



*LIGO Laboratory / LIGO Scientific Collaboration*

LIGO-T0900477-v4

*LIGO*

Date

Fast fitting non linear least squares algorithm of 2D gaussian beam images for LIGO optical lever

Andrea Lottarini

Distribution of this document:  
LIGO Scientific Collaboration

This is an internal working note  
of the LIGO Laboratory.

**California Institute of Technology**  
**LIGO Project – MS 18-34**  
**1200 E. California Blvd.**  
**Pasadena, CA 91125**  
Phone (626) 395-2129  
Fax (626) 304-9834  
E-mail: info@ligo.caltech.edu

**Massachusetts Institute of Technology**  
**LIGO Project – NW22-295**  
**185 Albany St**  
**Cambridge, MA 02139**  
Phone (617) 253-4824  
Fax (617) 253-7014  
E-mail: info@ligo.mit.edu

**LIGO Hanford Observatory**  
**P.O. Box 1970**  
**Richland WA 99352**  
Phone 509-372-8106  
Fax 509-372-8137

**LIGO Livingston Observatory**  
**P.O. Box 940**  
**Livingston, LA 70754**  
Phone 225-686-3100  
Fax 225-686-7189

<http://www.ligo.caltech.edu/>

**Abstract:**

The main purpose of the Optical Lever system is to reacquire the lock of the interferometer, when the lock is lost. The aim of this project is to study the feasibility of replacing the current quadrant photodetector (QPD) readout system with a CCD. Using a 2D fast centroid finding algorithm, we are able to create a more stable control signal that accounts for the thermal distortion on the surface of the suspended optics. We examined the behavior of the system on 2 meter and 40 meter optical lever. We created numerous different fitting algorithms in MATLAB and compared the results. The two candidate algorithms that showed the best precision and stability were then translated in C++ with optimization for fast real time execution. Both are based on the Gauss-Newton algorithm for non linear least square problems. Also, we learned how to remove most of the diffraction coming from the optics. We then measured the precision of the fit and long term stability of the algorithms in real conditions and in real time. Finally, we compared the results with the requirements for the Advanced LIGO. In conclusion, the algorithms have a short term precision under the  $\mu\text{m}$  that with a lever arm of several meters should safely result in a better precision than the required  $\mu\text{radian}$ .

# 1 Introduction

In LIGO, the optical lever system monitors the angular alignment of the suspended mirrors. The system consists of a laser source and a quadrant photodetector (QPD) that can be mounted on the same structure. The source points to the monitored mirror, and the laser reflects back to the QPD. The sensitivity of the signal from QPD depends on the spot size of the beam at the QPD. A telescope is used to adjust the spot size in order to achieve the best signal from the QPD.

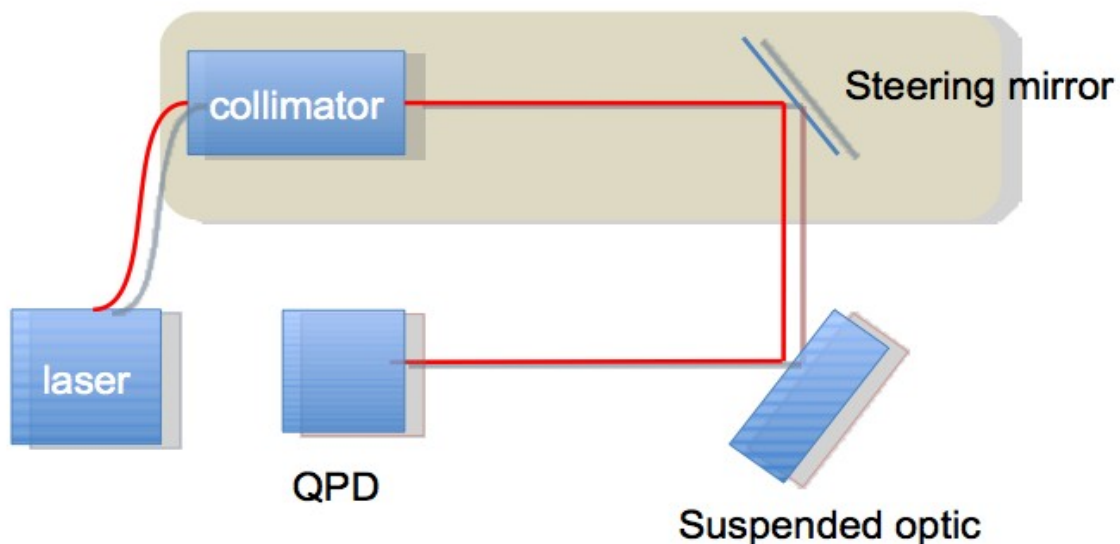
We are studying the possibility of replacing quadrant photodiode with CCD and fast acquisition/centroid finding algorithms to produce position feedback signals.

## 1.1 Purpose

The aim of the paper is to report the results of replacing the QPD with a CCD camera as suggested on LIGO-T0900239-v2.

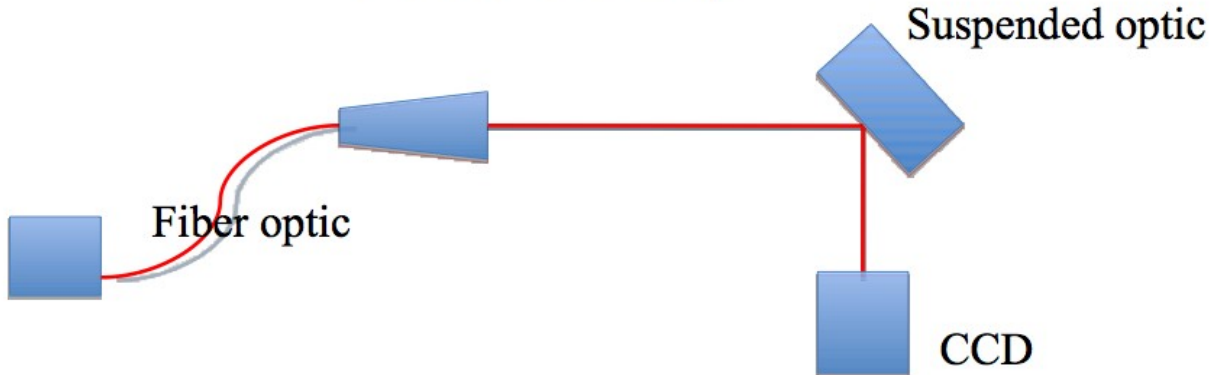
## 1.2 Optical levers setup

Below is the current setup of the optical lever system:



Below is the new and improved setup.

# New setup



By eliminating the steering mirror and using a stiffer launching point for the beam expander telescope, we expect to have a more stable signal.

## 2 Report

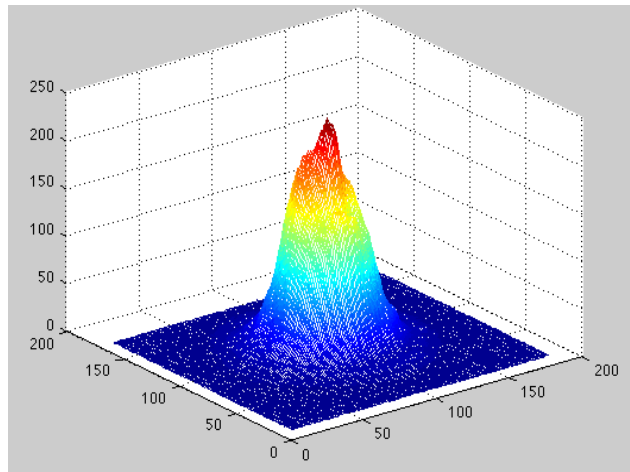
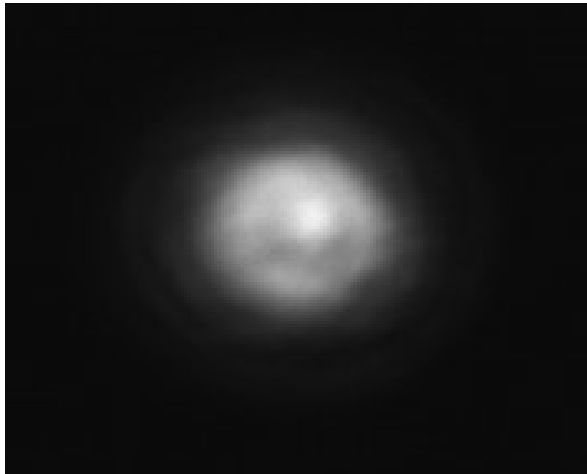
We utilized various cameras from Prosilica manufacturer: GC650, GC750, GC1280, GC1380, GC2450. They are ranging from CCD to CMOS and from low to high resolution.

We started working with the GC750 and the code from the 40mt experiment. After some major adjustment on the code, we decided to rewrite the entire code, starting directly from the Prosilica interface. The main reason was that acquisition and data analysis should be as fast as possible while the 40mt code was developed with different intents.

We extended the Prosilica API with C++ code to make a program capable to take snapshot from the camera at specified time interval.

Below to the left is a snapshot of the laser beam pointed directly towards the CCD of the camera and to the right is its 3D plot. No focusing device is used, only the direct output from the laser pigtail fiber.

Laser Beam	3D plot
------------	---------



The images above reveal that the laser beam has a nice gaussian profile, which we expected. However, we can see some distortion in the shape of the beam. Moreover, we wanted to measure how the shape of the beam changed using a telescope or a collimator. Along with Ian and Tara of the Optical levers group, we made some initial measurements. As specified on LIGO-T0900239-v2, we injected the laser on the telescope using an optical fiber and reflected the laser beam using a mirror. We still lack a stable support for the telescope and thus, we were not able to have many different images for comparison. However, we made various snapshots enough for an initial analysis. As expected, the laser beam passing through the telescope created an airy disk.

