Friedrich-Schiller-Universität Jena
Physikalisch-Astronomishe Fakultät
Abbe School of Photonics

seit 1558

Characterization of the Guided Second-Harmonic Signal in Lithium Niobate Nanowires by Fourier Imaging

Internship Report

Barbry Marc[1]
University of Bordeaux 1, Physics Department

Supervisors: Anton Sergeyev and Rachel Grange

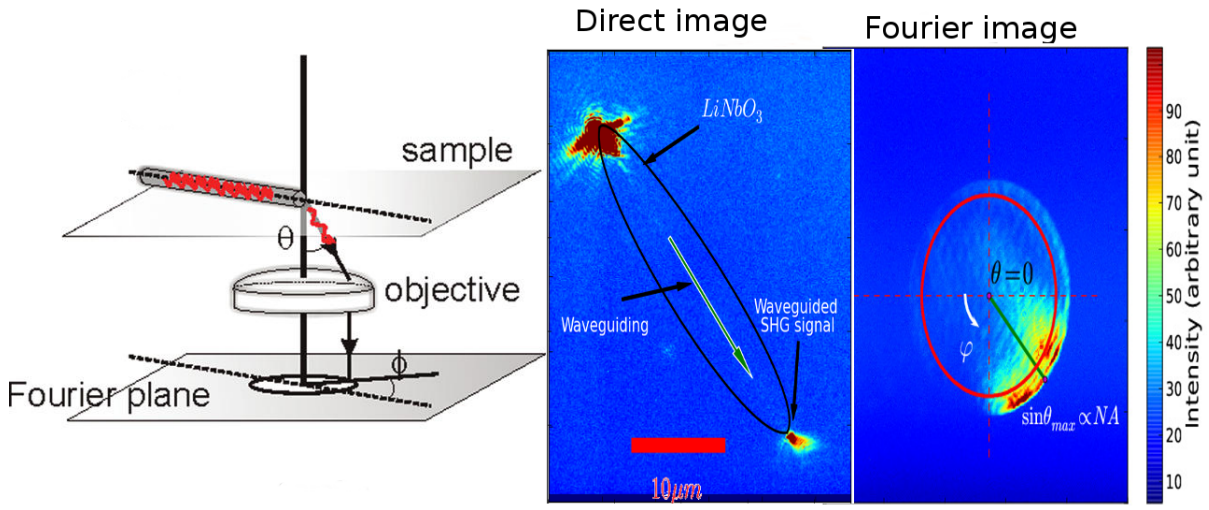
Jena, Germany

5 August 2013

[1] marc.barbry@mailoo.org

Abstract

The combination of nanomaterials and optics creates a huge exploration field for research. Spatial confinement leads to modification of optical properties resulting in new effects which can eventually be used to develop novel applications.
Apart of other types of nanomaterials, nanowires (NWs) show combination of two-dimensional confinement and third macroscopic dimension. As a result, they can be used for certain applications such as lasing, light generation and light transfer.
In this work, we study lithium niobate ($LiNbO_3$) NWs which are capable of producing and guiding second-harmonic (SH) signal. Moreover, it has been shown that the delivered SH signal is strong enough for dye excitation. As a result, the NWs of such kind can be applied for localized imaging of tiny object like cells. However, the previous studies do not give answer on light distribution at the NW output.
During this internship, I apply the Fourier optics for characterizing the SH signal distribution at the NW input and output. In this report, I describe the built experimental setup which performs Fourier transformation of the signal and I discuss the first obtained experimental results. Notably, I show that the output SH signal has the shape of a torch.



Contents

1 Introduction

The usual size of the nanomaterials is in the range of $1 - 100nm$[1]. Despite their small sizes, the fabrication and the use of such kind of materials are developing very quickly. Since the size of the NWs is comparable or even smaller than the wavelength of the light, the combination of optics and nanomaterials lead to numerous interesting effects. As a consequence, NWs can be applied in various fields such as NW lasing[2], photovoltaic applications[3] and nanoendoscopy[4].
Recently, a very interesting approach has been suggested to use NW in biological studies for performing localized imaging inside a living cell[5]. Previous experiments have already shown that it is possible to illuminate a cell locally by guiding light in a NW which is introduced into a cell[5].
However, this technique can be improved by introducing infrared light (IR). First of all, the IR is less scattered and absorbed. As a result, it can penetrate deeper in water and living tissue itself and cause less damage to the cell[6]. However, most of the fluorescent materials are excited in the visible range. Thus, an interim process is required which will transfer the delivered and coupled into a NW IR signal into the visible light. One of the solutions is application of a nonlinear process like second-harmonic generation (SHG).
For this application we suggest to use niobate ($NaNbO_3, KNbO_3, LiNbO_3$) NWs as they maintain nonlinear properties even at nanoscale[7], [8], [9]. Moreover, research on biocompatibility of nanoparticles of various crystals has shown lower cytotoxicity of $KNbO_3$ and $LiNbO_3$ nanoparticles [10]. In addition, it has been shown that the $LiNbO_3$ NWs guide enough SH signal for detectable dye excitation[11]. However, the published works leave the distribution shape of the guided light unstudied, though it is a very important factor for the application of localized imaging.
Nevertheless, the distribution of the guided signal can be studied by means of the Fourier transfromation. Applying this technique Fourier transformation one can decompose the studied signal into plane waves with various wave-vectors and image their spectrum. Using the spectrum, one can easily reconstruct the shape of the output signal[12].
Thus, the main goal of this internship is to characterize the distribution of the SH guided signal by means of Fourier transformation. In this report, I discuss the building up and characterization of the experimental setup. In addition, I discuss the first obtained experimental results of Fourier transformed SH signals produced by the NWs.

2 Theoretical background

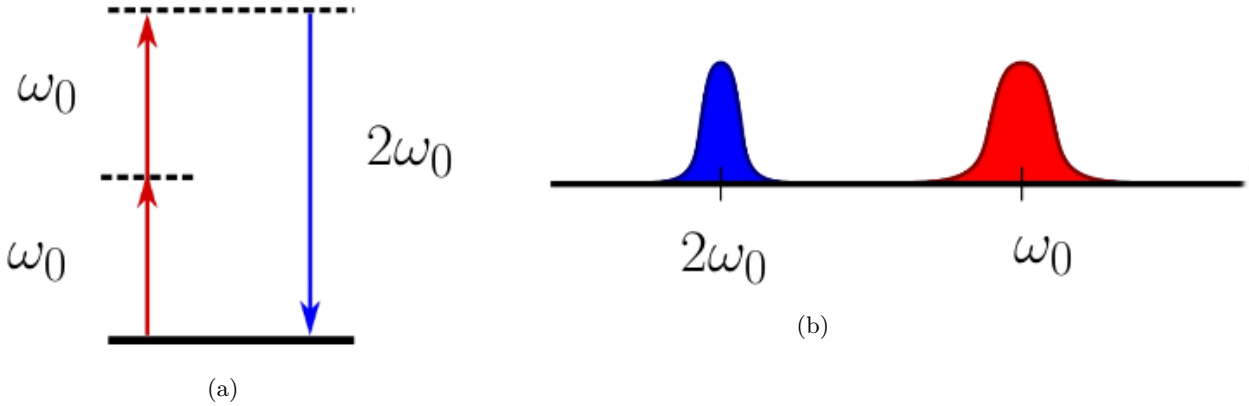
2.1 Second Harmonic Generation(SHG)

SHG is a well known nonlinear effect which has applications in various fields such as SHG microscopy[13]; and cancer cells treatment[14]. SHG is a quadratic effect which combines two photons of the same frequency into a single photon of the doubled frequency as shown in figure 2.1 and as result twice smaller wavelength[15].
The SHG effect as most of other nonlinear effects take place only when high power of the laser irradiation are involved. Nevertheless, efficient SHG can be achieved only in noncentrosymmetric materials. In centrosymmetric crystals, only weak SH signal is generared by broken crystal structure at the crystal surface[16]. The polarization density of the SH signal $\mathbf{P}(2\omega)$ is described by the following equation [15]:

$$\mathbf{P}(2\omega) = 2\varepsilon_0\chi^{(2)}\mathbf{E}^2(\omega) \quad (1)$$

Where $\varepsilon_0 = 8.85 \times 10^{-12} F \cdot m^{-1}$ is the vacuum permittivity, $\mathbf{E}(\omega)$ is the incident electric field of the fundamental wavelength, $\chi^{(2)}$ is the second order nonlinear electric susceptibility.

The advantage of the SHG signal is that it uses virtual energetic levels. As a result, the SHG in contrast to dye fluorescence does not show neither bleaching nor blinking.



(a) (b)

Figure 2.1: Frequency doubling in a nonlinear medium, (a) Energy diagram, (b) Optical spectrum

2.2 Fourier Optics

Fourier optics is used during this experiment in order to characterize the shape distribution of the SH signal. Here we presentet a short overview of the Fourier optics.
The general idea of the Fourier optics is to used a lens to performed Fourier transformation in the back focal plane of the lens of an object situated in the forward focal plane[17].
In this section the paraxial approximation (small angles to the optical axis) is used, which is sufficient for our experiment.
A thin lens changes only the phase of the optical field, the transfer function of a thin lens is given by

$$t_L(x,y) \quad = \quad e^{-i\frac{k}{2f}(x^2+y^2)} \tag{2}$$

Where f is the focal distance of the lens and $k = \frac{2\pi}{\lambda}$is the wave vector of the incident light and λ is the wavelength of the light. The Fourier transformation of the transfer function of the lens is:

$$T_L(\alpha,\beta) \quad = \quad -i\frac{\lambda f}{4\pi^2}e^{i\frac{f}{2k}(\alpha^2+\beta^2)} \tag{3}$$

Where $\alpha = \frac{k}{f}x$ and $\beta = \frac{k}{f}y$ are the projctions of the wavevector $\mathbf{k}$ on the axes of the cartesien coordinate system. From electric field $u_0(x,y)$ in the object plane, the following electric field is obtained in the focal plane of the lens:

$$u(x,y,2f) \quad = \quad -i\frac{4\pi^2}{\lambda f}e^{2ikf}U_0\left(\alpha,\beta\right) \tag{4}$$

Where $U_0(\alpha,\beta)$ is the Fourier transform of $u_0(x,y)$.

3 Experimental Setup and Characterization

3.1 Experimental Setup

In order to perform the measurements, the setup has been built (figure 3.3). For this experiment a continuous wave (CW) ND:YAG laser at $\lambda = 1064nm$ with an output power of $1W$ is used. This laser beam will cross several optical elements:

(a) a telescope which is used in order to increase the beam size. $f_1 = 50mm$, $f_2 = 150mm \Longrightarrow \frac{f_2}{f_1} = 3$. The telescope is used for decreasing the beam spot when it is focused by the focussing objective,

(b) the first half-wave plate and the polarised beam splitter are used to change the incident power onto the sample,

(c) the second half waveplate is used for changing the polarizarion of the input signal,

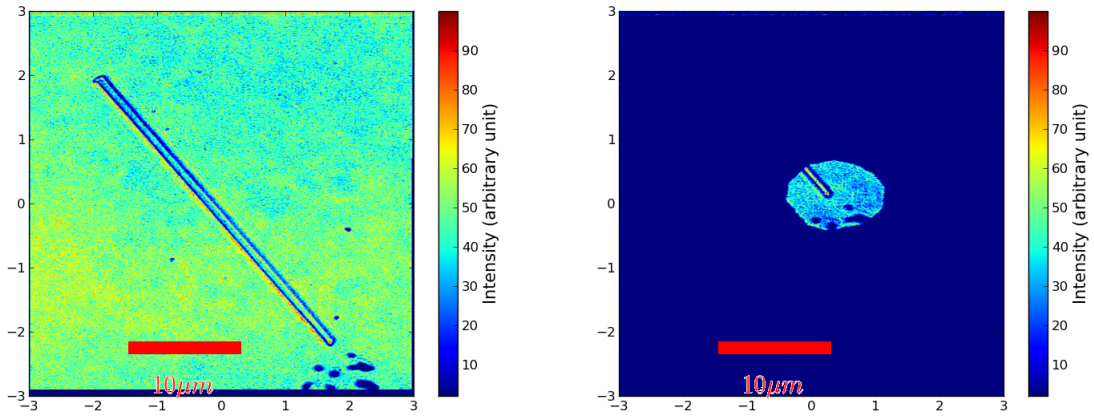
(d) the focusing 10X objective is used to focus the IR signal onto a NW,

(e) the sample with the NWs,

(f) an objective is used in order to collect the light from a the sample,

(g) the first lens L_1 ($f_1 = 150mm$) is located so that it builds up the image of irradiated SH signal in the forward focal plane,

(h) the aperture is placed in the back ocal plane so that the image can be cut and only a part of the image can be select for the study,



(a) (b)

Figure 3.1: Real image in white light for two aperture sizes, (a) the aperture is open, the signal from both sides is collected, (b) the aperture is partially closed, only the signal from the output side is collected.

(i) a beam splitter separates the signal in two differents arms:

- the first arm is composed of a lens L_2 with focal $f_2 = 150mm$, a bandpass filter $BG39$(B) and an electron multiplying coupled charge device (EMCCD). The EMCCD images the Fourier transformed SH signal.
- The second arm is composed of two lenses ($f_3 = 150mm$, $f_4 = 200mm$), a $BG39$ filter and a CCD camera (Watec 120N+) which is used to observe the SH signal.

