222, F222, I222, $I2_12_12_1$
25–46		mm2	C_{2v}	Pmm2, $Pmc2_1$, Pcc2, Pma2, $Pca2_1$, Pnc2, $Pmn2_1$, Pba2, $Pna2_1$, Pnn2, Cmm2, $Cmc2_1$, Ccc2, Amm2, Aem2, Ama2, Aea2, Fmm2, Fdd2, Imm2, Iba2, Ima2
47–74		mmm	D_{2h}	Pmmm, Pnnn, Pccm, Pban, Pmma, Pnna, Pmna, Pcca, Pbam, Pccn, Pbcm, Pnnm, Pmmn, Pbcn, Pbca, Pnma, Cmcm, Cmce, Cmmm, Cccm, Cmme, Ccce, Fmmm, Fddd, Immm, Ibam, Ibca, Imma
75–80	Tetragonal (68)	4	C_4 Chiral	P4, $P4_1$, $P4_2$, $P4_3$, I4, $I4_1$
81–82		4	S_4	P4, I4
83–88		4/m	C_{4h}	P4/m, $P4_2/m$, P4/n, $P4_2/n$, I4/m, $I4_1/a$
89–98		422	D_4 Chiral	P422, $P42_12$, $P4_122$, $P4_12_12$, $P4_222$, $P4_22_12$, $P4_322$, $P4_32_12$, I422, $I4_122$
99–110		4mm	C_{4v}	P4mm, P4bm, $P4_2cm$, $P4_2nm$, P4cc, P4nc, $P4_2mc$, $P4_2bc$, I4mm, I4cm, $I4_1md$, $I4_1cd$
111–122		42m	D_{2d}	P42m, P42c, $P42_1m$, $P42_1c$, P4m2, P4c2, P4b2, P4n2, I4m2, I4c2, I42m, I42d
123–142		4/mmm	D_{4h}	P4/mmm, P4/mcc, P4/nbm, P4/nnc, P4/mbm, P4/mnc, P4/nmm, P4/ncc, $P4_2/mmc$, $P4_2/mcm$, $P4_2/nbc$, $P4_2/nnm$, $P4_2/mbc$, $P4_2/mnm$, $P4_2/nmc$, $P4_2/ncm$, I4/mmm, I4/mcm, $I4_1/amd$, $I4_1/acd$
143–146	Trigonal (25)	3	C_3 Chiral	P3, $P3_1$, $P3_2$, R3

147–148		3	S_6	P3, R3
149–155		32	D_3 Chiral	P312, P321, P3$_1$12, P3$_1$21, P3$_2$12, P3$_2$21, R32
156–161		3m	C_{3v}	P3m1, P31m, P3c1, P31c, R3m, R3c
162–167		3m	D_{3d}	P31m, P31c, P3m1, P3c1, R3m, R3c,
168–173		6	C_6 Chiral	P6, P6$_1$, P6$_5$, P6$_2$, P6$_4$, P6$_3$
174		6	C_{3h}	P6
175–176		6/m	C_{6h}	P6/m, P6$_3$/m
177–182	Hexagonal (27)	622	D_6 Chiral	P622, P6$_1$22, P6$_5$22, P6$_2$22, P6$_4$22, P6$_3$22
183–186		6mm	C_{6v}	P6mm, P6cc, P6$_3$cm, P6$_3$mc
187–190		6m2	D_{3h}	P6m2, P6c2, P62m, P62c
191–194		6/mmm	D_{6h}	P6/mmm, P6/mcc, P6$_3$/mcm, P6$_3$/mmc
195–199		23	T Chiral	P23, F23, I23, P2$_1$3, I2$_1$3
200–206		m3	T_h	Pm3, Pn3, Fm3, Fd3, Im3, Pa3, Ia3
207–214	Cubic (36)	432	O Chiral	P432, P4$_2$32, F432, F4$_1$32, I432, P4$_3$32, P4$_1$32, I4$_1$32
215–220		43m	T_d	P43m, F43m, I43m, P43n, F43c, I43d
221–230		m3m	O_h	Pm3m, Pn3n, Pm3n, Pn3m, Fm3m, Fm3c, Fd3m, Fd3c, Im3m, Ia3d