Below are the images from different distances and their respective 3D plot:

Figure1: 144 feet between CCD and telescope:

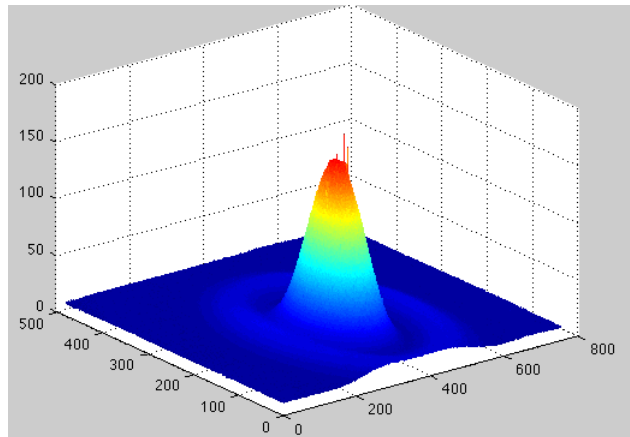
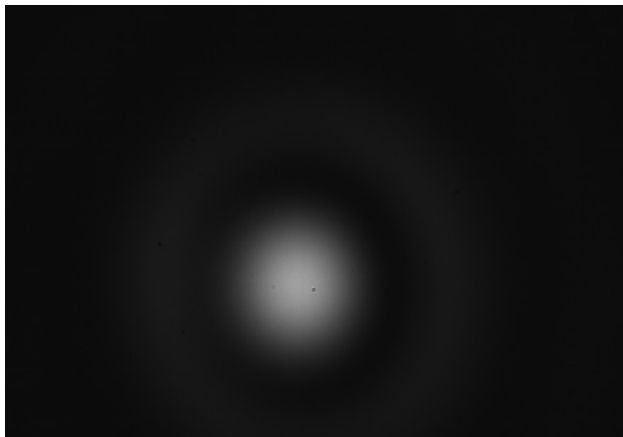
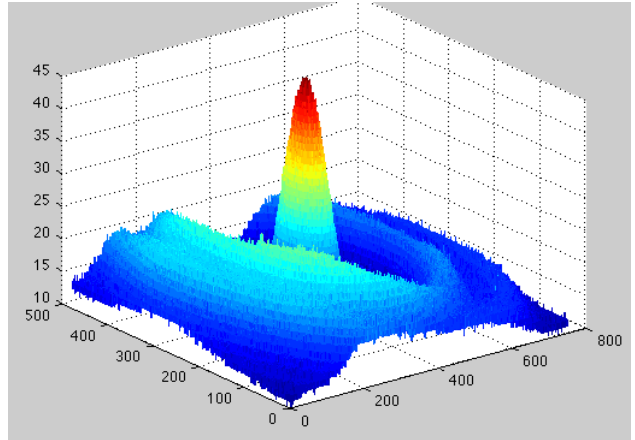


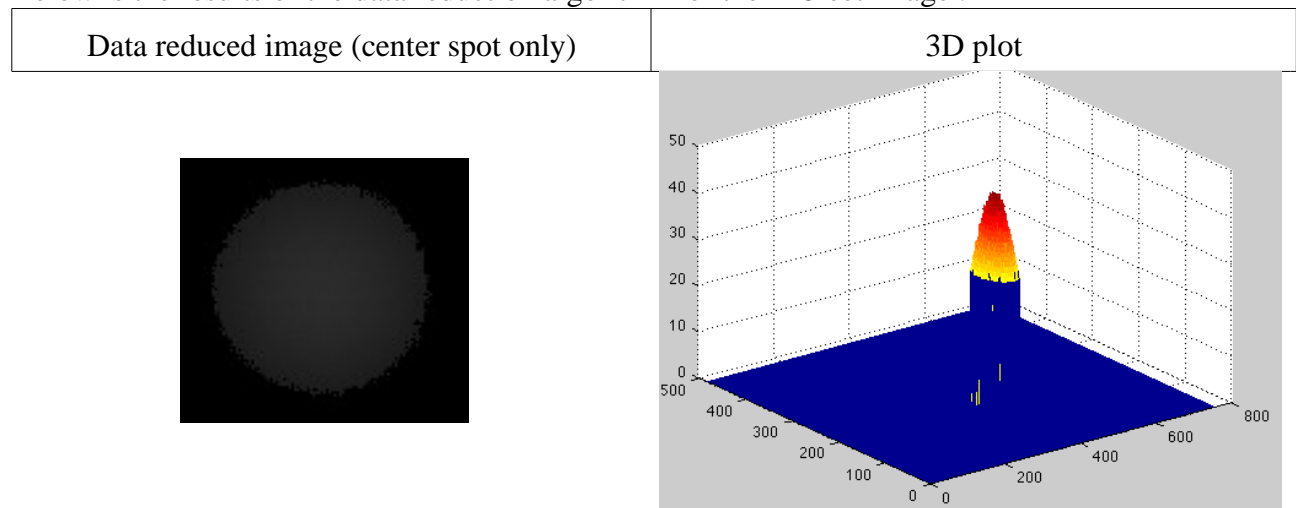
Figure2: 128 feet between CCD and telescope:



In the second picture, we have a bad alignment of the optics causing a huge diffraction pattern and a really low dynamic range; therefore, we take this picture as an extreme case.

At this point, we decided to make a data reduction algorithm that could isolate the center spot of the image. The goal was to demonstrate that the center of the image is recognizable by a computer algorithm and that it still has a gaussian profile.

Below is the results of the data reduction algorithm for the 128feet image :



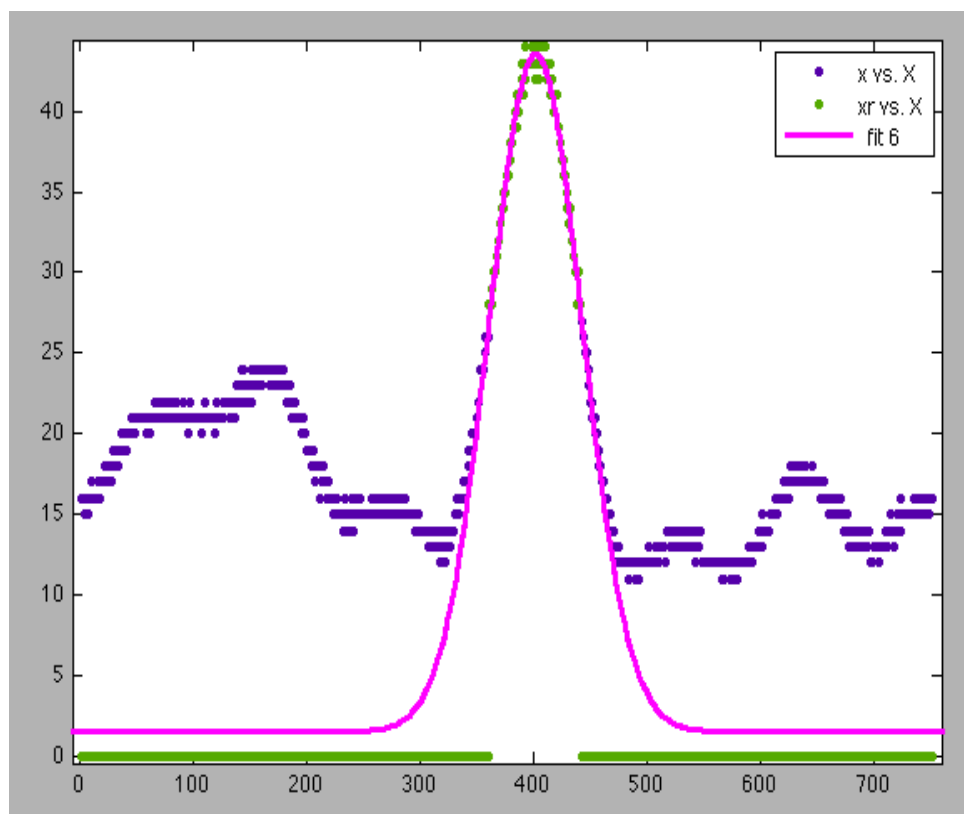
The noise on the 3D plot is irrelevant . An algorithm was created to cancel noise from the picture but it is negligible for our analysis.

It is important that the center of the laser beam is easily detectable even in heavily diffracted images. Moreover, we should consider that the diffraction pattern could be reduced even more by using better optics.

An approximate center detection algorithm was also made based on the luminosity in the central spot. This algorithm was used to study the morphology of the laser beam.

From now on, we will refer this algorithm as the “cookie cutter”.

Below is the 2D profile of the laser beam taken along the X axis passing through the central spot with the relative gaussian fit of the peak.

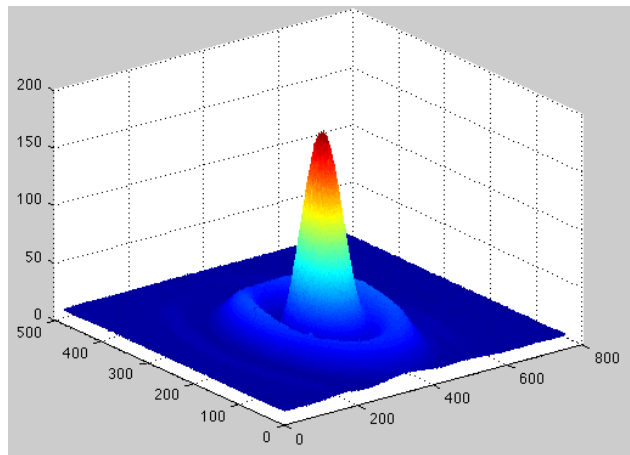
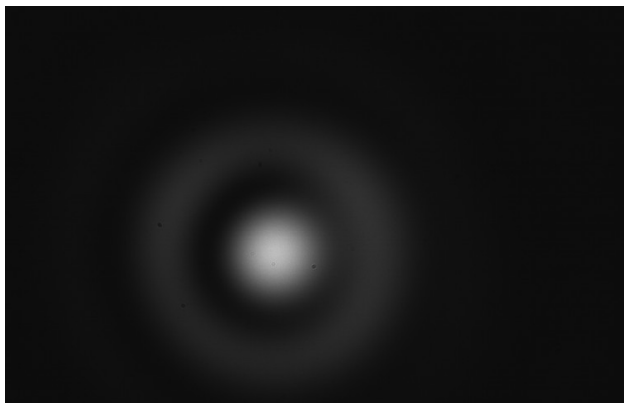


**1: Fit of the central region**

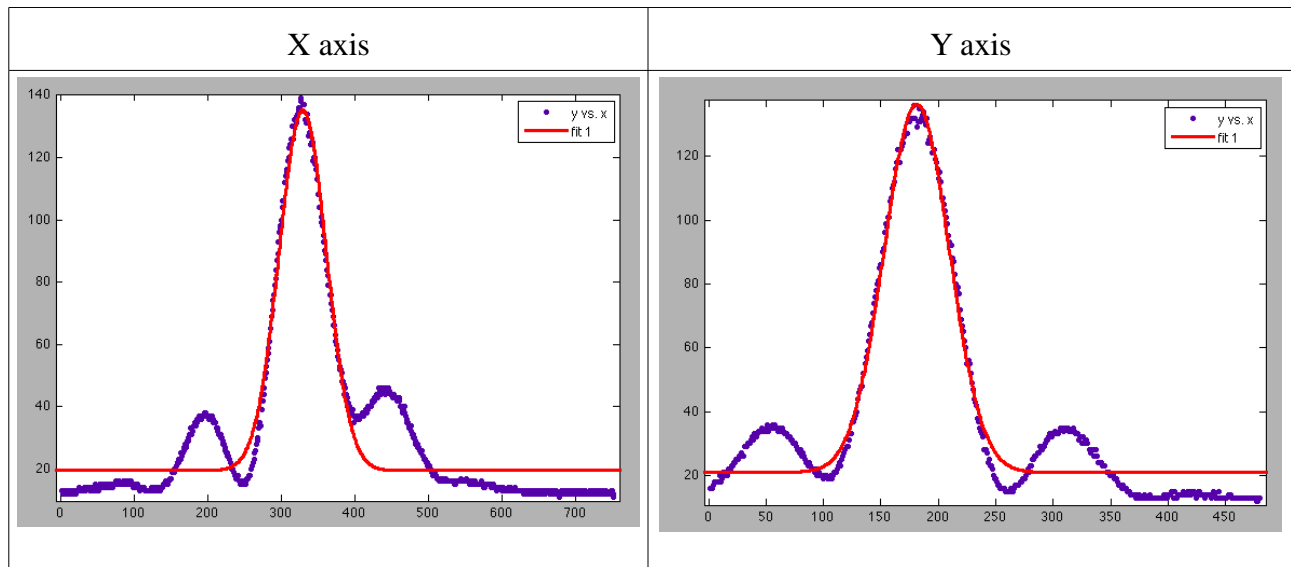
The image shows that the central spot detected by the centroid algorithm (green line) still has a gaussian profile while the rest of the image has an airy disk.