The system L_1 and L_2 forms a $4f$ setup which hve no influence when the aperture is open, but if the aperture size becomes small, the aperture act s a mask situated in the fourier plane, and in this case the short frequencies are removed 3.2 . The principe of the 4-f setup is as follow:

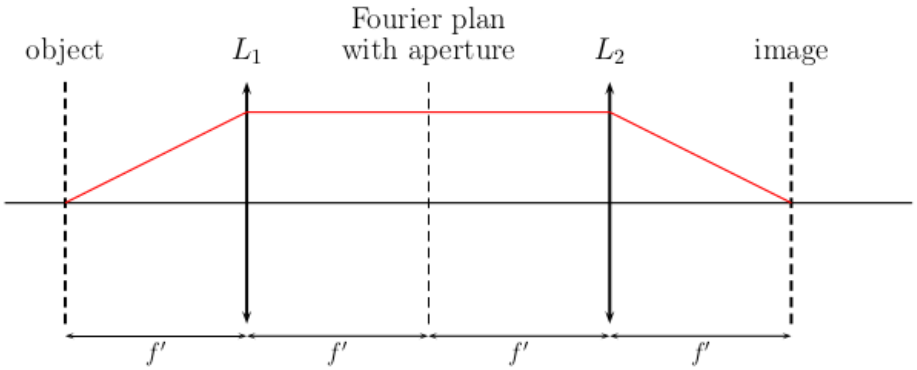


Figure 3.2: Typical 4-f setup

- a first thin lens performs the Fourier transformation of the object,
- a spatial filter in the Fourier plane manipulates the field,
- a second thin lens performs the Fourier back transformation.

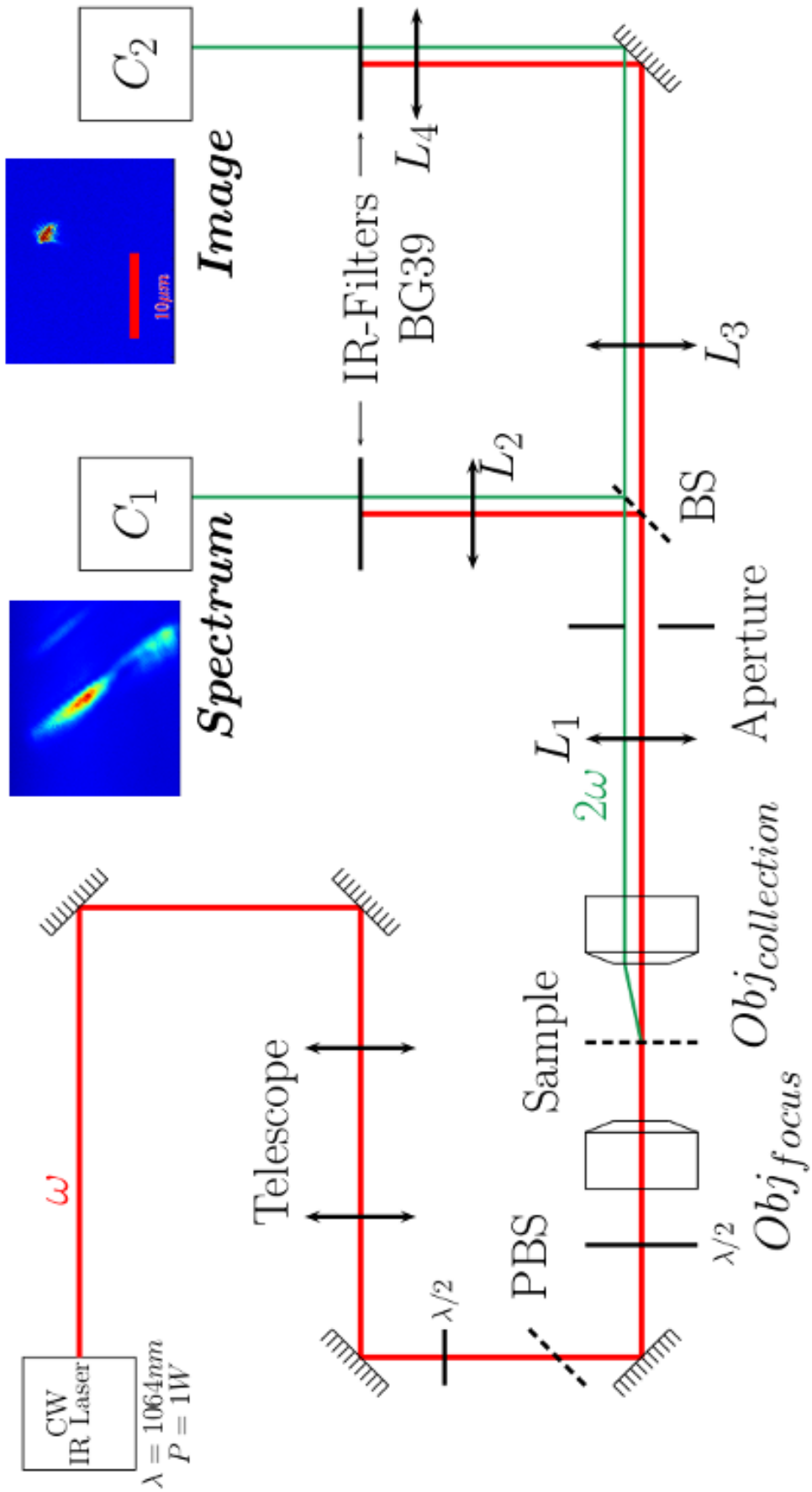


Figure 3.3: Setup diagram, see the detailed description in the text.

3.2 Building up of the setup

The building up of the setup requires thorough positioning which should be respected to enable correct measurements. The alignment has to be done with two different lasers:

- a CW laser diode at $\lambda_{SH} = 532nm$,
- a CW $ND : YAG$ laser at $\lambda = 1064nm$

The setup has to be aligned in the following order:

1. the first step is to align the mirrors before the focusing objective with the green laser,
2. the second step is to align and find the correct position of the elements after the focusing objective. In order to facilitate this step, rail and cage system have to be used:
   (a) the first thing to do is to place the Watec camera and the lens L_4, the Watec camera has to be in the focal plane of L_4,
   (b) then place the lens L_3: L_3 and L_4 form a $4f$ setup, therefore, a plane wave should be observed at the output of the lens L_4,
   (c) the next step is to place the lens L_1, so that a focused beam is obtained in the Watec camera,
   (d) to place the beam-splitter between the lens L_1 and L_3
   (e) afterwards, L_2 must be placed such as a plane wave is obtained at the output of $L2$
   (f) the EMCCD is placed in the focal plane of L_2, therefore, remove L_1 and place the EMCCD so a focused beam is observed on the EMCCCD,
   (g) bring back,
   (h) the position of the collection objective is easy to find, since it is combined with L_1 and L_2, you must obtain a focused beam on the EMCCD.

After this, the setup is pratically ready, you have just to do the step (1) with the IR laser, find the right position of the focusing objective is easier since you must obtain a focused beam on the Watec camera. After this, the sample has to be in the focal plane of the two objectives and measurement can be performed.

3.3 Fourier imaging method

In previous experiments [18] it has been shown that $LiNbO_3$ NWs can be used for guiding SH signal.

This signal can propagate inside the NW to the other side and we obtain an output signal (Figure 3.4a). Using the Fourier optics we image the wavevector distribution of the SH signals obseved at the input and output of a NW (figure 3.4a)[12]. As a result, we observe intensity distribution limited by a circle (figure 3.4b). The radius of the circle is proportionnal to the numerical aperture (NA) of the collection objective. Each point within the circle is identified by two angles φ and θ which correspond to a certain wavevector.
In figure 3.4c, the location of the angles φ and θ in respect to the NZ are shown. The angle φ is located in the plane perpendicular to the objective normal and varies between 0 and 2π. The angle θ is an angle between the normal of the objective and the corresponding wavevector. When the θ angles is equal to zero, it means that the wavevector is parallel to the objective normal and this point is located in the center of the circle. The angle θ increases radially and the maximal angle is defined by the numerical aperture of the objective:

$$\sin(\theta_{max}) = \frac{NA}{n} \tag{5}$$

Table 1: NA and corresponding θ_{max} of the different objectives, n is the refractive index of the objective.

objective	NA	n	θ_{max}
$100X$	0.75	1.5	30
$100X$ oil-immersion	1.3	1.518	58.9

During the internship, two objectives with different NAs are used. In table 1, the NAs of the objectives and the corresponding maximal θ_{max} are listed.

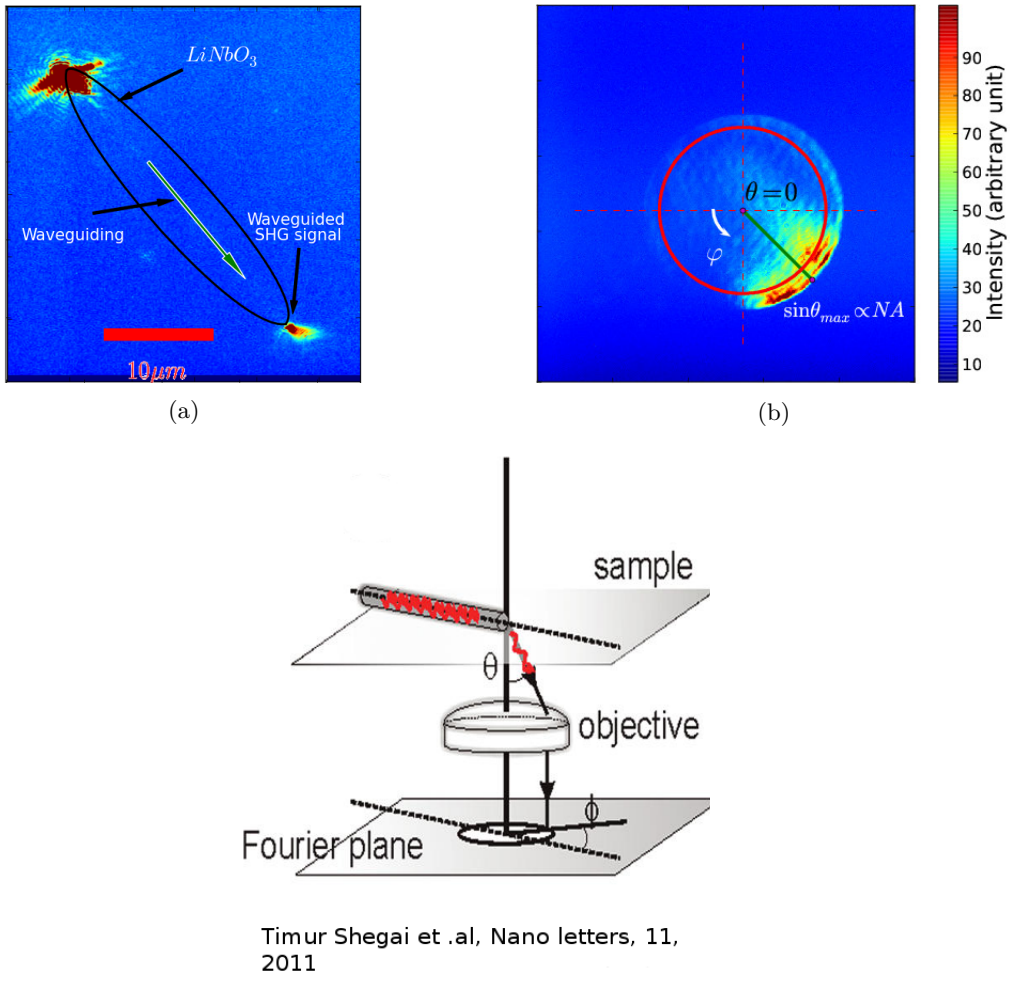


Figure 3.4: Unidirectional emission from a $LiNbO_3$ NW. (a) Illustration of the detection scheme and the coordinate system used. The NW is excited by a focused IR beam at the input end and the emitted light is detected in the Fourier plane of the optical microscope[12](b) Direct optical image of a wire, input signal in the top and guided signal in the down, (c) Corresponding Fourier image of the guided signal.

Figure 3.4c illustrates the principle of the experiment and a typical measurement is displayed. A $LiNbO_3$ NW immobilized on a glass slide is imaged by the collection objective. A laser beam $\lambda_{inp} = 1064nm$ is focused on one side of the NW (upper side) which produced second harmonic signal ($\lambda_{SH} = 532nm$). The SH signal is propagated inside the NW and the output signal is observed at the other side

of the NW.

3.4 Setup characterization with a square

In order to check if the setup is correctly built, we place a mask with a square openning in the sample position and illuminate with the green laser. Ze obtain the real and Fourier image of the mask. Using the image we retrieve the size of the square. To find the size of the square opening with the real image we characterize the pixelsize of the Watec camera $PX_{size-wat}$ of the image with a reference sample, it is find:

$$PX_{size-wat} = 0.06802\mu m \tag{6}$$

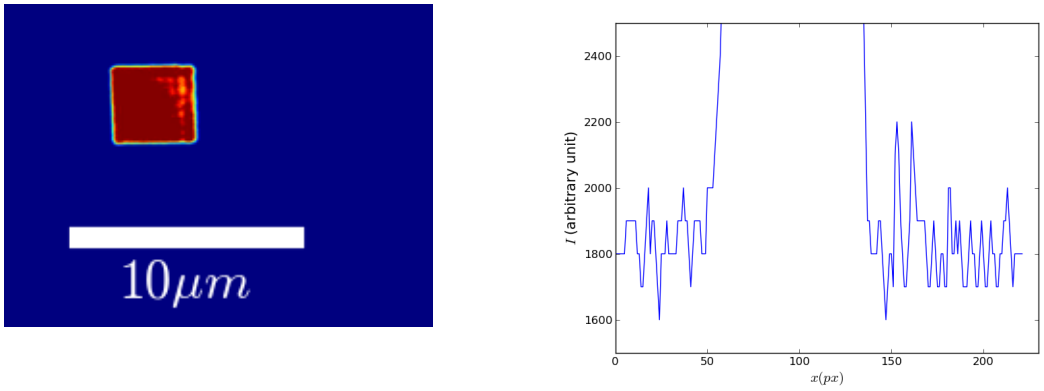
After ythat, we measured the number of pixels of the real image and find the size of the square opening:

$$\begin{aligned} a = & \ nb_{px-wat} * PX_{size-wat} \\ = & \ (4.93 \pm 0.21)\mu m \end{aligned} \tag{7}$$

With $nb_{px-wat} = 72$ is the nu,ber of pixel 3.5b.
To retrieve the size of a square opening from the Fourier image we use the Fourier optics. The Fourier transform of the square is given by figure 3.6a. The intensity distribution I of figure 3.6a is:

$$I(x,y) \propto sinC^2(\frac{\pi x}{\lambda f}a)sinC^2(\frac{\pi y}{\lambda f}a) \tag{8}$$

Where $f = 150mm$ is the focal of the lens, xand y are the coordinates of the point where is compute the intensity and a is the siye of the opening square.