As a next step, we tried to fit the entire image, using the best picture available. Note that choosing the best image is not a workaround for the fitting problem since in the future, we expect to work with better optics and consequently obtain better images.

We chose the image below:



Using the central detection algorithm, we were able to make a gaussian fit along the x and y axis of the image. Below are the results:

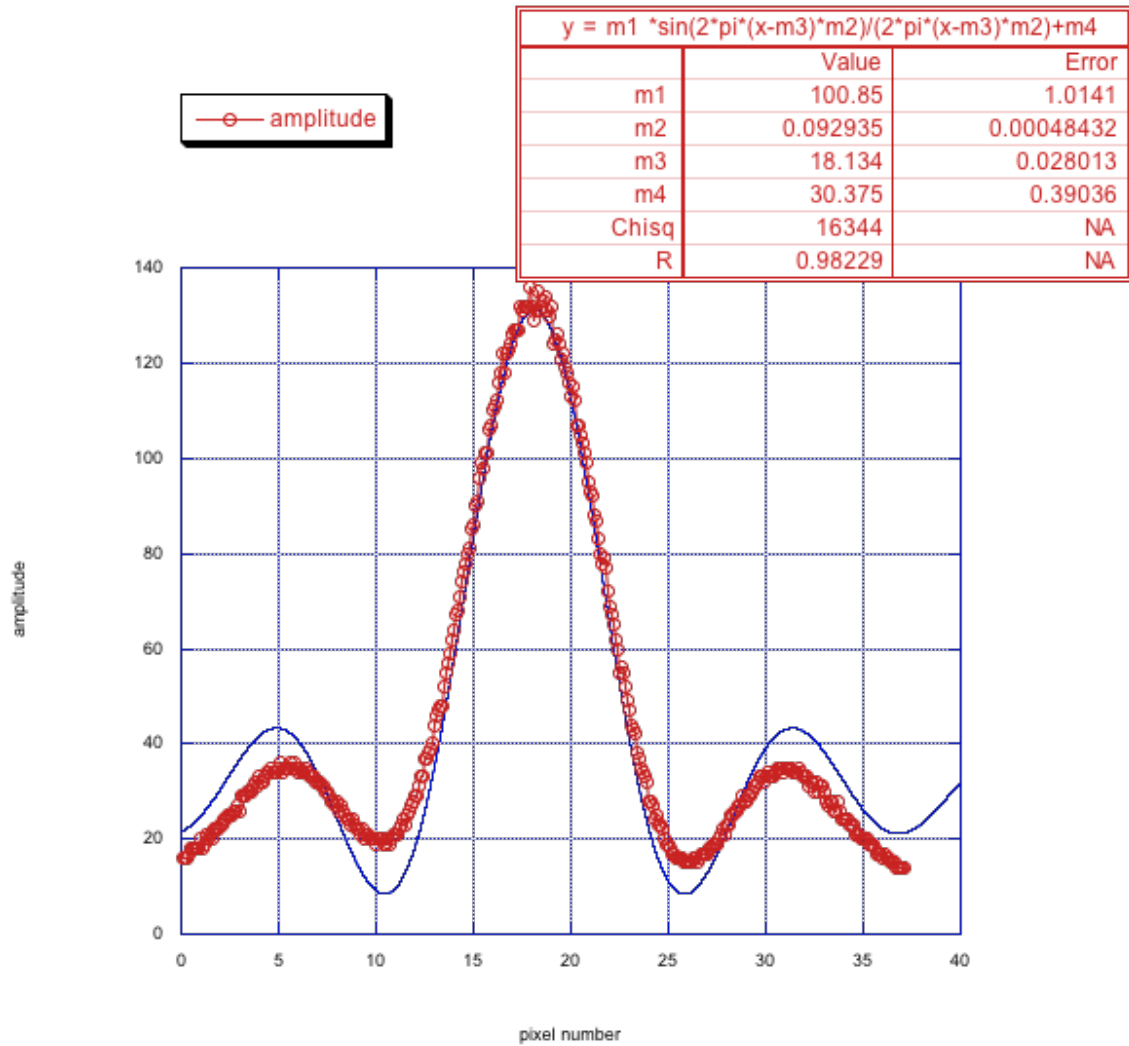


The image is clearly not symmetric about the x-axis while its almost symmetric about the y-axis. This is a known issue caused by the measurement techniques, and can be solved by adding a polynomial correction to the gaussian fit. We decided to go further by analyzing only the Y-axis of the image.



## 2.1 Analysis of a slice:

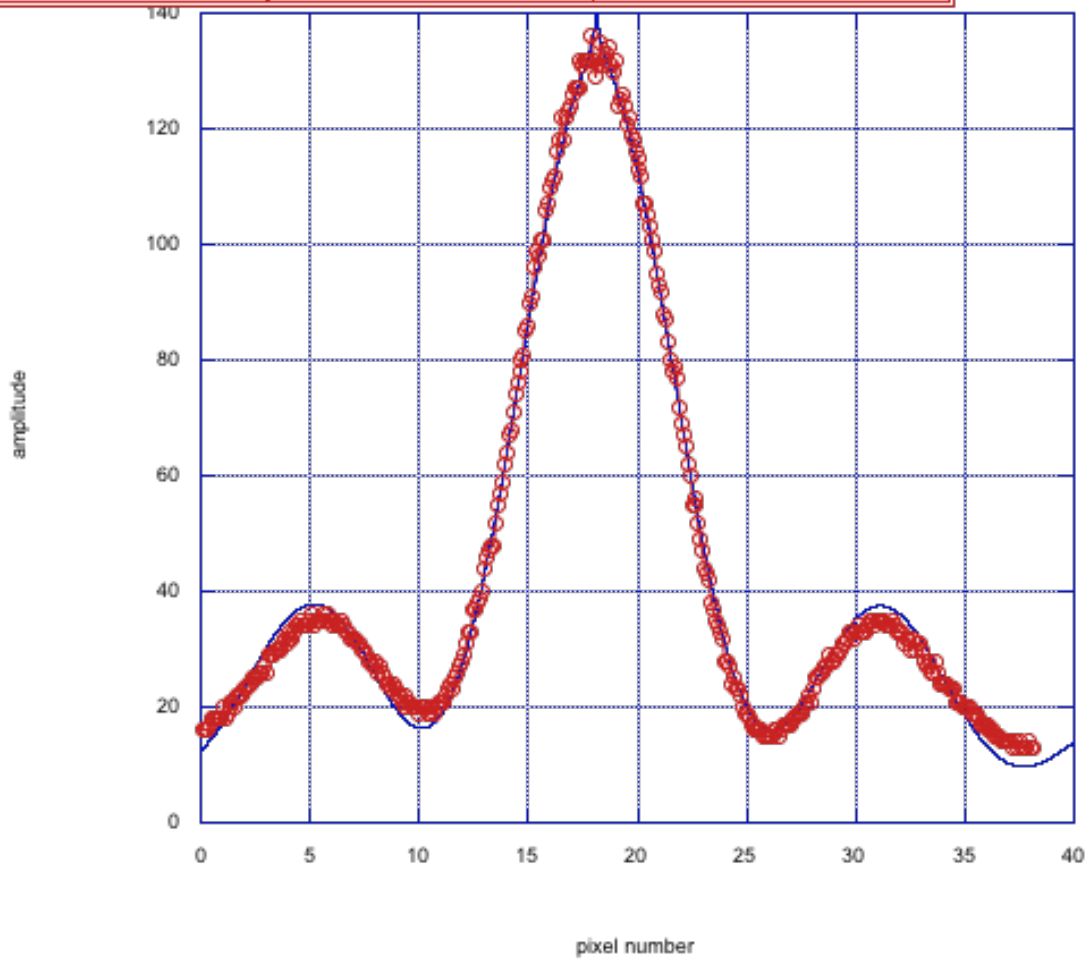
As shown below, the airy function  $\sin(x)/x$  fit does not work very well.



We used  $\frac{\sin(x)}{x}$  plus a power law correction and a baseline Gaussian to account for diffused light, which enhanced our results.

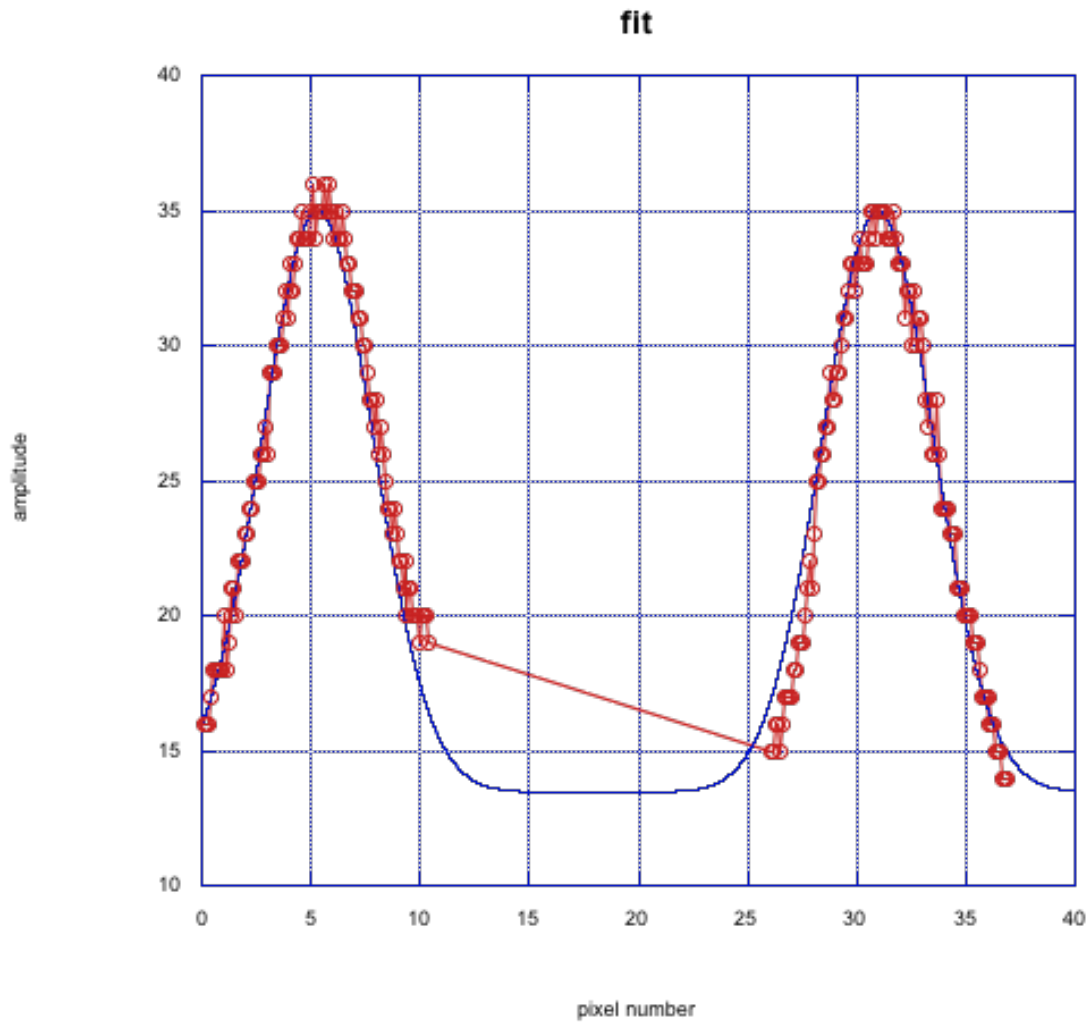
$$y = m1 \cdot \sin(2 \cdot \pi \cdot (x - m3) \cdot m2) / (2 \cdot \pi \cdot (x - m3) \cdot m2) \cdot (\text{abs}(2 \cdot \pi \cdot (x - m3) \cdot m2))^{m5} + m4 + m6 \cdot \exp(-((x - m3) / 20)^2) + m7 \cdot x$$

	Value	Error
m1	89.066	0.42159
m2	0.091585	0.00019926
m3	18.14	0.010077
m4	2.8346	0.62418
m5	-0.029987	0.0030918
m6	37.525	0.73943
m7	-0.010631	0.011287
Chisq	1991.8	NA
R	0.9979	NA



One of the two most interesting variables is the diameter of the Airy rings. Therefore, we cropped the data and fitted the airy wings independently, maintaining the center of the airy ring on the previously found central point (-18.133).