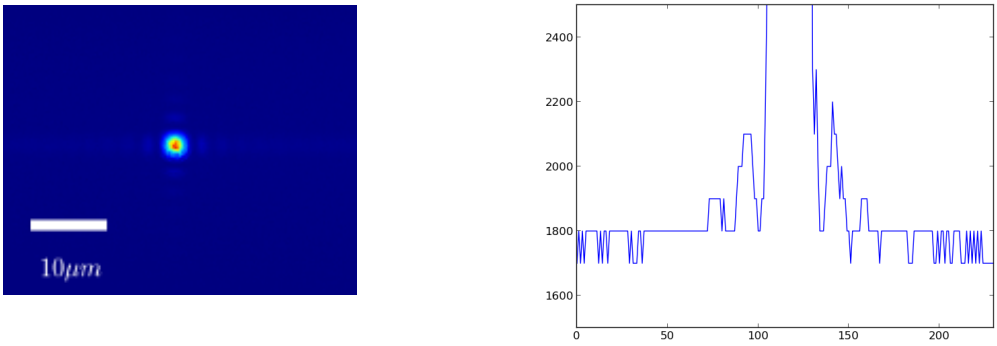
(a) (b)

Figure 3.5: Characterization of the real image of the square, (a) image from the Watec camera, (b) intensity profile

By taking the profile of figure 3.6a, one can measure the distance between two maxima d_p 3.6b:

$$d_p = nb_{px-emccd}PX_{size-emccd} \tag{9}$$

$$d_p = (237.336 \pm 5)\mu m \tag{10}$$



(a) (b)

Figure 3.6: Characterization of the Fourier image of the square, (a) image from the EMCCD, (b) 1D intensity profile of the fourier picture

$nb_{px-emccd} = 29.67$ and $PX_{size-emccd} = 8\mu m$. But if $x = d_p$, the intenity I is maximal:

$$\begin{aligned} \Longrightarrow sin(\frac{\pi d_p}{\lambda f}a) &= 1 \\ \Longleftrightarrow a &= \frac{\lambda f}{2d_p}.L_{factor} \\ a &= (5.21 \pm 0.452)\mu m \end{aligned} \tag{11}$$

Where $L_{factor} = \frac{164}{100*150}$ is a normalization factor and has to be used in order to return in the plane of the sample. $164mm$ is the focal of the required lens for the $100X$ objective, $150mm$ is the focal of the used lens and $100mm$ is the magnification of the objective.
By comparison of the sizes given by the real and the fourier images, one can conclude that the setup is correctly built since both value are found close to each other.

3.5 Characterization of the setup transmission

Before performing the measurements on the NWs, it is necessary to characterize the transmision of the setup at $532nm$. Ze measure the transmission in both arms of the setup:

- the Fourier imaging arm composed of:
  - oil-immersion objective,
  - L_1,
  - beam-splitter,
  - L_2

  the transmission of this arm is:

$$T_{fourier} = 0.374 \tag{12}$$

- The real imaging arm composed of:
  - oil-immersion objective,
  - L_1,
  - beam-splitter,
  - L_3

– L_4

the transmission of this arm is:

$$T_{real} = 0.339 \tag{13}$$

3.6 Principle of the EMCCD operation

As shown in the previous section, a EMCCD camera (Andor Ixon3 885) is used for this experiments. Owing to the built-in electron multiplying CCD, the camera possesses high sensitivity and gives a possibility to detect down to one single photon. Moreover, the camera allows to decrease the temperature of the CCD and thereby to improve considerably the signal to noise ratio by decreasing the amount of thermally excited electrons. The knowledge of acquisition parameter of the camera and high sensitivity makes it possible to find out the amount of the power of the incident light with a high accuracy (see program power.py, appendice A.2)
When incident onto a pixel of the CCD, a photon generates an electron with a certain probability called quantum efficiency. The electron signal is multiplied by a factor of an electron-multiplying gain and then digitized by an analog-to digital converter. The digitized data is presented in units called counts which correspond to a certain number of electrons which are captured by a pixel. The amount of electrons corresponding to a count is controlled by the setting named pre-amplifier gain parameter.
Thus, a pixel of an image gives information on the amount of photons captured by the corresponding pixel of the camera during a certain period of time. Knowing the parameters of the electron-multiplying gain, the pre-amplifier gain, the exposure time, the number of accumulated images and the quantum efficiency, one can retrieve the number of the incident photons from the counts of taken image:

$$nb_{inc-ph} = \frac{C_{tot}.G_{pre}}{G_{EM}.\eta_{eff}.t_{exp}.nb_{pic}} \tag{14}$$

Where,

- nb_{inc-ph} is the number of photons which fall on to the CCD.
- $C_{tot} = \sum_{ij}(C_{ij} - noise)$ is the total number of counts of the CCD, C_{ij} is the count level of the pixel ij and $noise$ is the noise calculated from a background picture.
- $G_{pre} = 1.34$ is the pre-amplification gain.
- G_{EM} is the electron multiplying gain parameter.
- $\eta_{eff} = 0.55$ is the quantum efficiency of the CCD for $\lambda = 532nm$.
- t_{exp} is the exposure time.
- nb_{pic} is the number of picture taken for the measurement.

3.7 Characterization of the Watec Camera

The Watec camera (Wat 120N+) is simpler than the EMCCD camera. But the quantum efficiency (η_{eff}) is not provided by the manual. The literature is checked and the quantum efficiency of around 50% is found for $\lambda \sim 500nm$. But, in order to check this value (and futhermore the programs and the coherence of the measurements) the quantum efficiency is measured experimentally with the laser diode at $\lambda = 532nm$.

$$nb_{inc-ph} = \frac{C_{tot}}{G\eta_{eff}t_{exp}} \Longleftrightarrow \eta_{eff} = \frac{C_{tot}}{nb_{inc-ph}Gt_{exp}} \tag{15}$$

Where,

- nb_{inc-ph} is the number of photons which fall on to the CCD.
- $C_{tot} = \sum_{ij}(C_{ij} - noise)$ is the total number of counts of the CCD, C_{ij} is the count level of pixel ij and $noise$ is the noise calculated from the background picture.
- G is the gain parameter, $G_{min} = 0dB$ and $G_{max} = 38dB$.
- η_{eff} is the quantum efficiency for $\lambda = 532nm$.
- t_{exp} is the exposure time.

In order to compute η_{eff}, the power of the input signal is measured. When the power is known, $nb_{orig-ph}$ can be found easily,

$$nb_{orig-ph} = \frac{P}{E_{ph}} \tag{16}$$

Where P is the power, and $E_{ph} = h\nu = 2.34eV$ is the energy of a photon. Then, nb_{inc-ph} is found by,

$$nb_{inc-ph} = nb_{orig-ph}.T_{tot}.T_{filter} \tag{17}$$

Where T_{tot} is given by equation 13, and $T_{filter} = 10^{-4}$ is the transmission of the filter placed before the camera in order to not saturate the CCD.
The following measurement is performed:

Table 2: η_{eff} of the Watec camera

$P(\mu W)$	η_{eff}
220 ± 2	0.451 ± 0.004
200 ± 2	0.481 ± 0.005
139 ± 2	0.664 ± 0.009

Therefore, in average the quantum efficiency of the Watec camera is:

$$\eta_{eff} = 0.533 \pm 0.006 \tag{18}$$

Which corresponds to the value found in the literature.

4 Experimental results

After checking the setup, measurements with several NWs are performed. First, the dependence of the power given by the NWs is studied. Then, the output signal is imaged first with a $100X$ objective of $NA = 0.75$ which is easy to adjust, and second with an oil-immersion objective to increase the numerical aperture up to $NA = 1.3$.

4.1 Power dependence

It is already shown[18] that the SH signal of the $LiNbO_3$ NWs follows the quadratic law (see section 2.1):

$$\mathbf{P}(2\omega) \quad = \quad 2\varepsilon_0\chi^{(2)}\mathbf{E}^2(\omega) \tag{19}$$

Since the power of a electromagnetic signal is propportionnal to the squared polarization density and to the square of the electric field;

$$Power(\omega) \propto \mathbf{E}^2(\omega) \tag{20}$$

$$\tag{21}$$

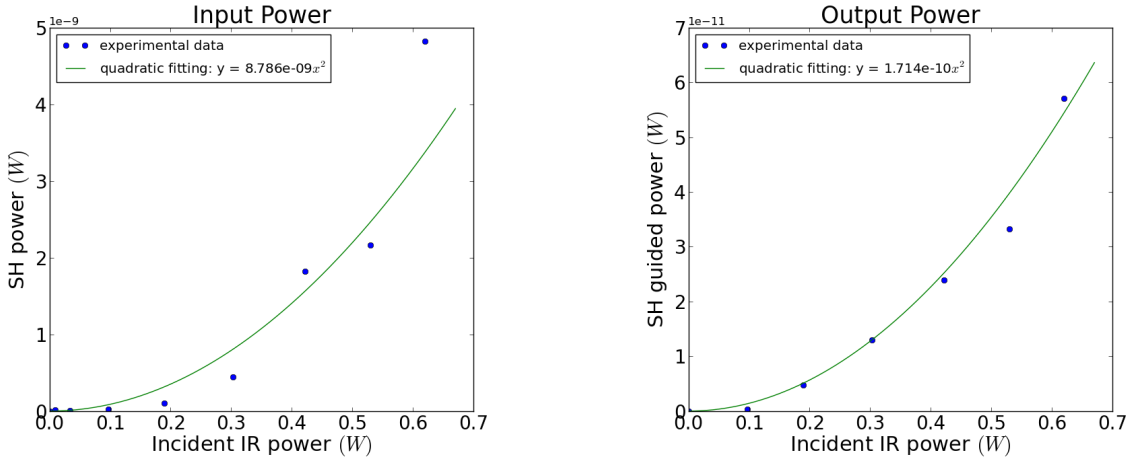
By injecting equation 20 inside 19, we can conclude that the power of the SH light is proportionnal to the square of power of the fundamental power;

$$Power(2\omega) \quad \propto \quad Power^2(\omega) \tag{22}$$

In this section, the power dependence of the SH signal is determined for both sides as a function of the power of the infrared input beam in order to show than we observe the SH signal.
The SH signal power emitted by the NW is given by,

$$P = nb_{orig-ph}E_{ph} \tag{23}$$

Where $E_{ph} = h\nu = 2.34eV$ is the energy of a photon, and $nb_{orig-ph} = \frac{nb_{inc-ph}}{T_{arm}}$ with nb_{inc-ph} is given by equation (14) for the EMCCD camera and by equation (15) for the Watec camera. Then, the transmission $T_{arm} = 0.374$ for the EMCCD and $T_{arm} = 0.339$ for the Watec camera (see section 3.5)
First, we check if the power of the input SH signal is in agreement with the quadratic law (22).



(a) (b)

Figure 4.1: SH power from the EMCCD, the experimental data are represented with dots while the fitting is represented by the green line, (a) from the input signal, (b) from the guided signal

From figure 4.1a the quadratic dependence can be clearly seen. The equation of the fitting is

$$y = 8.786 \times 10^{-9} x^2 \tag{24}$$

The power from the guided signal is also characterized 4.1b.
A quadratic shape is also observed,

$$y = 1.714 \times 10^{-10} x^2 \tag{25}$$

This results are in agreement with previous experiments which have already shown this shape for the guided signal [18], and confirm that it is really SH signal that it is observed on the cameras.

4.2 Fourier imaging of the SH signal

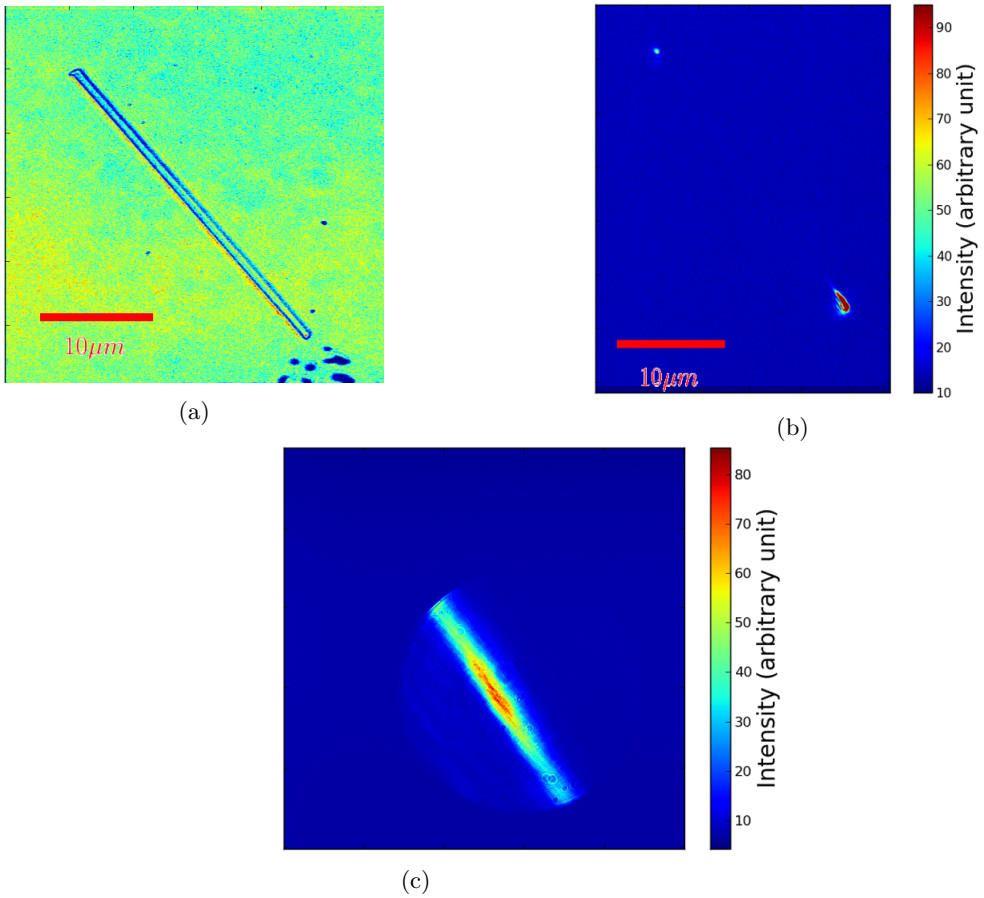
4.2.1 With $100X$ objective

Here are given some measurements performed with the $100X$ objective. The numerical aperture of this objective is ($NA = 0.75$) and it has a high working distance ($4mm$), which facilitates the measurements. The maximum θ angle θ_{max} is given in table 1: $\theta_{max} = 30°$ On figure 4.2, both signals (the guided and the non-guided signals) can be observed since the aperture is kept open. Figure 4.2b is the real image of the signal taken by the Watec camera. The figure illustrates the SH signals observed at the input and output of the NW. The origin of the guided signal is unknown, two cases are possible:

- all the SH signal is produced at the input and a part of this SH is guided inside the NW and gives the guided signal,
- a part of the IR signal is guided inside the NW and propagates SH signal which is observed at the second end of the NW.