$m1 * (\exp(-((x-18.133+m3)/m2)^2) + \exp(-((x-18.133-m3)/m2)^2)) + m4$		
	Value	Error
m1	21.519	0.49916
m2	3.6202	0.099477
m3	12.834	0.022114
m4	13.465	0.54366
Chisq	313.15	NA
R	0.98274	NA

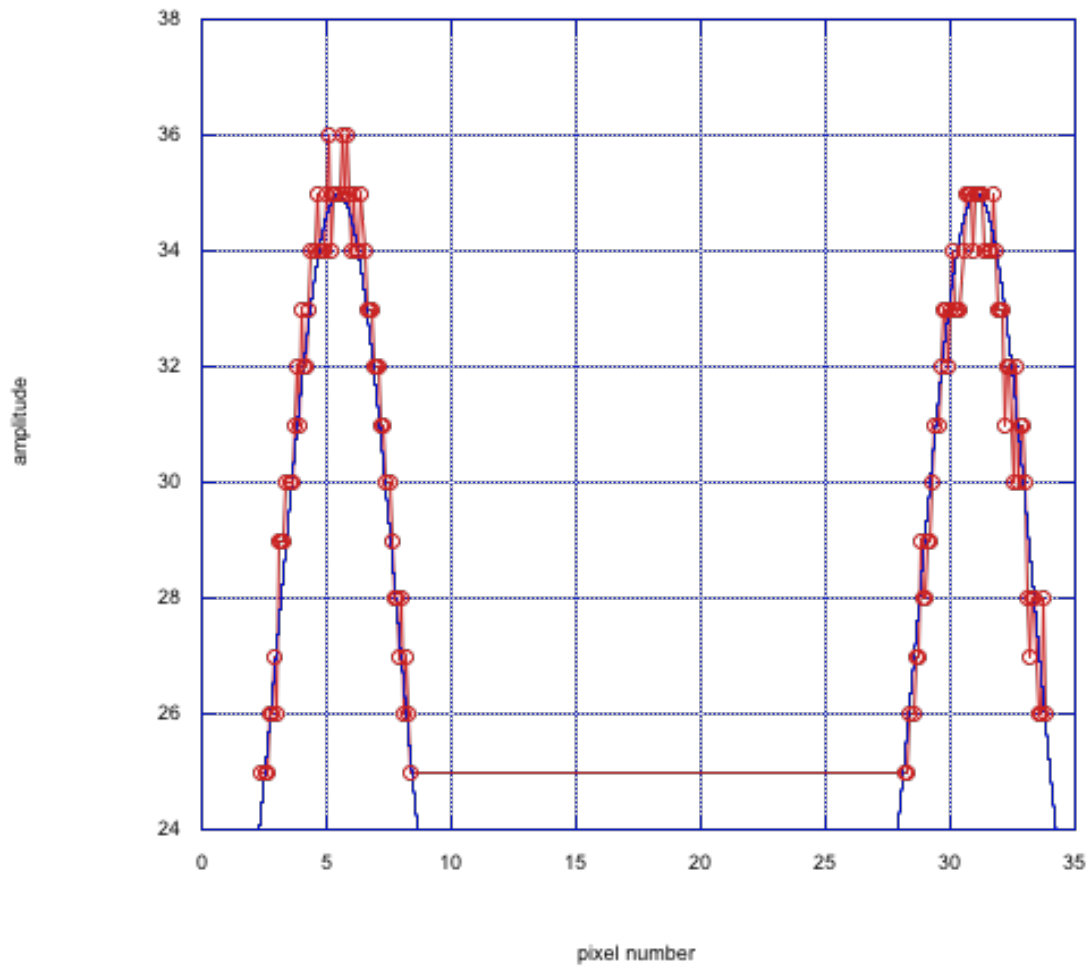


Fitting only the tips of the Gaussians works slightly better.

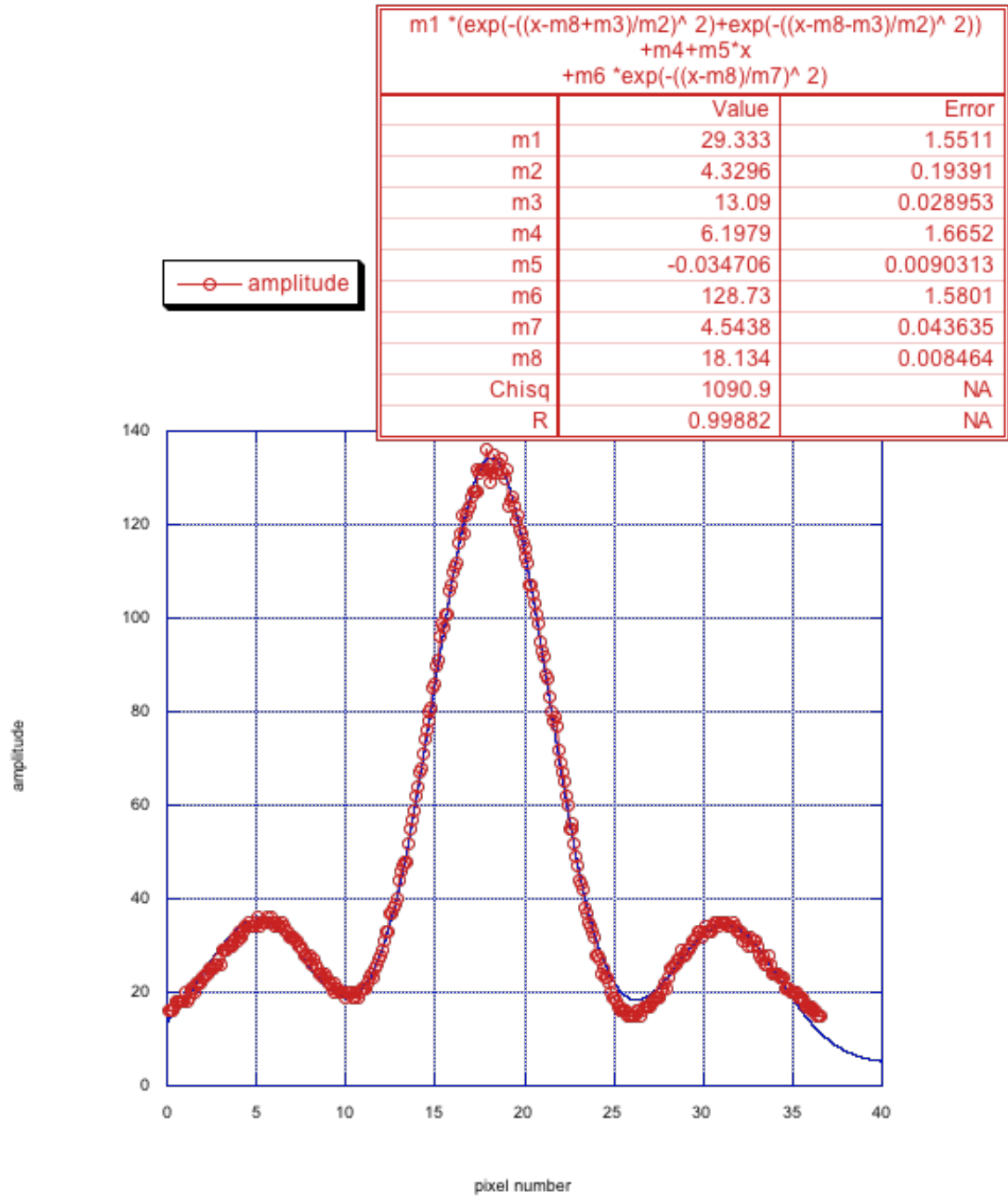
$$y = m1 * (\exp(-((x-m5+m3)/m2)^2) + \exp(-((x-m5-m3)/m2)^2)) + m4$$

	Value	Error
m1	18.308	2.8163
m2	3.3146	0.36098
m3	12.792	0.018552
m4	16.685	2.8841
m5	18.268	0.018629
Chisq	62.191	NA
R	0.97472	NA

fit



Fitting the image with three Gaussians (the two smaller peaks on either side of the image are equi-spaced, relative to the center of the central gaussian curve) produces the best results.



To improve the precision of the measurement of the Airy disk diameter, we took the two peaks, and allowed the fit to find the separation and the centering (by leaving the 18.133 mm centering free).

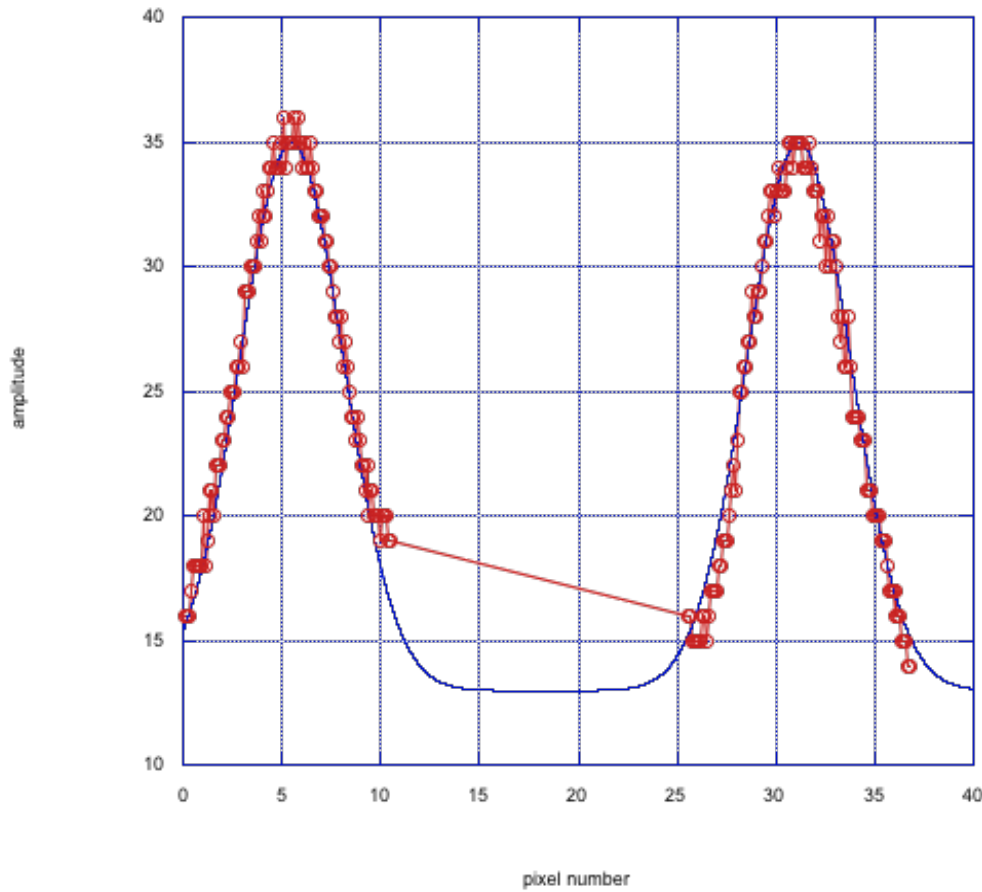
The size of the airy ring is potentially important because it is sensitive to the thermal lensing effect of the optics. A good measurement of the airy disk diameter could give us a way to measure thermal lensing at the position where the optical lever beam hits the mirror. By combining multiple beams, we can have an estimate of the profile of the optics.