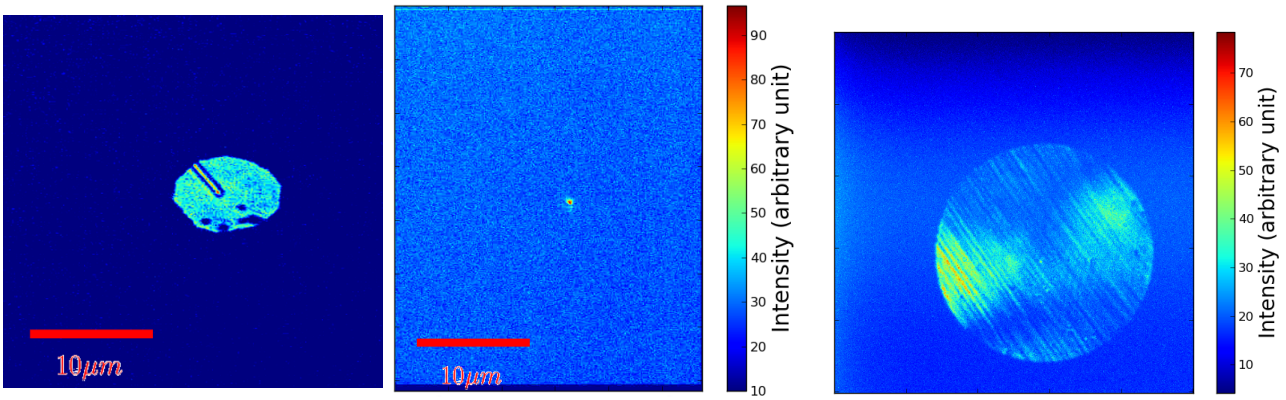
Figure 4.2 is not really useful to characterize the guided signal, as the input signal is stronger than the guided signal. sHowever, figure 4.2c helps to understand how the light is distributed for the input signal and figure 4.2b can be helpful for eliminating problems during the experiment (like defocusing).
Figure 4.2c is the picture used to characterize the SH signal. On this picture we can study the distribution of the intensity of the light as a function of the spherical angles θ and φ as explained in the section 3.3. The characterization of the input signal is not the task of this work, but it is interesting to study it too. It can be seen on figure 4.2c a line, which means the intensity is approximately zero for $\theta > \theta_{line}$ except for two specific φ angles.

Figure 4.3, is use in order to study the spectrum of the output, we use the aperture to block the light from the input. Figure 4.3c is used in order to characterize the output signal. It can be observed that the output signal is more broadened in comparison to the input. However, due to the small collection angle of the objective we detect only a small part of the guided SH signal. As a result, the distribution shape of the guided signal is not clear.



(a) (b) (c)

Figure 4.2: Signal given by the 100X objectif, (a) NW illuminated with white light, (b) SHG image from the Watec camera (real image) the input corresponds to the strong signal in the lower part of the figure, While the guided signal corresponds to the little point on the top of the figure, (c) SHG image from the EMCCD camera (Fourier transform of the generate SH)



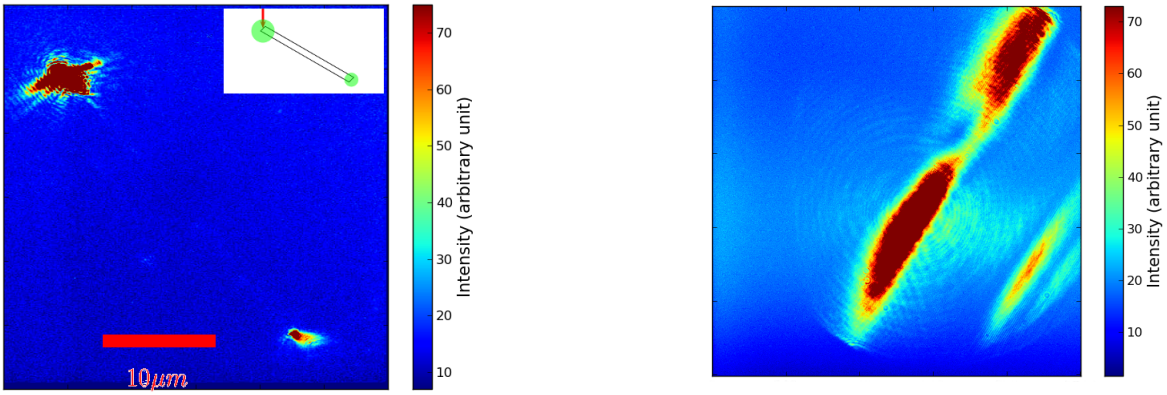
(a) (b) (c)

Figure 4.3: Guided signal given by the 100X objective ($NA = 0.75$), (a) NW illuminated with white light, (b) SH guided signal from the watec camera (real image), (c) SH guided signal from the EMCCD (fourier transformed image).

Therefore, an objective with a higher NA will is needed.

4.2.2 With $100X$ oil-immersion objective

For futher studies, a $100X$ oil-immersion objective with $NA = 1.3$ and $\theta_{max} = 58.9°$ (see table 1) is used. The θ_{max} is pratically the double of the $100X$ objective, therefore, more light is collected that will allow to obtaine more information about light distribution in order to characterize the signal.



(a) (b)

Figure 4.4: Input and guided SH signal given by the oil objective for the first NW, (a) from the Watec camera (real image), (b) from the EMCCD camera (fourier transformed image).

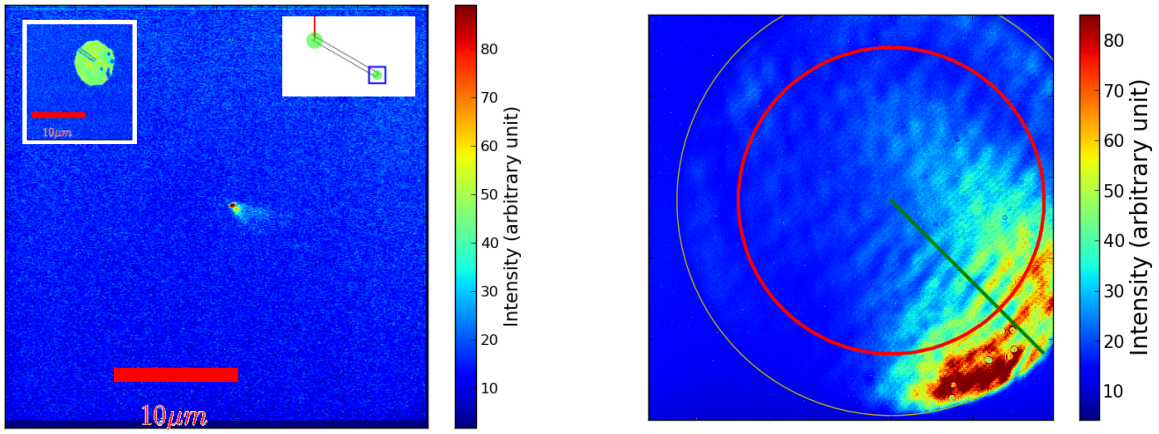
From figure 4.4, it is clear that the amount of the collected light is higher than for the previous $100X$ objective (The same NW is used as in figure 4.2 and figure 4.3). In figure 4.4a the shape of the guided signal can already be observed and seems to be a cone of light (Unlike to section 4.2.1 the guided signal is in the down part of the figure). When the fourier transform of the SH signal shown in figure 4.4a is performed, the line at the center is still observed, but there appears a second line for a $\theta_{sec} \sim 40°$, this line was not observed with the previous $100X$ objective since the angle θ_{sec} at which the second line is observed, is larger than the collection angle of the first $100X$ objective ($\theta_{max,100} \sim 30° < \theta_{sec}$) $\theta_{sec} > \theta_{max,100X}$. Therefore, the oil immersion objective allow to observe light distribution features which are not observed with the $100X$ objective.
Moreover, the central line in figure 4.4b is not identical to the line in figure 4.2c. In figure 4.4b two peaks can be observed at the line with the intensity drop in the center while in figure 4.2c only one peak is observed in the center. However, the same NW is used for both measurements but not illuminated at the same facet. Therefore, as the two facets have probably not the same shape, the distribution of the light at the input is dependent of the shape of the NW.
The $100X$ oil-immersion objective also allow to obtain more information on the distribution shape of the guided signal

Futhermore, the results are more interesting for the output signal, for which the shape appear clearly.

Figure 4.5 shows the real (4.5a) and the fourier (4.5b) images of the guided SH signal. On the real picture, the cone shape can be well observed: the light is diffracted. This cone shape appears to give some triangle figure in the fourier domain.
I look the intensity profile from the center of the spectrum to the edge of the spectrum. The radial profile is shown in figure 4.6a. One can see the intensity is lower in the center and increase gradually toward the edge of the spectrum for a φ angle of 315°. The circular profile is shown in figure 4.6b and one can see the signal is distributed in a cone of light between the angles 270° and 350°.

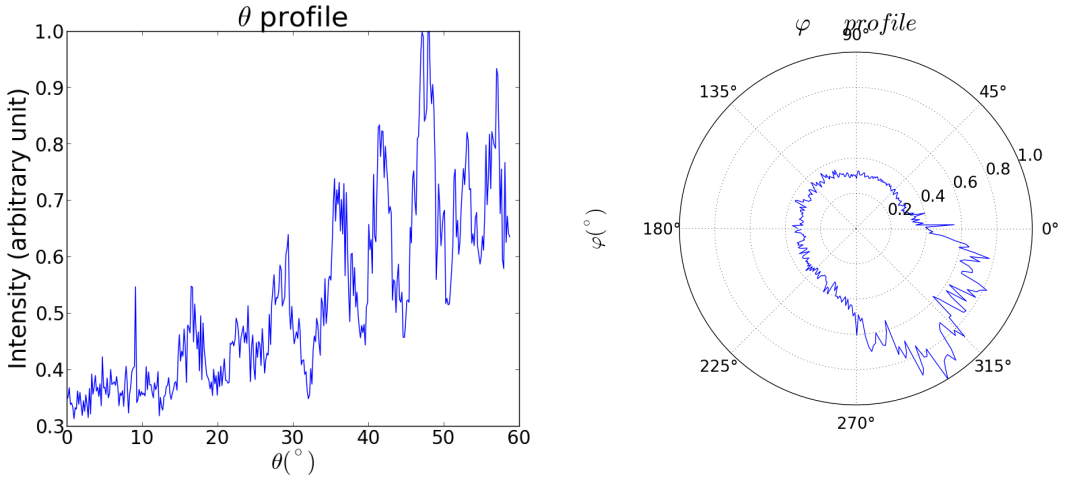
On figure 4.6a, the intensity profile appears clearly periodic with a period,

$$d_{peak} = 2.74\mu m \tag{26}$$



(a) (b)

Figure 4.5: SH guided signal given by the oil objective, (a) from the watec camera (real image) with white light picture in the top left side, (b) from the EMCCD camera (Fourier transformed image).



(a) (b)

Figure 4.6: θ and φ profiles for the first NW, (a) radial profile for $\varphi = 315°$, (b) circular profile for $\varphi = 40°$

The origine of this periodicity is still unknown and several possibilities are considered:

- the geometry of the NW,
- the size of the NW,
- noise during the measurements...

Measurement with another NW is performed in figure 4.7.

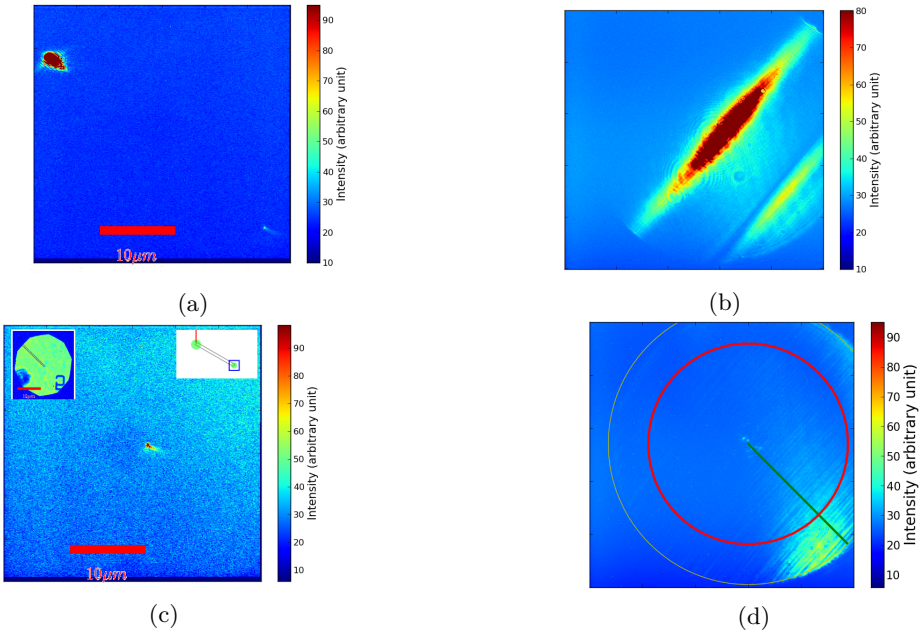


Figure 4.7: Input and guided SH signal given by the oil objective for a second NW, (a) input from the Watec camera, (b) input from the EMCCD camera, (c) guided from the Watec camera, on the top left side of the image a white light picture of the nanowire can be observed, (d) guided from the EMCCD camera

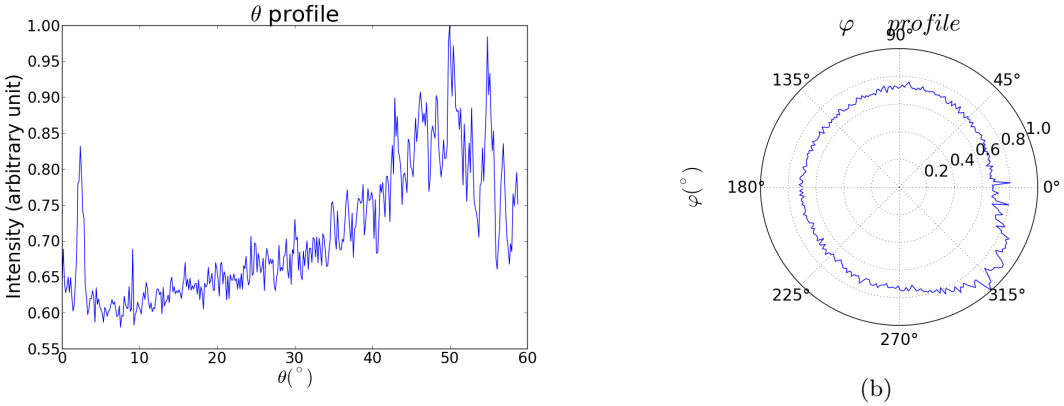


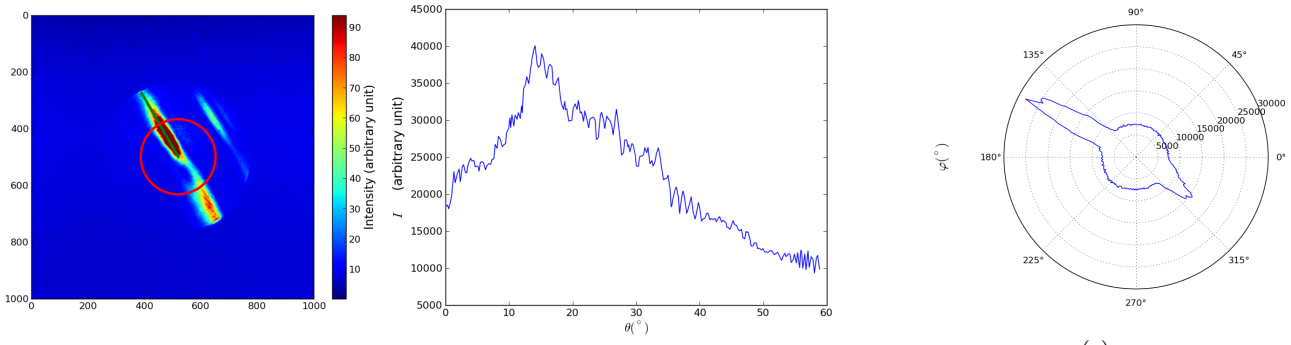
Figure 4.8: Radial and circular profiles for a second NW, (a)radial profile for $\varphi = 315°$, (b)circular profile for $\theta = 42°$

The similar profiles are observed than for the other NW even if the intensity of the signal is weaker. On figure 4.8b it can be observed that the distance between the peaks is smaller than for the previous NW.

$$d_{peak} \quad = \quad 0.687\mu m \tag{27}$$

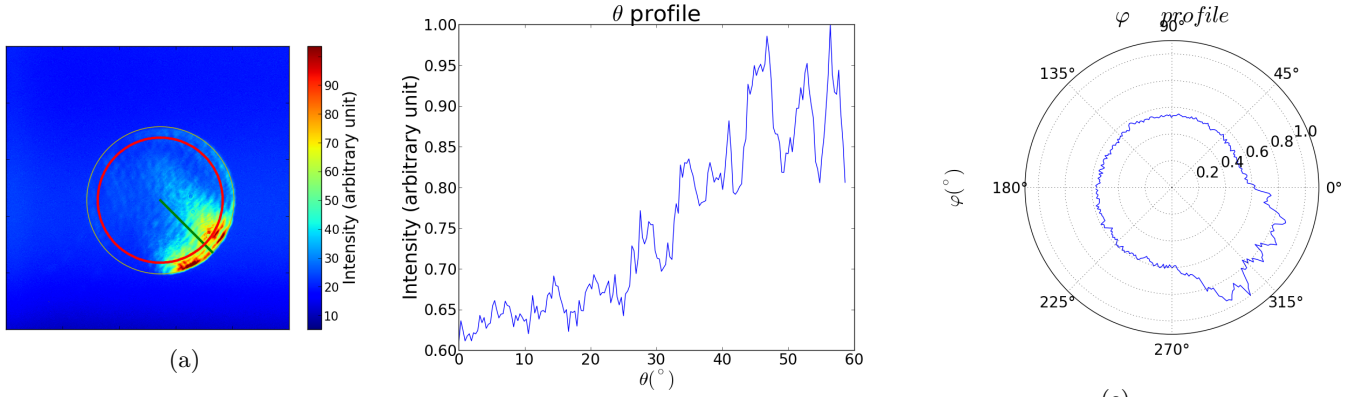
One origin of the difference between the period observed in the radial profiles for the the two NWs could be caused by the size of the aperture. For the first NW, the aperture size was very small, unlike to the second NW where the size of the aperture is bigger (compared the topleft insets in figures 4.5a and 4.7c). Therefore, for the first NW, the aperture acts like a circular mask in the Fourier plane which gives a low pass filter. This could explain the oscillation difference in the radial profile of the SH guided signal for both NWs. The problem in this figures is that all the NA circle is not visible, therefore the lens L_2 is

changed and a lens with a focal of $15cm$ is used.
The radial and circular profiles for the SH signal at the input and guided signal are shown in figures 4.9 and 4.10



(a) (b) (c)

Figure 4.9: SH input and output intensity distribution for $f_{L_2} = 15cm$, (a) in the fourier plane, (b) θ profile for $\varphi = 120°$, (c) φ profile for $\theta = 20°$



(a) (b) (c)

Figure 4.10: SH output intensity distribution for $f_{L_2} = 15cm$, (a) in the fourier plane, (b) θ profile for $\varphi = 215°$, (c) φ profile for $\theta = 55°$

The problem with this new focal is that the picture size is drastically reduced, therefore, the study is less easier, but the information are still in the picture and it is sure that all the NA circle is observed.

4.3 Discussion of the experimental results

What can be deduced from the measurements in the fourier plane? For the guided signal it is clear that the intensity of the light increases when θ increases for certain value of φ, this shape can be observed on figures 4.7d, 4.10a... From the circular profiles of one of the NW 4.6b the illumination angle in the plane perpendicular to the objective normal is in the range of $\Delta\varphi = 70^{circ}$. Therefore, the light distribution in the fourier plane looks like a triangle with a vertex in the center of the NA circle and an angle at the vertex of $\Delta\varphi$.
The interpretation of this distribution is as follows; the φ distribution means that the light is directed

and diffracted in only one direction and the comparison with the direct pictures (figures 4.5a 4.7c), allow to determined the direction of the light which is in the same direction than the NW. The extension in φ means that the beam is not a line but possess some width and this width is increasing with θ. The fact that the light intensity increases with θ means that more photons propagate in the points out direction. iLight propagates in the NW direction and less photons deviate from this direction 4.11.

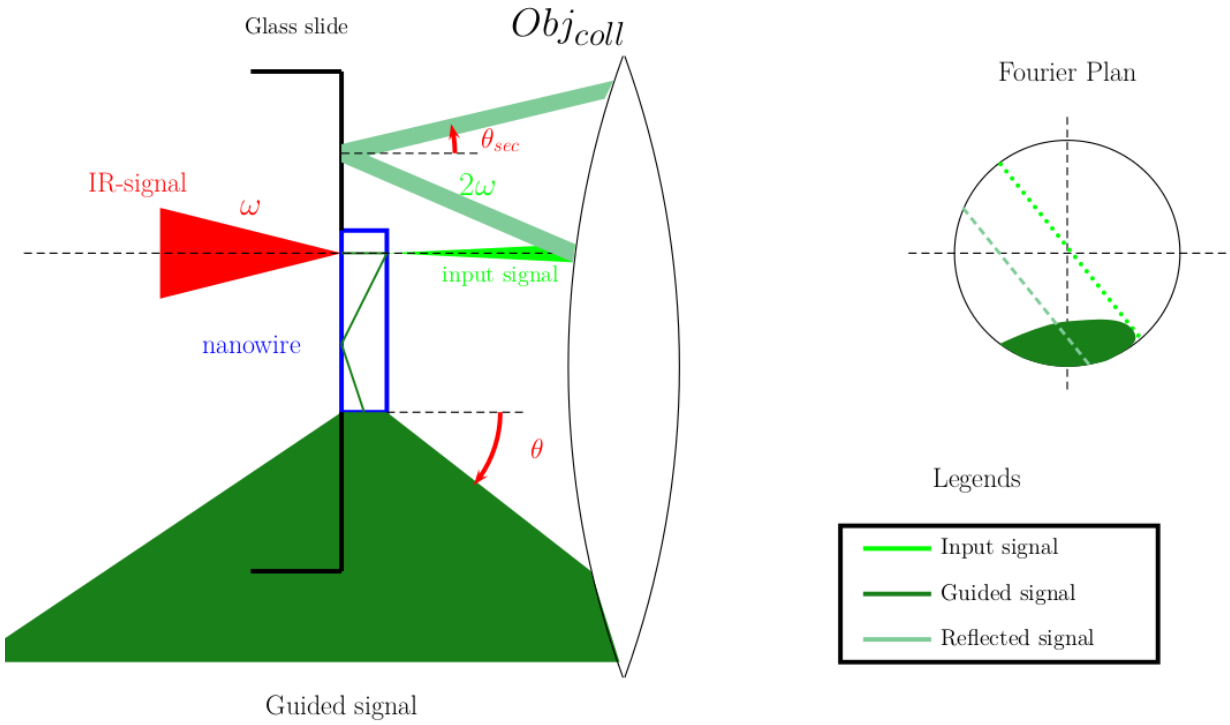


Figure 4.11: Diagram of the light propagation with the different origin of the SH signal.

The Fourier picture of the input signal is easier to understand since the light arrives straight to the objective, in the contrary to the guided signal. The first thing to explain is the observed lines. This line means that the photons are extended in space for two directions, one interpretation could be that the IR beam possesses a width in the focal plane of the focussing objective. Therefore, the SH signal is not like a point source but has some extension along the NW 4.12.

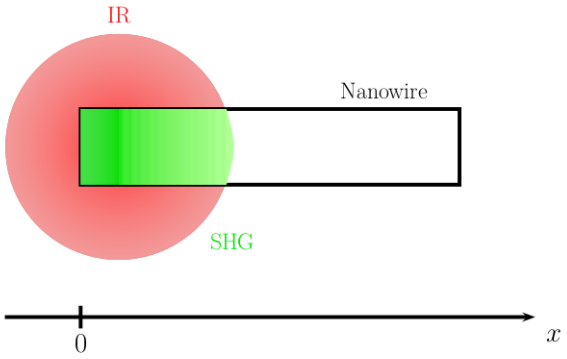
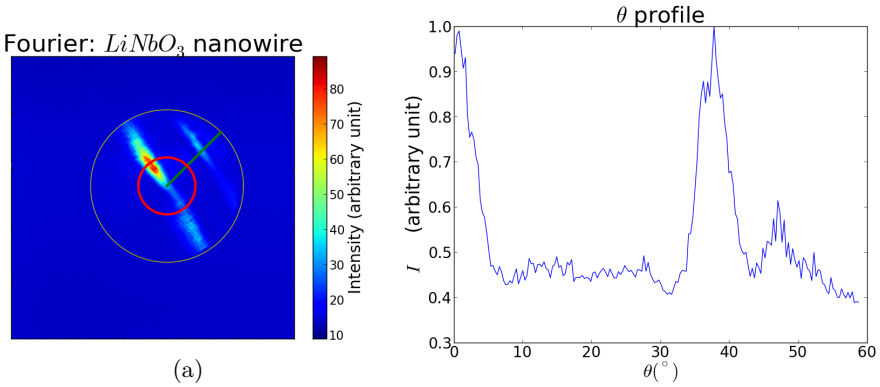


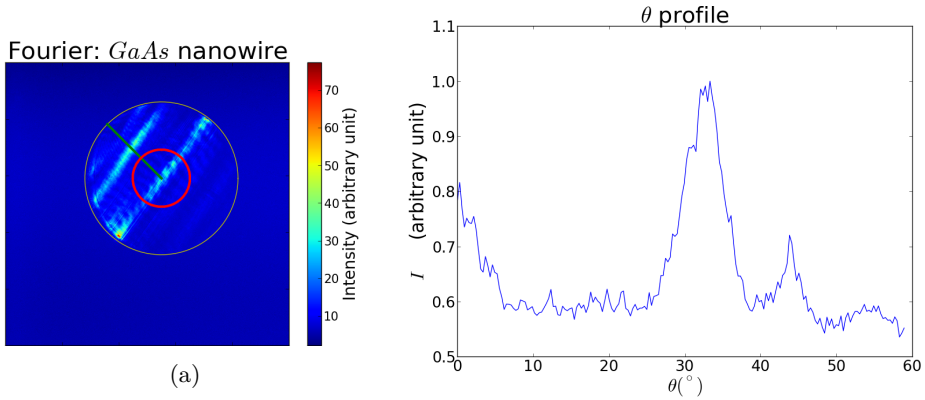
Figure 4.12: SHG with a Gaussian Beam along the NW, the signal is weaker at the extremity of the IR beam than at the center

But this interpretation can only explain the line in the center of the Fourier plane. But as it can be observed on figure 4.9a, the second line appears in the Fourier plane for the input signal and this line appears for $LiNbO_3$ NW (figure 4.13). For check if this second line is caused by the shape of the nanowire, measurement with $GaAs$ NW which possessed a cylindrical form (contrary to the $LiNbO_3$ which possesses a Parallelepiped form) and also for $GaAs$ NW (figure 4.14) the second line appears.