The measurement improves ~24% (error from .0221 to .0178)

$$y = m1 * (\exp(-((x-m5+m3)/m2)^2) + \exp(-((x-m5-m3)/m2)^2)) + m4$$

	Value	Error
m1	21.981	0.42187
m2	3.6974	0.083999
m3	12.832	0.017853
m4	12.983	0.46152
m5	18.323	0.017847
Chisq	214.22	NA
R	0.98878	NA

fit

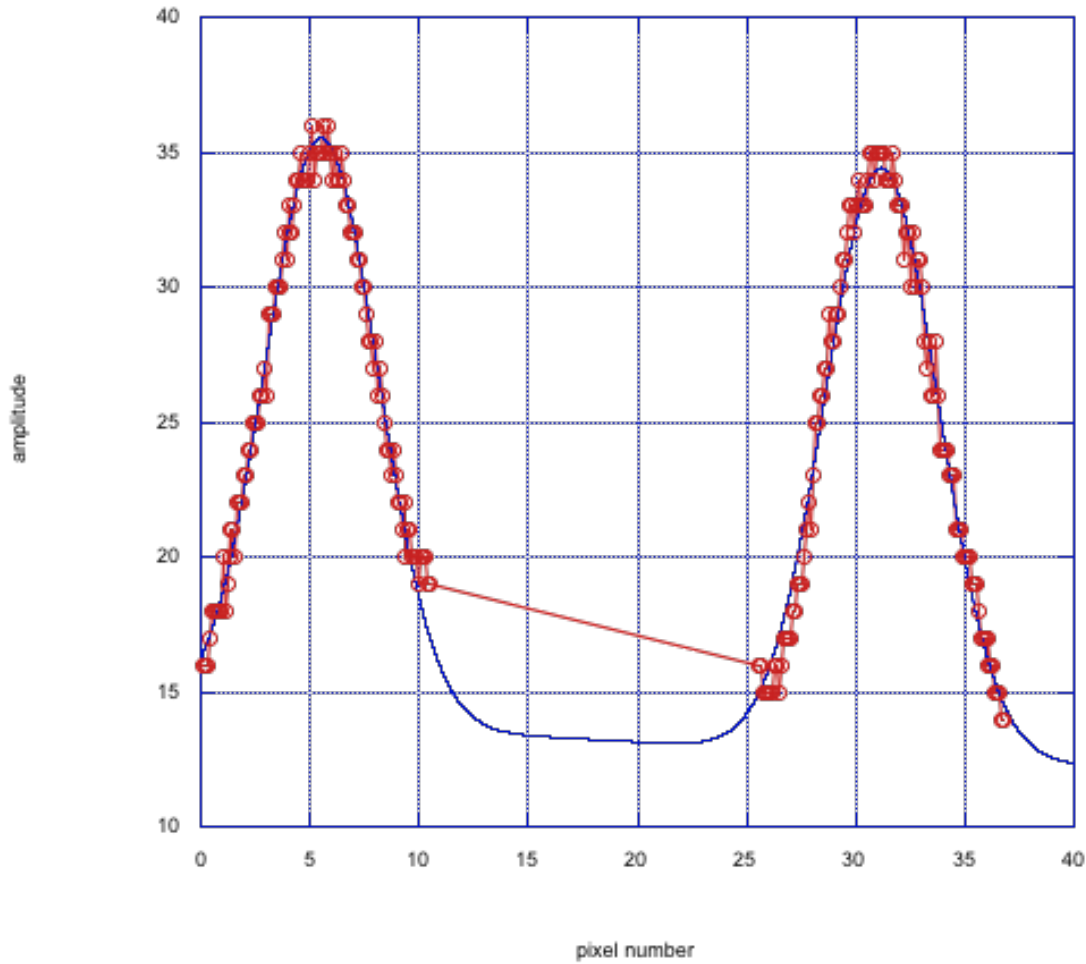


Finally, we add a small slope improving the total error by another 17% (error from .0178 to .0148) while the error on the Airy disk diameter improved by 0.11%.

$$y = m1 * (\exp(-((x-m5+m3)/m2)^2) + \exp(-((x-m5-m3)/m2)^2)) + m4 + m6 * x$$

	Value	Error
m1	21.753	0.34024
m2	3.6648	0.068515
m3	12.83	0.014874
m4	14.009	0.38686
m5	18.355	0.015228
m6	-0.042902	0.0043408
Chisq	146.56	NA
R	0.99234	NA

fit



Note that the precision of the positioning of the central spot is (three Gaussian fit)  $18.134 \pm 0.0085$ , which is  $4.6 \cdot 10^{-5}$ . A 2 mm spot (a preliminary estimate) gives us the precision of positioning of 0.9 microns, with a single slice.

Now, we know that the beam profile is neither a gaussian or an airy.

At this point, we had two options:

1. Define an iterative algorithm that starting with the output of the cookie cutter will slice the beam in the x and y axis and fits the beam profile determining an x and y center as long as width and amplitude.
  - Using one of the fitting functions defined before
2. Fit the whole image using a 2D fitting program in the non linear least square form.

We decided to go for the second option because it will produce the best results even if it is slower than the first (and more time consuming for programming).

## 2.2 2D fitting

We decided to start writing the fitting program in MATLAB to discern which function works best with our pictures and then translate everything in C++ for faster execution.

The algorithm is the Gauss-Newton algorithm for solving non linear least square problems. At first we didn't try to fit the whole image but only the central region determined by the cookie cutter, in other words, the region determined by the cookie cutter algorithm is used as the trust region for the Gauss-Newton algorithm.

The Gauss-Newton algorithm was chosen over the Marquadt method because it is faster. The major drawback is that we have to choose the initial parameter accurately otherwise the algorithm will not converge. This is not a big problem since the centroid algorithm gives us a nice estimate for the initial parameters.

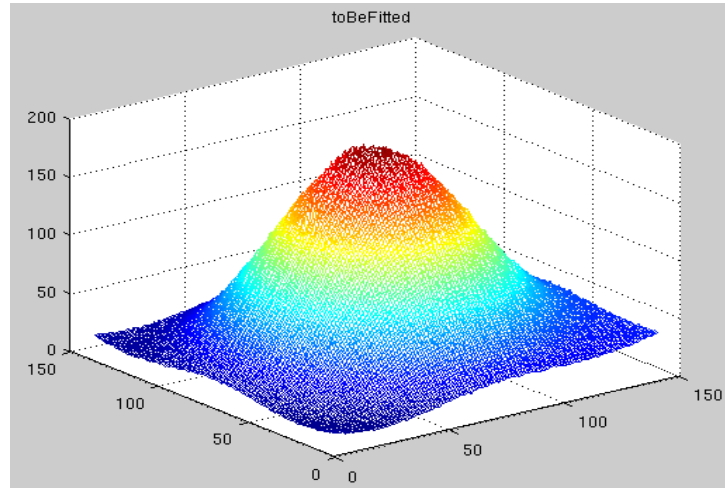
The first function that we used was a 2D gaussian since it is the shape that a laser beam should have when there is no diffraction or noise. A slope was added under the gaussian to take in consideration noise (both coming from the camera and diffused light) and the angular alignment of the beam in respect to the CCD.

This is the formula that we used:

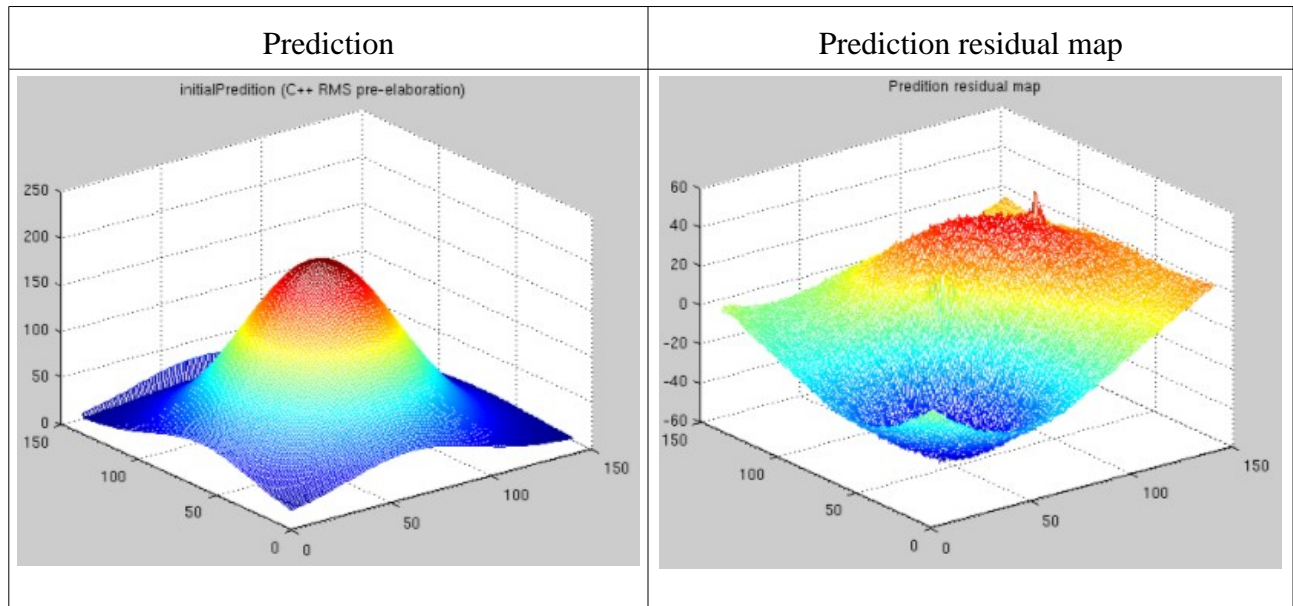
$$A \cdot e^{-\left(\frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2}\right)} + a \cdot x + b \cdot y + c$$

We started from the square section of the gaussian beam (selected by the cookie cutter algorithm) shown below.