(a) (b)

Figure 4.13: Observation of the second line at $\theta \sim 40^\circ$ for $LiNbO_3$ NW, (a) in the Fourier plan, (b) θ profile for $\varphi = 45^\circ$



(a) (b)

Figure 4.14: Observation of the second line at $\theta \sim 35^\circ$ for $GaAs$ NW, (a) in the Fourier plan, (b) θ profile for $\varphi = 135^\circ$

The first phenomenum which can explain the presence of this line in the Fourier plane would be some reflection of the SH input signal on the objective then reflected again on the glass sample, afterward directed on the objective with some shift angle θ_{sec} as presented on figure 4.11.
Another explanation suggests that the second line is caused the Gaussian shape of the IR incident beam the SH generation is not homogeneous but stronger at the center of the IR signal and weaker on the limit of the IR signal as presented in figure 4.12. Therefore, the SH signal can be seen as a convolution product of a windows function $\Pi(x)$ defined by equation (29) and a gaussian function:

$$I(x) \quad = \quad \Pi(x) \star e^{-x^2} \tag{28}$$

With $I(x)$ the intensity distribution along the NW.

$$\Pi(x) \quad = \quad \begin{cases} 1 & \text{if} \quad x \geq 0 \\ 0 & \text{elsewhere} \end{cases} \tag{29}$$

Where x is the axis along the NW.
Therefore, the Fourier transform of the intensity is given by:

$$\tilde{I}(\alpha) = \begin{cases} e^{-\alpha^2} & \text{if} \quad \alpha < 0 \\ sinc^2(\alpha) & \text{if} \quad \alpha > 0 \end{cases} \tag{30}$$

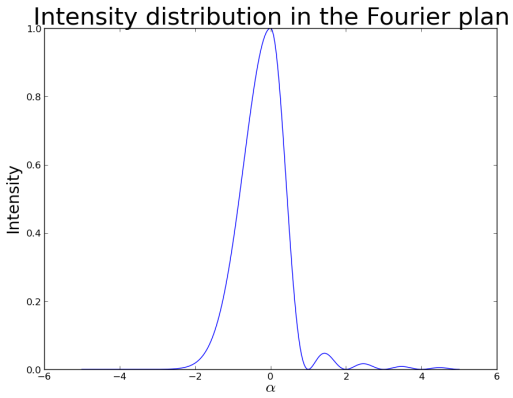


Figure 4.15: Intensity distribution in the Fouier plane. The second line corresponds to the second maxima.

The intensity distribution in the Fourier plan is shown in figure 4.15. The principal pic correspond to the central line on figure 4.13a while the second maximum correspond to the second line.

5 Conclusion

During this internship, the shape of the guided SH signal of the $LiNbO_3$ NWs has been characterized in order to understand how the light is distributed and if $LiNbO_3$ NWs can be used in order to perform localized imaging. The study of the light distributions as a function of the spherical angles θ and φ shows that the beam given by the end of the NW has a similar shape as a torch light with a cone of light. The intensity decreases when the distance from the NW increases but it shines a larger area. Therefore, the SH guided signal of $LiNbO_3$ can be used for localized imaging since it can illuminated area with sufficient intensity. Moreover, several points should be studied, in particular the influence of the shape of the end of the NW, it has been shown previously that the shape of the NW can change the light distribution at the input. Therefore, it would be probably the case with the output. By cutting the NW, one can change the shape at the end, and can modify the output signal. This point should be studied because it would be probably possible to obtain a better shape of the light for cell imaging, in particular it would be possible to change the light direction (figure 5.1).

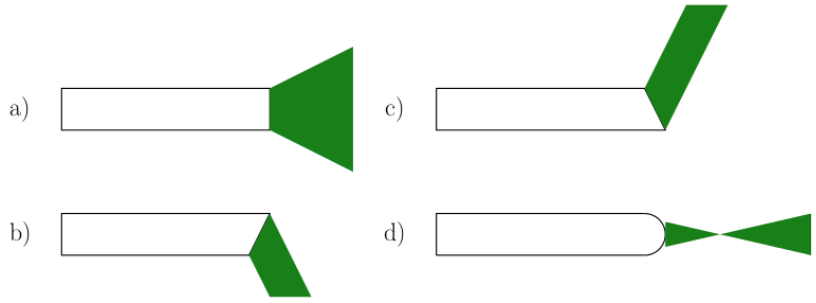
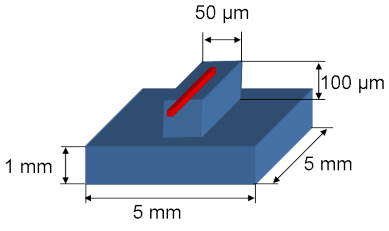
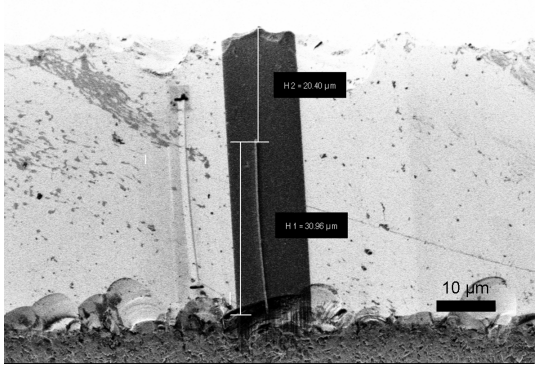


Figure 5.1: Hypothetical guided signal shape for several shape of the output end of the NWs, (a) straight facet, (b) cut facet to the down, (c) cut facet to the top, (d) curved facet

Now that the shape of the guided signal is known at the output, the power of this guided signal should be characterized. Since the actual setup allows to collect, only a small part of the guided signal, it was not possible to do measurement of the power of the guided signal. Therefore, the setup has to be modified, in particular instead to put the NW on a glass slide, the NW should be put on a stage as shown in figure 5.2. With this setup it will be possible to do measurement of the guided signal power and also to observe totaly the guided signal.



(a)



(b)

Figure 5.2: NW on a stage, (a) diagram, (b) picture

References

[1] Comission recomendation of 18 october 2011 on the definition of nanomaterial. *Official Journal of the European Union*, (54):(L275):38–40, 2011.

[2] M. H. Huang, S. Mao, H. Feick, H. Yan, Y. Wu, H. Kind, E. Weber, R. Russo, and P. Yang. Room-temperature ultraviolet nanowire nanolasers. *Science (New York, N.Y.)*, 292(5523):1897–9, June 2001.

[3] Matt Law, Lori E Greene, Justin C Johnson, Richard Saykally, and Peidong Yang. Nanowire dye-sensitized solar cells. *Nature materials*, 4(6):455–9, June 2005.

[4] Ruoxue Yan, Ji-Ho Park, Yeonho Choi, Chul-Joon Heo, Seung-Man Yang, Luke P Lee, and Peidong Yang. Nanowire-based single-cell endoscopy. *Nature Nanotechnology*, 7(December):191–196, March 2011.

[5] R. Yan P. Yang and M. Fardy. Semiconductor nanowire: what's next? *Nano Lett*, 10:1529–1536, 2010.

[6] F F Jöbsis. Noninvasive, infrared monitoring of cerebral and myocardial oxygen sufficiency and circulatory parameters. *Science (New York, N.Y.)*, 198(4323):1264–7, December 1977.

[7] F Dutto, C Raillon, K Schenk, and a Radenovic. Nonlinear optical response in single alkaline niobate nanowires. *Nano Letters*, 11(6):2517–2521, 2011.

[8] R. Grange, J. W. Choi, C. L. Hsieh, Y. Pu, A. Magrez, R. Smajda, L. Forro, and D. Psaltis. Lithium niobate nanowires synthesis, optical properties, and manipulation. *Applied Physics Letters*, 95(14):143105, 2009.

[9] Yuri Nakayama, Peter J Pauzauskie, Aleksandra Radenovic, Robert M Onorato, Richard J Saykally, Jan Liphardt, and Peidong Yang. Tunable nanowire nonlinear optical probe. *Nature*, 447(7148):1098–1101, July 2007.

[10] Davide Staedler, Thibaud Magouroux, Rachid Hadji, Cécile Joulaud, Jérôme Extermann, Sebastian Schwung, Solène Passemard, Christelle Kasparian, Gareth Clarke, Mathias Gerrmann, Ronan Le Dantec, Yannick Mugnier, Daniel Rytz, Daniel Ciepielewski, Christine Galez, Sandrine Gerber-Lemaire, Lucienne Juillerat-Jeanneret, Luigi Bonacina, and Jean-Pierre Wolf. Harmonic nanocrystals for biolabeling: a survey of optical properties and biocompatibility. *ACS nano*, 6(3):2542–9, March 2012.

[11] A. Sergeyev, Reinhard G., A. Solntsev, F. Schrempel, Ernst-Bernhard Kley, T. Pertsch, and R. Grange. Second-harmonic generation in lithium niobate nanowires for local fluorescence excitation. *Nano Letters*, 76(1990):1356–1360, 2012.

[12] T. Shegai, V. D. Miljkovi, K. Bao, H. Xu, P. Nordlander, and P. Johansson. Unidirectional broadband light emission from supported plasmonic nanowires. *Nano Letters*, pages 706–711, 2011.

[13] Rachel Grange, Thomas Lanvin, Chia-Lung Hsieh, Ye Pu, and Demetri Psaltis. Imaging with second-harmonic radiation probes in living tissue. *Biomedical optics express*, 2(9):2532–9, September 2011.

[14] Davide Staedler, Thibaud Magouroux, Marc Dubled, Daniel Rytz, Sandrine Gerber-lemaire, Luigi Bonacina, and Jean-pierre Wolf. Frequency Doubling Nanocrystals for Cancer Theranostics. pages 1–16.

[15] R..W. Boyd. *Nonlinear Optics, Third Edition*. Academic press: Burlington, Mass, 2008.

[16] Jerry Dadap. Optical second-harmonic scattering from cylindrical particles. *Physical Review B*, 78(20):205322, November 2008.

[17] M. Saleh B. E. A., Teich. *Fundamentals of Photonics*. Wiley & Sons, Inc., New York, USA, 2007.

[18] A. Sergeyev. Second-harmonic waveguiding and resonance modes in oxide and semiconductor nanomaterials. *Abbe School of Photonics*, 2012.

Appendices

A Code sources

A.1 Executables

Files used in order to execute some action (plot graphics...)

prog.py

```
#executable in order to plot all the picture
#from the real camera
#must be used in the folder real/ which
#don't contains sif files
#wrote by marc barbry
#01/07/2013

from plot import *
import os
from sys import exit

fichier = os.listdir(os.getcwd())

#create the folder for the results
try:
  os.mkdir("result-pic")
except OSError:
  pass

pic = list()
picture = list()
folder, picture = list_folder(fichier)



klass = list_pic(folder, "result")
pic = klass.pict
result = list()
trans = list()
pix = [pixel_size_R, pix_mat]

for i in range(0, len(pic)):
  print(pic[i])
  result.append(extract(pic[i]))
  name = pic[i][0:len(pic[i])-4]
  graph_2D(result[i].data, name)
  trans.append("result-pic/" + name + "_2D.png")
  scale_bar(pic[i], pix[0])

for i in range(0, len(trans)):
  print(trans[i])
  scale_bar(trans[i], pix[1])
```

don.py

```
#program which plot the data from the sif camera
#must be used in the folders sif-files/ after convertion
#of the sif files in don files with
#the program sif_convert.m
#note: the don files are a kind of files
#create for this internship
#in order to be able to work the data from
#the EMCCD camera with python
#since the sif files can not be read with python.
#wrote by marc barbry
#01/07/2013

from plot import *
import os
from sys import exit

fichier = os.listdir(os.getcwd())

#create the folder for the results
try:
  os.mkdir("result-pic")
except OSError:
  pass

pic = list()
picture = list()
folder, picture = list_folder(fichier)


klass = list_pic(folder, "result")
pic = klass.pict
result = list()
trans = list()


for i in range(0, len(pic)):
  data = np.loadtxt(pic[i])
  print(pic[i])
  name = pic[i][0:len(pic[i])-4]
  graph_2D(data, name)
```

sif_convert.m

```
%program which convert the sif files to don files
%need to be in the folder sif-files/
%enter the patch of the folder which contains the
%sif files that you want convert
%wrote by marc barbry
%01/07/2013

function [] = sif_convert(name)
D = dir(name);
liste = {D(:).name};
len_liste = size(liste);
name = strcat(name, '/');

for k=1:len_liste(2)
    liste(k) = strcat(name, liste(k));
end

len_liste = size(liste);
```

```
for k=3:len_liste(2)
    chaine = liste(k)
    data_tot = sif_reader(chaine);
    data = data_tot.imageData;
    don = '.don';
    chaine = strcat(chaine, don);
    save(chaine{1}, 'data');

    name_info = strcat(chaine{1}, '_info.dat');
    fid = fopen(name_info,'w');

    fprintf(fid, '#Synopsis:\n');
    fprintf(fid, '#data=sifread(file)\n');
    fprintf(fid, '#file which contains all the...
    information abour the measurement:\n');
    fprintf(fid, '#file: %s\n', chaine{1});
    fprintf(fid, '#\n');
    fprintf(fid, '#temperature CCD temperature [C]\n');
    fprintf(fid,'%f\n', data_tot.temperature);
    fprintf(fid,'#.delayExpPeriod delay[s]\n');
    fprintf(fid, '%f\n', data_tot.delayExpPeriod);
    fprintf(fid, '#.exposureTime  Exposure time [s]\n');
    fprintf(fid, '%f\n', data_tot.exposureTime);
    fprintf(fid, '#.accumulateCycles  Number of accumulation cycles\n')
    fprintf(fid, '%f\n', data_tot.accumulateCycles);
    fprintf(fid, '#.accumulateCycleTime...
    Time per accumulated image [s]\n');
    fprintf(fid, '%f\n', data_tot.accumulateCycleTime);
    fprintf(fid, '#.stackCycleTime Interval in image series [s]\n');
    fprintf(fid, '%f\n', data_tot.stackCycleTime);
    fprintf(fid, '#.pixelReadoutTime Time per pixel readout [s]\n');
    fprintf(fid, '%f\n', data_tot.pixelReadoutTime);
    fprintf(fid, '#.pre_amp_gain  Pre amplification gain of the CCD\n')
    fprintf(fid, '%f\n', 1.34);
    fprintf(fid, '#.gainDAC  EM gain of the camera\n');
    fprintf(fid, '%f\n', data_tot.gainDAC);

    fclose(fid);
end
end
```