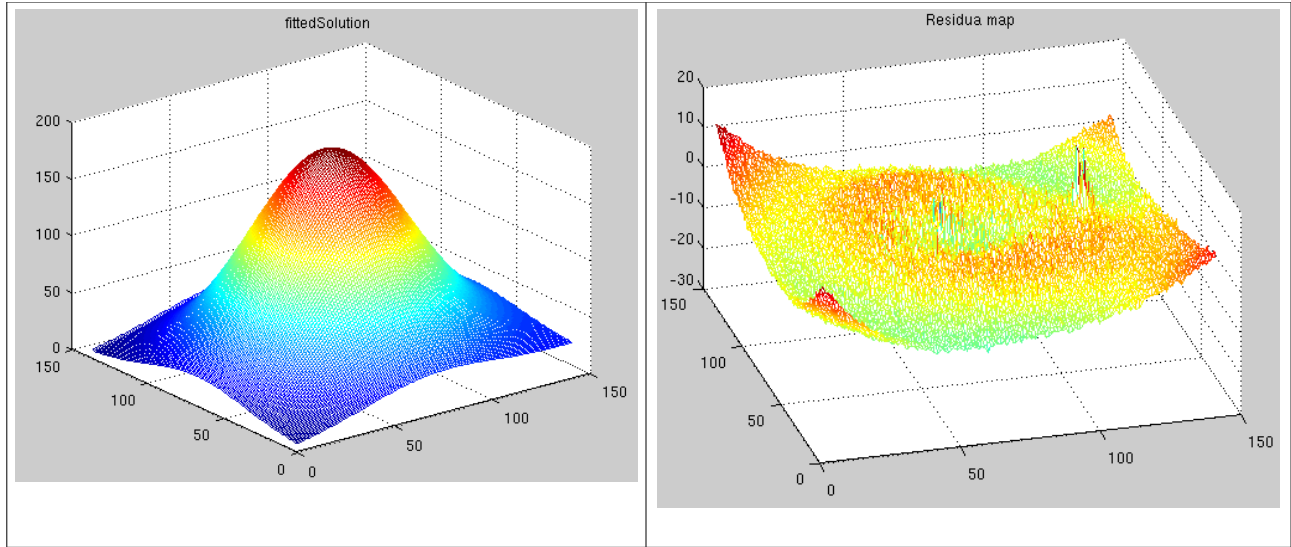


We created the initial prediction from the output of the cookie cutter:

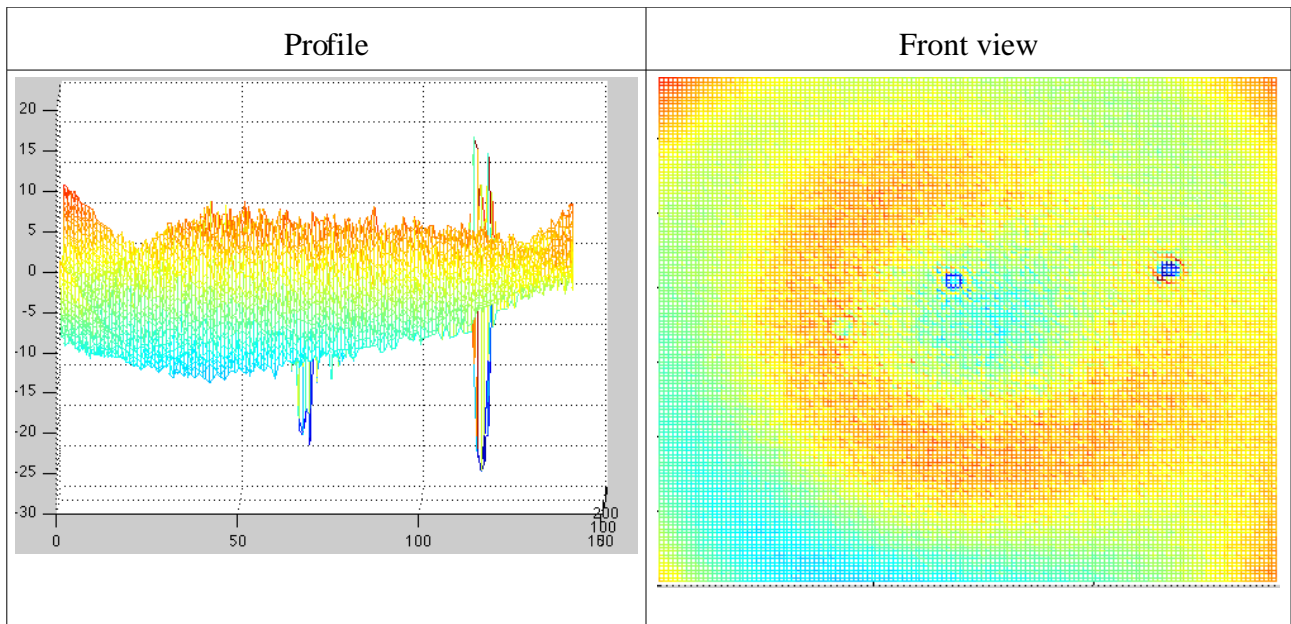


Then, we let the iterative fit algorithm run for ten iteration obtaining the images below:

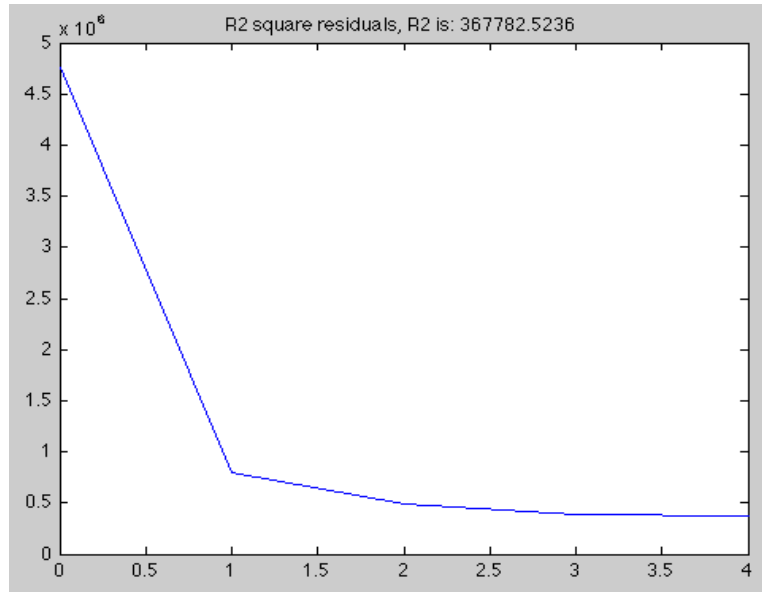




As we can see the algorithm is less precise on the borders of the image, this is because of the first airy disk surrounding the central gaussian spot. We also noticed that the camera that we borrowed from 40mt experiment has two burnt spots:



Then, we examined the convergence of the algorithm and added convergence criteria for the method.



As seen from the graph above, the convergence of the algorithm is quite fast and in fact, the first iteration of the algorithm reduces most of the residuals.

If we manage to make only one iteration per image, we can keep the fit in a linear zone and make it run faster.

In order to improve the precision of the algorithm, we added an elliptical mask over the image.

As an alternative, we could use the output from the cookie cutter as a mask for the image.

One of the problems with the Gaussian fit is that, because of the Airy rings trying to fit a bigger portion of the image (not only the part selected by the cookie cutter) the algorithm does not converge.

Another problem was the dimension of the pixels, which are not perfectly square. We were not able to measure if the laser beam has an elliptical shape or a rotation. In fact if we try to fit the rotation of the beam using the formula:

$$A * e^{-\left(\left(\frac{\cos^2 \Theta}{2\sigma_x^2} + \frac{\sin^2 \Theta}{2\sigma_y^2}\right) * (x-x_0)^2 + 2 * \left(\frac{\sin 2\Theta}{4\sigma_x^2} + \frac{\sin 2\Theta}{4\sigma_y^2}\right) * (x-x_0) * (y-y_0) + \left(\frac{\sin^2 \Theta}{2\sigma_x^2} + \frac{\cos^2 \Theta}{2\sigma_y^2}\right) * (y-y_0)^2\right)}$$

Since the algorithm sets parameter to 0 this is an evidence, that the algorithm is not fitting the difference in width of the real beam but the distortion introduced by the camera.

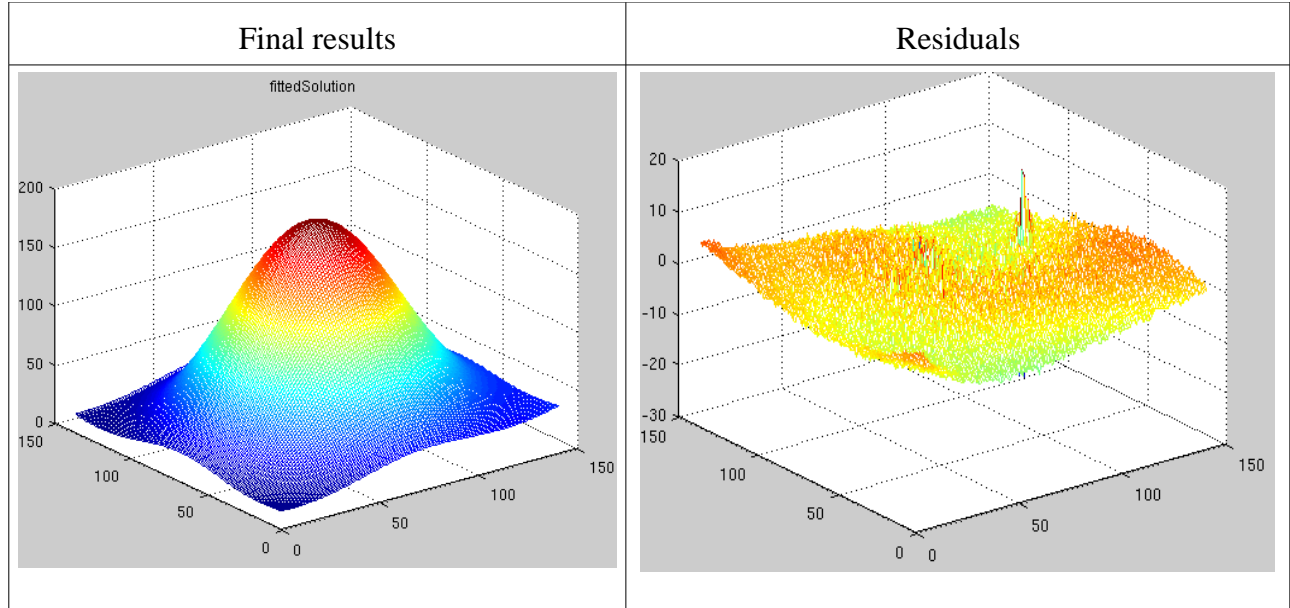
At this point we tried to fit the whole image with an airy. Since we used more pixels ( which means more information), we expected to have better results.

The formula that we used was:

$$f = A \cdot \left( \sin \left( \sqrt{\sigma_x \cdot (x-x_0)^2 + \sigma_y \cdot (y-y_0)^2} \right) \right)^2 / \left( \sigma_x \cdot (x-x_0)^2 + \sigma_y \cdot (y-y_0)^2 \right)$$

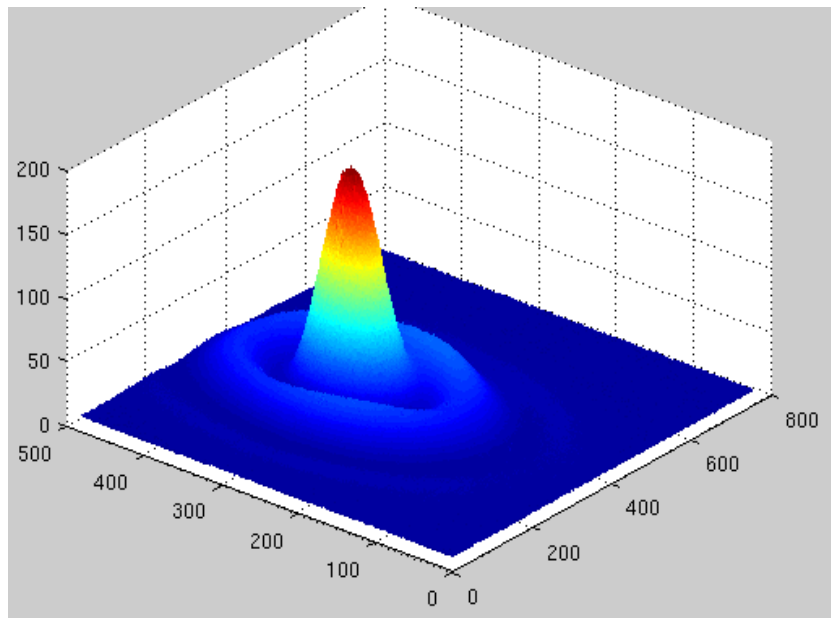
In fact, we did obtain better results and now there are no more limitations about the area that we want to fit. This is an important achievement since we don't have to worry about the convergence, but only about the velocity of the algorithm.

Here it is the result of the fit for the same image:

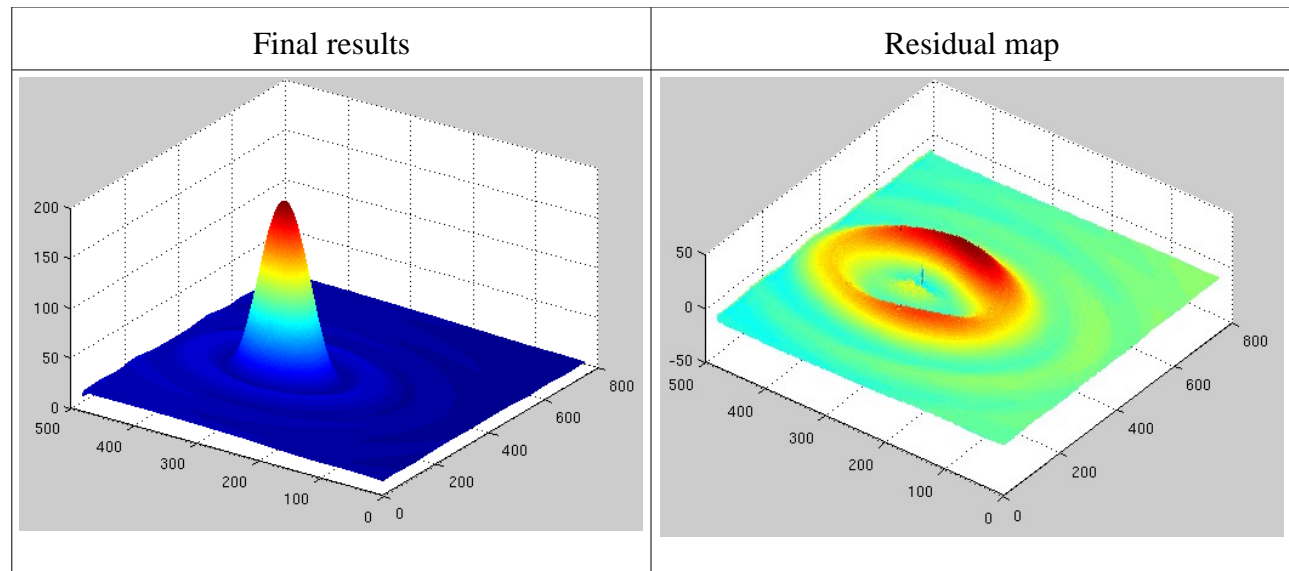


Now we can see that the fit is precise even at the borders of the image.

Below, we tried to fit the whole image:



and we obtained:



The results were not as precise as we hoped. We started trying convolutions of the gaussian and airy functions, but with no outstanding results.

We also tried to verify if the camera introduces some non linearity. We tested this by compensating the luminosity of every pixel before analyzing each picture. The compensating function that we used was:

$$I_m(i) = I_m(i) (1 + I_m(i))$$

From these tests, it appeared that the camera may not have a linear dynamic range and that for high values of luminosity there is compression. (Under the hypothesis that what we should see is a perfect airy).

The alternative hypothesis is aberration in the optical collimator, which is probably the best explanation for the deviations from the fit.

In any case, the fit precision obtained within the cookie cutter area is probably sufficient for the task.

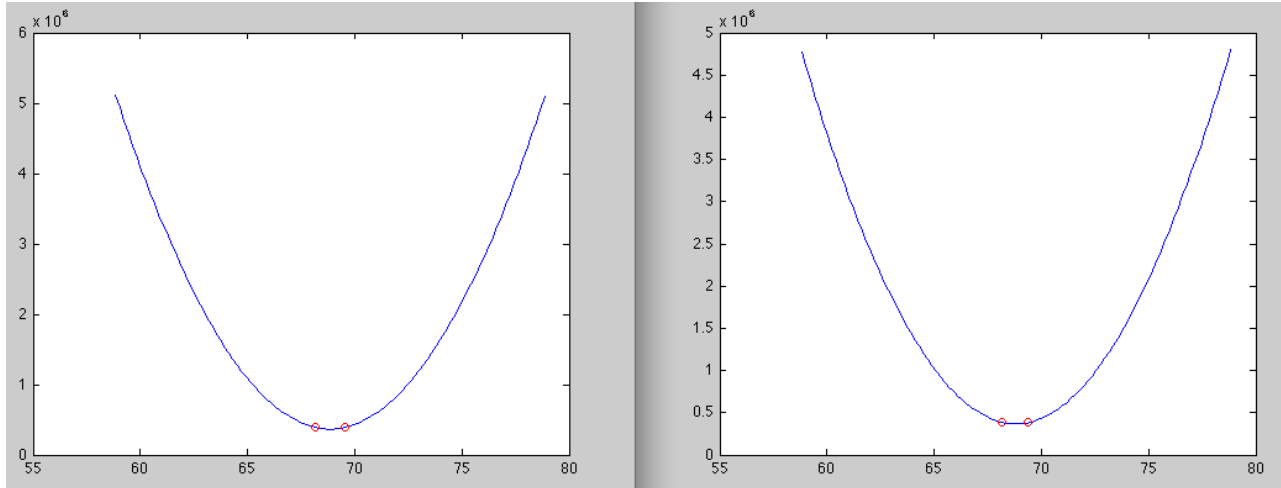
In the mean time some events occurs that shifted our attention towards different problems:

- We borrowed again a camera from the 40mt experiment (the GC650). It became clear that during the first days we made a really poor job injecting the fiber in the beam expander telescope to the point that we were not sure if we needed an airy fitting program anymore.
- We discovered that the fiber suffer from a microphonic effect [section 2.3] which is not well understood yet.
- We received a new camera from Prosilica that shoved a really interesting and bothersome diffraction pattern [section 2.4].

We started analyzing the statistical confidence intervals of our gaussian fit algorithm (limited to the cookie cutter region). We did that by calculating the interval needed to have a 5% increase in the square residuals.

Result shown below were pretty outstanding:

X position: 1.338 pixel → 8.02 m	Y position: 1.236 pixel → 7.4 m
----------------------------------	---------------------------------



This is reasonable because by using so many points for the fitting function, we can have a precision smaller than the single pixel. We compared with the advanced LIGO requirements.

The blue highlighted values below are the precision needed for the positioning while the red values are the precision needed for the thermal distortion determination after the TCS correction.

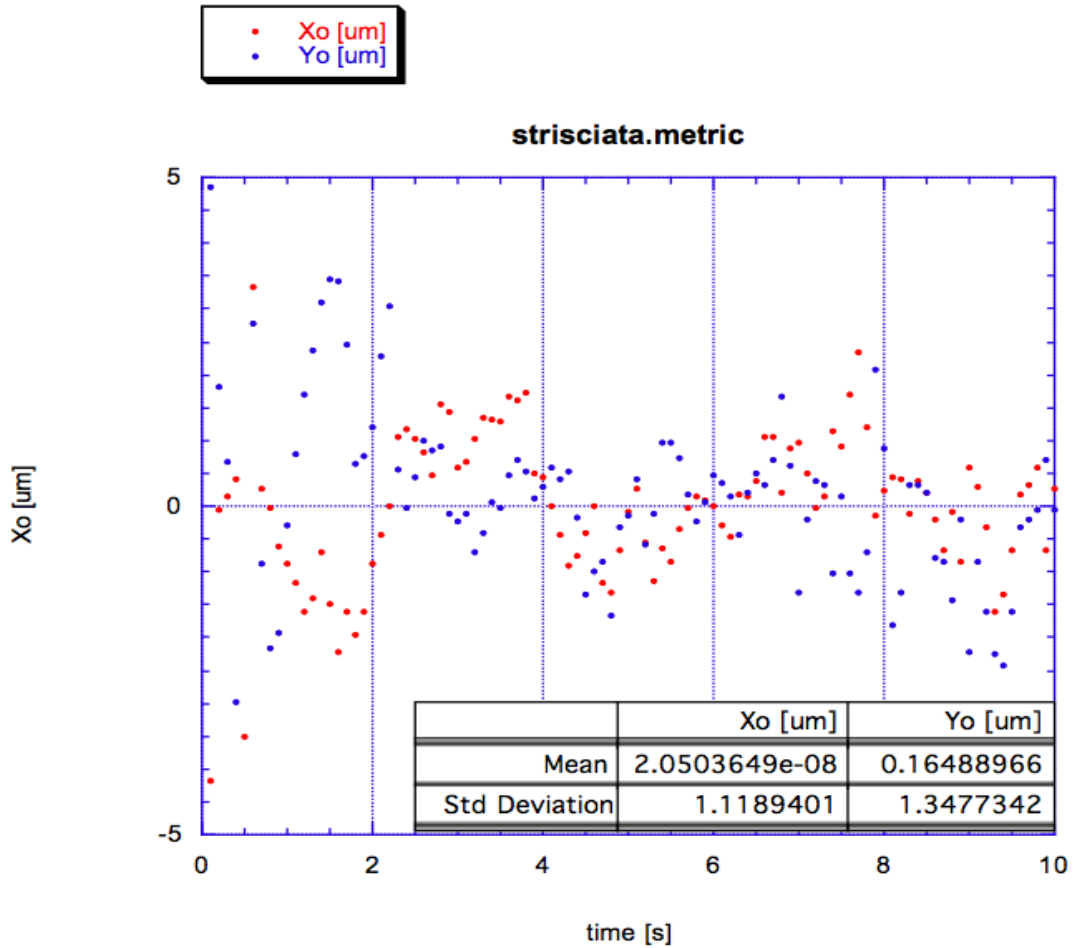
	ITM	ETM
Half Lever Length	32.8 m	5.3 m
Spot diameter (2w) at CCD, no distortion	4 mm	2 mm
Spot diameter (2w) at CCD, with thermal distortion	4.00348 mm	2.00011mm
	$8.7 \times 10^{-4}$	$1.1 \times 10^{-4}$
Movement at CCD with 1μradian movement	0.0656 mm	0.0106 mm

If the results shown above can be reproduced in a real time application, (not just in a post-analysis) the CCD camera has the precision needed for the Advanced LIGO.