A.2 Functions

Files which contains all the functions used during the internship.

sif_reader.m

```
%SIFREADNK Read SIF multi-channel image file.
%
% This function reads a sif file taken with
% Andor Solis software on our Shamrock spectrograph
% and Andor Newton CCD camera. With each new
% iteration of andor's solis software, they make a new
% format for the sif file. This one works for us,
% but small modifications may ben necessary for
% other Andor cameras.
%
% The main improvement of this function from Marcel
% Leutenegger's is to read the non-linear corrections
% to the wavelength axis.
%
% This function uses the index of refraction of air
% to convert between wavelength and energy. This can be
% seen in the help section for convertUnits.
%
% This function can read a kinetic data file.



% modified by Todd Karin and Kai-Mei Fu
% to read andor's wavelength data.
%
% Todd Karin
% 06/02/2012



% Version History
%
% 2/17/2012
% modified so that program can read kinetic
%data files as well.
%
% 2/11/2012
% Added in an energy axis in units of eV
%
% 2/8/2012
% Modified code to add non-linear corrections
% to wavelength axis.
%
% 1/22/2012
% Program works with data taken on FVB mode.
% But there are some issues remaining to be solved.
% Some of the data files do not contain the line
% that contains the wavelength calibration Susupect
% this has something to do with multitrack vs. FVB modes.
%
% 1/21/2012
% Attempted to modify code to be insensitive to whether
% data was taken on %multi-track or full vertical
% binning modes. I changed imagesc to plot.
% Might have trouble now with multi-track mode.
%
% 07/13/2012
% Vectorized the creation of the wavelength, energy and
% frequency axes for speed. Optimized a few
% other parts of the code.



% modified significantly to do Andor iXon kinetic series
% by Kai-Mei Fu Nov 2011

%WARNINGS
%version is for kinetic series externally triggered
%if you are not in external then exposureTime is stored
%in delayExpPeriod program cannot currently accomodate
%a background or reference image

%Synopsis:
%
%  data=sifread(file)
%     Read the image data from file.
%     Return the image data in the following structure.
%
%  .temperature            CCD temperature [C]
%  .delayExpPeriod         delay after trigger [s]
%  .exposureTime           Exposure time [s]
%  .cycleTime              Time per full image take [s]
%  .accumulateCycles       Number of accumulation cycles
%  .accumulateCycleTime    Time per accumulated image [s]
%  .stackCycleTime         Interval in image series [s]
%  .pixelReadoutTime       Time per pixel readout [s]
```

```
%  .detectorType           CCD type
%  .detectorSize           Number of read CCD pixels [x,y]
%  .fileName               Original file name
%  .shutterTime            Time to open/close the shutter
%  .frameAxis              Axis unit of CCD frame
%  .dataType               Type of image data
%  .imageAxis              Axis unit of image
%  .imageArea              Image limits [x1,y1,first image;
%                                        x2,y2,last image]
%  .frameArea              Frame limits [x1,y1;x2,y2]
%  .frameBins              Binned pixels [x,y]
%  .timeStamp              Time stamp in image series
%  .imageData              Image data (x,y,t)
% where t is the image kinetic series

%Note:
%
%  The file format was reverse engineered by identifying
% known information within the corresponding file. There
% are still non-identified regions left over but the
% current summary is available on request.
%

%        Marcel Leutenegger  November 2006


function data =sifreadnk(file)
f=fopen(file,'r');
if f < 0
   error('Could not open the file.');
end
if ~isequal(fgetl(f),'Andor Technology Multi-Channel File')
   fclose(f);
   error('Not an Andor SIF image file.');
end
skipLines(f,1);
data=readSection(f);
fclose(f);


%Read a file section.
%
% f      File handle
% info   Section data
% next   Flags if another section is available
%
function info=readSection(f)
o=fscanf(f,'%d',6); %% scan over the 6 Bytes
info.temperature=o(6); %o(6)
skipBytes(f,10);%% skip the space (why 10 not 11?)
%skipLines(f,1);
%info.whatisthis=readLine(f)
o=fscanf(f,'%f',5);%% Scan the next 5 bytes
info.delayExpPeriod=o(2);
info.exposureTime=o(3);
info.accumulateCycles=o(5);
info.accumulateCycleTime=o(4);
skipBytes(f,2); %% skip 2 more bytes
o=fscanf(f,'%f',2);
info.stackCycleTime=o(1);
info.pixelReadoutTime=o(2);
o=fscanf(f,'%d',3);
info.gainDAC=o(3);
skipLines(f,1);
info.detectorType=readLine(f);
%skipLines(f,1);
%info.whatisthis=readLine(f)
info.detectorSize=fscanf(f,'%d',[1 2]);
info.fileName=readString(f);
skipLines(f,4);
% Added the following to extract the center wavelength
% and grating
o=fscanf(f,'%f',8);
info.centerWavelength = o(4);
info.grating = round(o(7));

skipLines(f,10); % added this in
o=fscanf(f,'%f',4);
info.minWavelength = o(1);
info.stepWavelength = o(2);
info.step1Wavelength = o(3);
info.step2Wavelength = o(4);
info.maxWavelength = info.minWavelength...
+ info.detectorSize(1)*info.stepWavelength;

% Create wavelength, energy and frequency axes.
da = 1:(info.detectorSize(1));
info.axisWavelength = info.minWavelength +...
da.*(info.stepWavelength +...
da.*info.step1Wavelength + da.^2*info.step2Wavelength);

skipLines(f,6);

info.frameAxis=readString(f); %'Pixel number'
info.dataType=readString(f);  %'Counts'
info.imageAxis=readString(f);  %'Pixel number'
o=fscanf(f,'65541 %d %d %d %d %d %d %d %d 65538...
%d %d %d %d %d %d',14); %% 14 is lines in o?
temp = o;
info.imageArea=[o(1) o(4) o(6);o(3) o(2) o(5)];
info.frameArea=[o(9) o(12);o(11) o(10)];
info.frameBins=[o(14) o(13)];
s=(1 + diff(info.frameArea))./info.frameBins;
z=1 + diff(info.imageArea(5:6));

info.kineticLength = o(5);
if prod(s) ~= o(8) || o(8)*z ~= o(7);
   fclose(f);
   error('Inconsistent image header.');
end

skipLines(f,2+info.kineticLength);

info.imageData = reshape(fread(f,prod(s)*z,...
'single=>single'),[s z]);

o=readString(f);
if numel(o)
   fprintf('%s\n',o);
end




%Read a character string.
%
% f      File handle
% o      String
%
function o=readString(f)
n=fscanf(f,'%d',1);
if isempty(n) || n < 0 || isequal(fgetl(f),-1)
   fclose(f);
   error('Inconsistent string.');
end
o=fread(f,[1 n],'uint8=>char');


%Read a line.
%
% f      File handle
```

```
% o      Read line
%
function o=readLine(f)
o=fgetl(f);
if isequal(o,-1)
   fclose(f);
   error('Inconsistent image header.');
end
o=deblank(o);


%Skip bytes.
%
% f      File handle
% N      Number of bytes to skip
%
function skipBytes(f,N)
[ret,n]=fread(f,N,'uint8');
if n < N
   fclose(f);
   error('Inconsistent image header.');
end


%Skip lines.
%
% f      File handle
% N      Number of lines to skip
%
function skipLines(f,N)
for n=1:N
   if isequal(fgetl(f),-1)
      fclose(f);
      error('Inconsistent image header.');
   end
end
```

plot.py

```
#kernel of the other programs
#contains all the basic functions used
#during this internship
#wrote by marc barbry
#01/07/2013

from __future__ import division
import os
import Image
import ImageDraw as ID
from pylab import *
import numpy as np
import mayavi.mlab as mlab
import matplotlib.pyplot as plt


########################
#                      #
#     Constantes       #
#                      #
########################

#quantum efficiency of the EMCCD
global eta_eff
eta_eff = 0.55

#planck cte
global h
h = 6.64*10**-34

#light frequency
global nu
nu = 3*10**8/(532*10**-9)

#setup transmission
global transmission
transmission = 0.374

#pixel size for the picture from the camera (_R) and
#for the picture calculate by matplotlib (_mat)
#thorlab camera
#pixel_size_R = 0.049079754 #micrometer
#pix_mat = 0.117647058824
#konig camera
global pixel_size_R
pixel_size_R = 0.06802721088435375 #micrometer

global pixel_mat
pix_mat = 0.07142857142857142


class extract:
  """
  class which extract the data of pictures from camera
  take in input the name of the picture
  return the array corresponding to the picture
  contains the function extract
  Example of call:
    image = extract(image_name)
  """
  def __init__(self, name):
    self.name = name
    im = Image.open(self.name).convert('LA')
    self.pix = im.load()
    self.maxx = im.size[0]
    self.maxy = im.size[1]
    self.data = self.extract_data()

  def extract_data(self):
    data = arange(self.maxx*self.maxy).\\
    reshape(self.maxx,self.maxy)
    for x in range(self.maxx):
      for y in range(self.maxy):
        data[x, y] = self.pix[x, y][0]

    return data

class profile_theta:
  """
  class used for plot the profile of the spectrum images
  as function of the angle theta, adjusted for the
  oil-objectif with NA = 1.3 and the refractive index
  of the oil n =1.518
  should be call from the function graph_2D_prof for a optimal use
  contains the function:
    graph_1D
  """

  def __init__(self, data, name, x0, y0, r, phi):
    #input variables
    self.name = name
    self.data = data

    #interne variables
    self.x = list() #abscisse
    self.y = list()
    self.xp = list()
```

```
    self.maxi = len(self.data[0, :])
    middle = int(self.maxi/2)

    self.xmin = int(x0-r*cos(phi))
    self.ymin = int(y0-r*sin(phi))
    self.xmax = int(x0+r*cos(phi))
    self.ymax = int(y0+r*sin(phi))

    f_x = 1
    f_y = -1
    dx = x0
    dy = y0
    dx_r = x0
    dy_r = y0
    dr = 1
    radius = 0


    #general loop
    while (radius<r):
      self.x.append(int(dy))
      self.xp.append(int(dx))
      self.y.append(data[int(dy), int(dx)])
      radius = radius + dr

      dx = dx + f_x*dr*cos(phi)
      dy = dy +f_y*dr*sin(phi)

      #Boundaries conditions
      if (int(dx) == len(data[0, :]-1)) or int(dx) == 0\\
      or (int(dy) == len(data[0, :])-1) or int(dy) == 0:
        radius = r+1
        print(dx)


    self.radius = np.zeros(len(self.x))
    self.theta = np.zeros(len(self.x))
    self.x_cyl = np.zeros(len(self.x))
    self.y_cyl = np.zeros(len(self.x))

    #theta_max = 58.9 deg
    alpha = 58.9/r #coefficient de conversion en degres
    for i in range(len(self.x)):
      self.radius[i] = np.sqrt((self.x[i] - y0)**2\\
      + (self.xp[i] -x0)**2)
      self.x_cyl[i] = self.radius[i]*cos(phi) + x0
      self.y_cyl[i] = y0-self.radius[i]*sin(phi)
      self.theta[i] = self.radius[i]*alpha



  def graph_1D(self, i):
    figure(i)
    plot(self.theta, self.y, label="data")
    xlabel(r'$\theta(^{\circ})$', fontsize=15)
    ylabel(r'$I\quad$ (arbitrary unit)', fontsize = 15)
    self.name = self.name[0:len(self.name)-4]
    self.name = self.name + "_proftheta_1D.png"
    savefig(self.name)
    #show()

class profile_phi:
  """
  class used for plot the profile of the spectrum images
  as function of the angle phi, adjusted for the
  oil-objectif with NA = 1.3 and the refractive index of
  the oil n =1.518. Schould be call from the function
  graph_2D_prof for a optimal use
  contains the function:
    graph_1D
  """
  def __init__(self, data, name, x0, y0, r, theta):
    #input variables
    self.name = name
    self.data = data

    #interne variables
    self.x = list() #abscisse
    self.y = list()
    self.xp = list()
    self.angle = list()

    self.maxi = len(self.data[0, :])

    f_x = 1
    f_y = -1
    alpha = 58.9/r#58.9 is the theta_max
    r = theta*180/(np.pi*alpha)#determination of the radius

    dx = x0 + r
    dy = y0
    dphi = 0.02
    phi = 0



    #general loop
    while (phi < 2*np.pi):
      self.x.append(int(dy))
      self.xp.append(int(dx))
      self.y.append(data[int(dy), int(dx)])
      self.angle.append(phi)
      phi = phi + dphi

      dx = x0 + f_x*r*sin(phi)#*dphi
      dy = y0 + f_y*r*cos(phi)#*dphi
#      print(dx, dy)

      #Boundaries conditions
      if (int(dx) == len(data[0, :]-1)) or int(dx) == 0\\
      or (int(dy) == len(data[0, :])-1) or int(dy) == 0:
        radius = r+1
        print(dx)

    self.x = self.x[1:len(self.x)]
    self.xp = self.xp[1:len(self.xp)]
    self.y = self.y[1:len(self.y)]
    self.angle = self.angle[1:len(self.angle)]


  def graph_1D(self, i):
    labelx=-0.2
    fig = plt.figure()
    ax = fig.add_axes([0.15, 0.1, 0.8, 0.8], polar=True)
    ax.plot(self.angle, self.y)
    ax.set_ylabel(r'$\varphi(^{\circ})$', fontsize=18)
    ax.yaxis.set_label_coords(labelx, 0.5)

class profile_temoin:
  """
  class which plot the profile of a picture
  and which allow to compute the pixel size of a picture
  if the size of some object are know, here temoins picture
  contains the function:
    approximation
    pixel_size
    graph_1D
  """
  def __init__(self, data, f, name):
    self.name = name
    self.data = data
```

```
    self.f = f
    self.x = np.arange(len(self.data[:, f]))
    self.y = self.data[:, f]

  #give a approximation of the profile
  def approximation(self):
    wall = list()
    nose = list()
    self.approx = list()

    maxy = 0.9*max(self.y)
    miny = 0.2*max(self.y)

    for i in range(len(self.y)):
      if self.y[i] > maxy:
        wall.append(self.y[i])
      if self.y[i] < miny:
        nose.append(self.y[i])

    self.average_W = sum(wall)/len(wall)
    self.average_n = sum(nose)/len(nose)
    middle = 0.5*(self.average_W + self.average_n)/2.0

    for i in range(0, len(self.y)):
      if self.y[i] > middle:
        self.approx.append(self.average_W)
      else:
        self.approx.append(self.average_n)

  #compute the pixel size
  def pixel_size(self, square):
    maxx = self.approx.index(max(self.approx))


    i = maxx
    while self.approx[i] != self.average_n:
      i = i + 1
    minimum = i

    while self.approx[i] == self.average_n:
      if i == len(self.approx)-1:
        self.approx[i] = 1
        self.radius = (minimum - maxx)/2.0
      else:
        i = i + 1
    maximum = i
    self.square_size = maximum - minimum
    self.pixel_size = square/self.square_size #in \mu m
    print(self.pixel_size)

  #plot the profile
  def graph_1D(self, i):
    self.approximation()
    figure(i)
    plot(self.x, self.y, label="data")
    plot(self.x, self.approx, label="approx")
    self.name = self.name[0:len(self.name)-4]
    self.name = self.name + "_1D.png"
    savefig(self.name)
    #show()

class list_pic:
  """
  class which take all the picture of the folder "folder"
  and which create the folder "name" with the same
  arborescence than "folder" in order to save the computed
  data in this folder.
  contains the function:
    ite_folder
  """
  def __init__(self, folder, name):
    self.folder = folder
    self.name = name
    self.pict = list()
    self.ite_folder(self.folder, self.name)

  def ite_folder(self, folder, name):
    for i in range(len(folder)):
      doc = os.listdir(folder[i])
      subfolder, picture = list_folder(doc)
      for b in range(len(subfolder)):
        subfolder[b] = folder[i] + "/" + subfolder[b]
      for a in range(len(picture)):
        self.pict.append(folder[i] + "/" + picture[a])
      create_folder("result-pic" + "/" + folder[i])
      if len(subfolder) != 0:
        self.ite_folder(subfolder, name)

def graph_2D(data, name):
#functon which plot a 2D representation of a picture
  f = figure()
  # create imshow subplot
  ax = f.add_subplot(111)
  #change v_max and v_min in order to change
  #the color scale
  result = ax.imshow(data, cmap=cm.jet, origin='upper',\\
  vmax=abs(data).max(), vmin=0.3*abs(data).max())

  cbar = colorbar(result, orientation='vertical')
  cbar.set_label(r"Intensity (arbitrary unit)",\\
  fontsize=10)
  cbar.ax.set_yticklabels(['10', '20', '30', '40', '50',\\
  '60', '70', '80', '90'])# horizontal colorbar

  name = name + "_2D.png"
  savefig("result-pic/" + name)


def graph_2D_prof(data, phi, theta, name):
  #function which plot the 2D representation of the fourier
  #picture and with used the class profile_theta in order
  #to plot the profile of theta

  f = figure()
  # create imshow subplot
  ax = f.add_subplot(111)

  result = ax.imshow(data, cmap=cm.jet, origin='upper',\\
  vmax=abs(data).max()/1, vmin=0.05*abs(data).max())

  x0 = 520
  y0 = 500
  r= 260
  phi = phi*np.pi/180
  theta = theta*np.pi/180

  x, y = circle(x0, y0, r)
  #ax.plot(x, y, 'y')
  ax.plot(x0, y0, 'o', color= 'red', markersize=2)

  #theta profile
  prof_theta = profile_theta(data, name, x0, y0, r, phi)

  ax.plot(prof_theta.xp, prof_theta.x, 'green',\\
  linewidth = 3)

  #phi profile
  prof_phi = profile_phi(data, name, x0, y0, r, theta)

  ax.plot(prof_phi.xp, prof_phi.x, 'red', linewidth = 3)
  ax.axis([0, 1000, 1000, 0])
```

```
  cbar = colorbar(result, orientation='vertical')
  cbar.set_label(r"Intensity (arbitrary unit)",\\
  fontsize=15)
  cbar.ax.set_yticklabels(['10', '20', '30', '40', '50',\\
  '60', '70', '80', '90'])# horizontal colorbar

  prof_theta.graph_1D(5)
  prof_phi.graph_1D(6)
  name = name + "_2D.png"
  savefig("result-pic/" + name)
  show()

def graph_3D(data, name):
#functon which plot a 3D representation
#of a picture with mayavi
  x, y = np.mgrid[0:len(data[:, 0]):1, 0:len(data[0, :]):1]
  mlab.surf(x, y, data)
  name = name+ "_3D.png"
  mlab.savefig("result-pic/" + name)
  mlab.show()


def circle(x0, y0, r):
  #just plot a circle used in graph_2D_prof in order to
  #find the NA circle
  theta = np.linspace(0, 2*np.pi, 100)
  x = r*cos(theta) + x0
  y = r*sin(theta) + y0

  return x, y

def scale_bar(name, pix):
#function which display on a picture the scale bars
#only for the file from the real camera and
#them math representation
#given by matplotlib (function graph_2D)

  checkListe = name.split("/")
  #nb of pixel for 10micro
  px10 = int(10/pix)
  for i in range(0, len(checkListe)):
    if checkListe[i] == "real":
      #print("pix = ", pix)
      im = Image.open(name)
      draw = ID.Draw(im)
      #draw  line
      if pix == pix_mat:
        x1 = int(0.3*im.size[0])
      else:
        x1 = int(0.1*im.size[0])

      y1 = int(0.75*im.size[1])
      x2 = x1 + px10
      y2 = y1
      draw.line((x1, y1) + (x2, y2), fill="red",\\
      width = 15)
      del draw
      name = name[0:len(name[i])-5]
      name = name + "_scale.png"
      word = Image.open("image/ten-micro.png")\\
      .convert('RGBA')
      x3 = x1 + int((x2-x1)/2) - 50
      y3 = y1 + 25

      arr=np.array(np.asarray(word))
      r,g,b,a=np.rollaxis(arr,axis=-1)
      mask=((r==255)&(g==255)&(b==255))
      arr[mask,3]=0
      word=Image.fromarray(arr,mode='RGBA')

      im.paste(word, (x3, y3), mask=word)
      if checkListe[0] == "result-pic":
        im.save(name)
      else:
        im.save("result-pic/" + name)

def list_folder(fichier):
#function which give the list of the subfolder and the list
#of files of a folder
  folder = list()
  list_fichier = list()

  for i in range(len(fichier)):
    a = True
    b = True
    for j in range(len(fichier[i])):
      if '.' == fichier[i][j]:
        a = False
        b = False
        j = len(fichier[i]) + 1

    if fichier[i] == "result-pic":
      a = False
      b = True

    if a:
      folder.append(fichier[i])
    if b==False and fichier[i][len(fichier[i])-1] != 't':
      list_fichier.append(fichier[i])

  return folder, list_fichier

def create_folder(name):
#function which create the folder name
  try:
    os.mkdir(name)
  except OSError:
    pass

def test_graph():
  x = np.linspace(0, 2*np.pi, 1000)

  #box = dict(facecolor='yellow', pad=5, alpha=0.2)
  labelx=-0.1
  fig = plt.figure()
  #ax = fig.add_subplot(111)
  ax = fig.add_axes([0.25, 0.25, 0.6, 0.6], polar=True)
  ax.plot(x, x)
  #polar(self.angle, self.y, label="data")
  ax.set_ylabel(r'$\varphi(^{\circ})$', fontsize=15)
  ax.yaxis.set_label_coords(labelx, 0.5)

  plt.show()
```

power.py

```
#program used for the computation of the power
#wrote by marc barbry
#01/07/2013

from plot import *
import re

class measure:
  def __init__(self, array, info):
```

```
    self.array = array
    self.info = info
    self.data_pic = list(np.zeros(len(self.array)))
    self.info_data = list()#np.zeros(len(self.array))
    self.T = np.zeros(len(self.array))
    self.exp_time = np.zeros(len(self.array))
    self.nb_pic = np.zeros(len(self.array))
    self.pre_amp = np.zeros(len(self.array))
    self.G = np.zeros(len(self.array))
    self.P_out = np.zeros(len(self.array))

    input_data = np.loadtxt('IR-power.txt')
    self.P_inp = input_data[:, 1]

    for i in range(len(self.array)):
      print(self.array[i])
      print(self.info[i])

      self.data_pic[i] = np.loadtxt(self.array[i])
      self.info_data.append(np.loadtxt(self.info[i]))
      self.T[i] = self.info_data[i][0]
      self.exp_time[i] = self.info_data[i][1]
      self.nb_pic[i] = self.info_data[i][2]
      self.pre_amp[i] = self.info_data[i][7]
      self.G[i] = self.info_data[i][8]
      if self.G[i] == 0:
        self.G[i] = 1.0
      self.P_out[i] = power(self.data_pic[i], self.T[i],\\
      self.exp_time[i], self.nb_pic[i], self.pre_amp[i],\\
      self.G[i])

  def fit(self):
    self.P_inp[0] = 0
    self.P_out[0] = 0
    alpha, beta, gamma = np.polyfit(self.P_inp[0:len(self.\\
    P_inp)-1], self.P_out[0:len(self.P_inp)-1], 2)
    self.x = linspace(0, max(self.P_inp)+50, 1000)
    self.y = alpha*self.x**2 + beta*self.x + gamma



def power(data_pic, T, exp_time, nb_pic, pre_amp, G):
  noise = 0
  count = 0

  for n in range(0, 150):
    for j in range(0, 150):
      noise = noise + data_pic[n, j]
      count = count + 1
  noise = noise/count

  for n in range(len(data_pic[:, 0])):
    for j in range(len(data_pic[0, :])):
      data_pic[n, j] = data_pic[n, j] - noise
      if data_pic[n, j]<0:
        data_pic[n, j] =0

  nb_inc_ph = (sum(data_pic)*pre_amp)/(G*eta_eff*exp_time*nb_pic)
  nb_org_ph = nb_inc_ph/transmission #nb photon per second
  power = nb_org_ph*h*nu

  return power


def order(array, info):

  array[0] = 'image_data/pic0.sif.don'
  info[0] = 'info/pic0.sif.don_info.dat'
  for i in range(1, len(array)):
    array[i] = 'image_data/pic{0}.sif.don'.format(2*len(array)-2*(i+1)
    info[i] = 'info/pic{0}.sif.don_info.dat'.format(2*len(array)-2*(i+1
  return array, info
```

B Filter

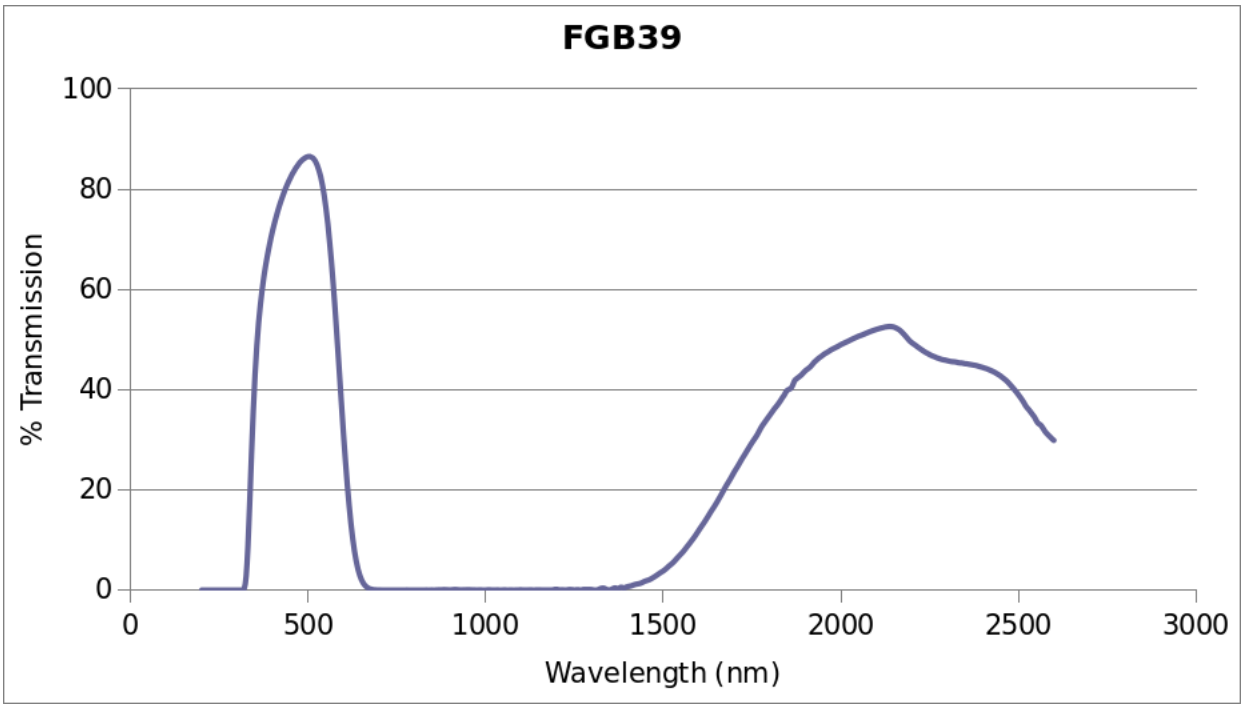


Figure B.1: Transmission of the BG39 filter used in order to remove the IR signal