We used a new camera, a GC650, to make this new test and analyzed the behavior of the system by taking streams of images with various delays. Then, we fitted 100 sequential images taken with a GC650 by using the small collimator and a 2m lever arm.

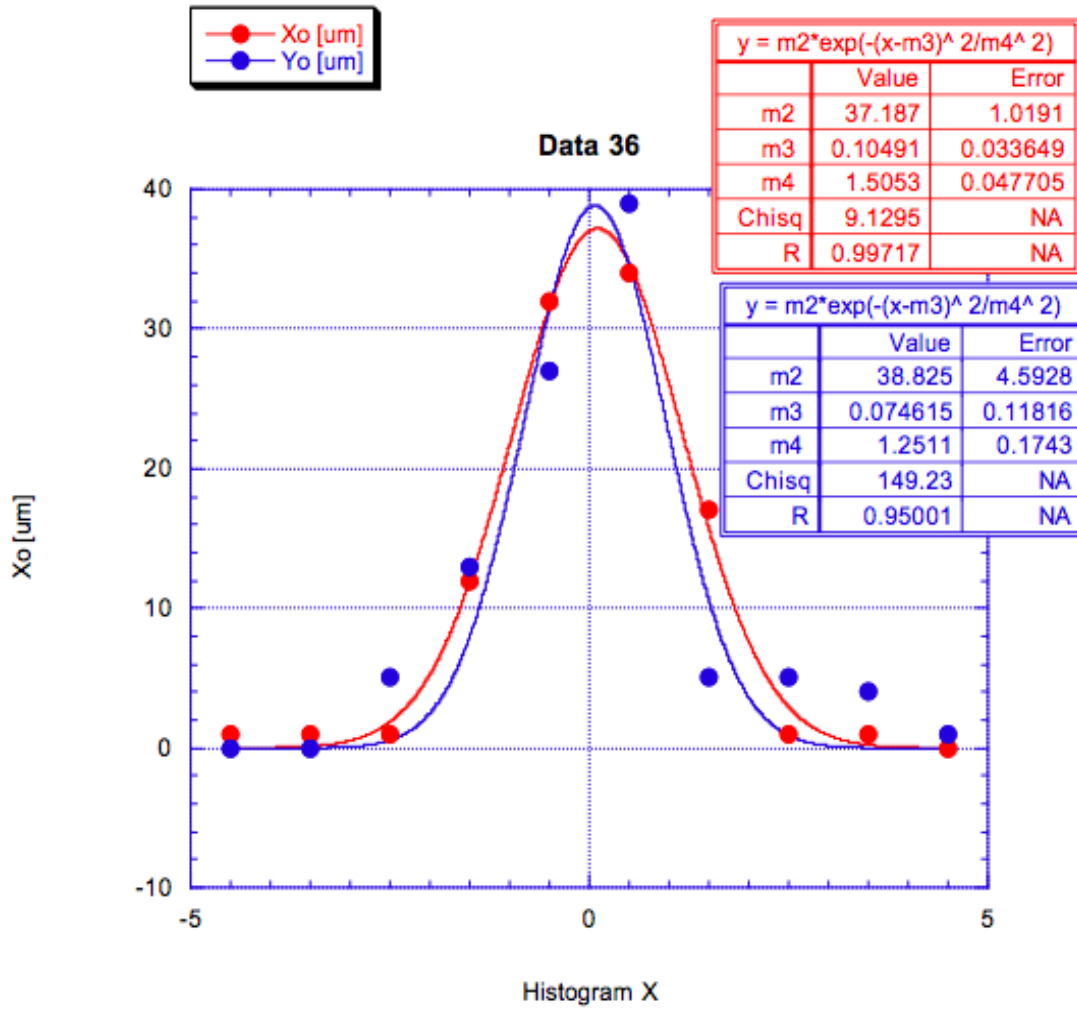
All fit positions were subtracted from average to compare results around zero and to study the stability of the algorithm. We repeated the experiment with a stable fiber and then with a vibrating fiber.

Fits with quiet fiber



The system does not show a significant short term drift, which is much better than what we observed under the same conditions with the QPD system.

Below are the distribution of the samples:



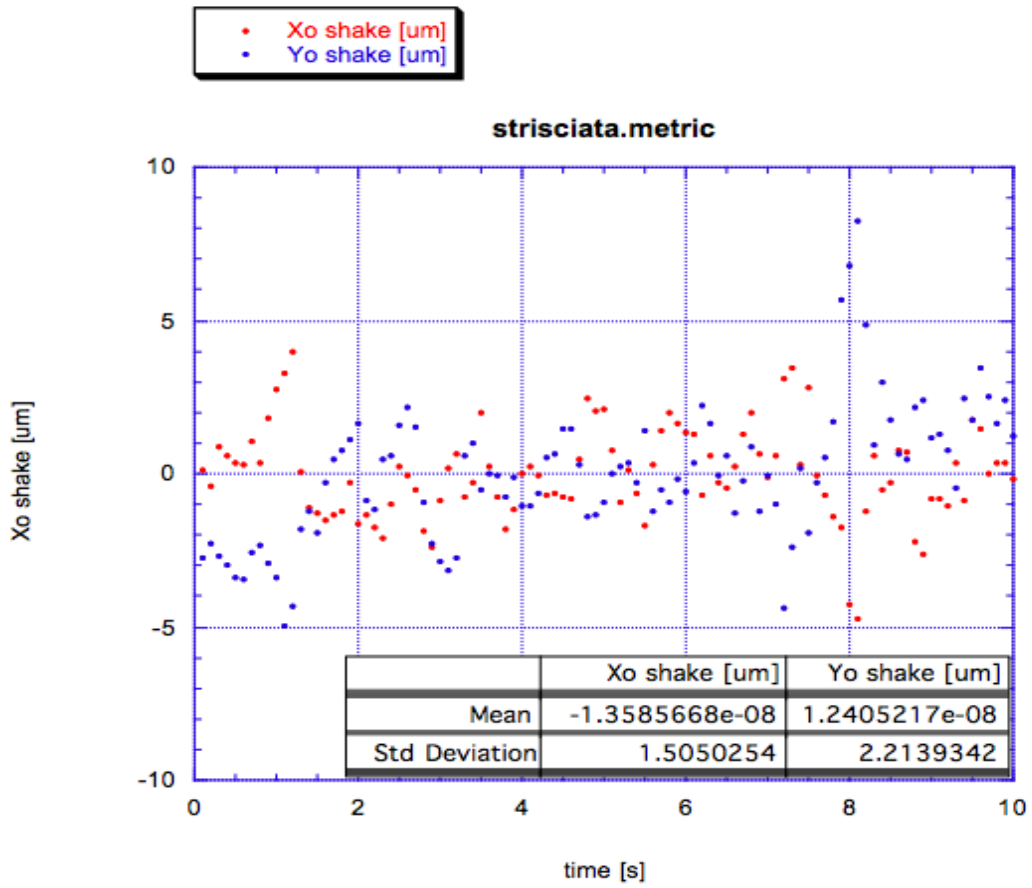
We noticed that we have a nice gaussian distribution. This means that by averaging the measured values over multiples samples, we can increase the statistical confidence of the positioning.

Then, we analyzed what happens if we shake the fiber.

The fiber is strain relieved at the connection with the collimator using a clamp. It is strain relieved near the laser as well, the fiber is shake in between the clamps.

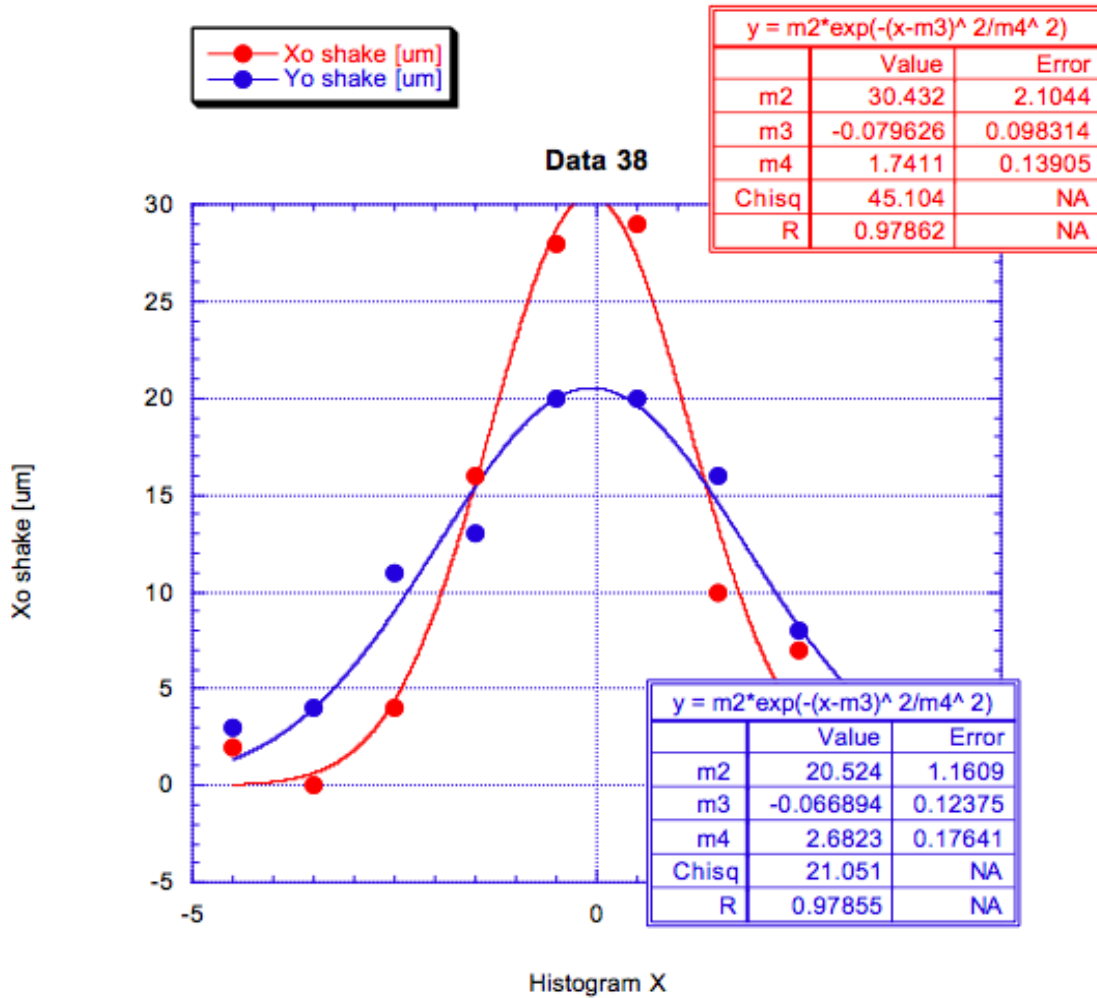


Fits with shaken fiber, almost double sigma



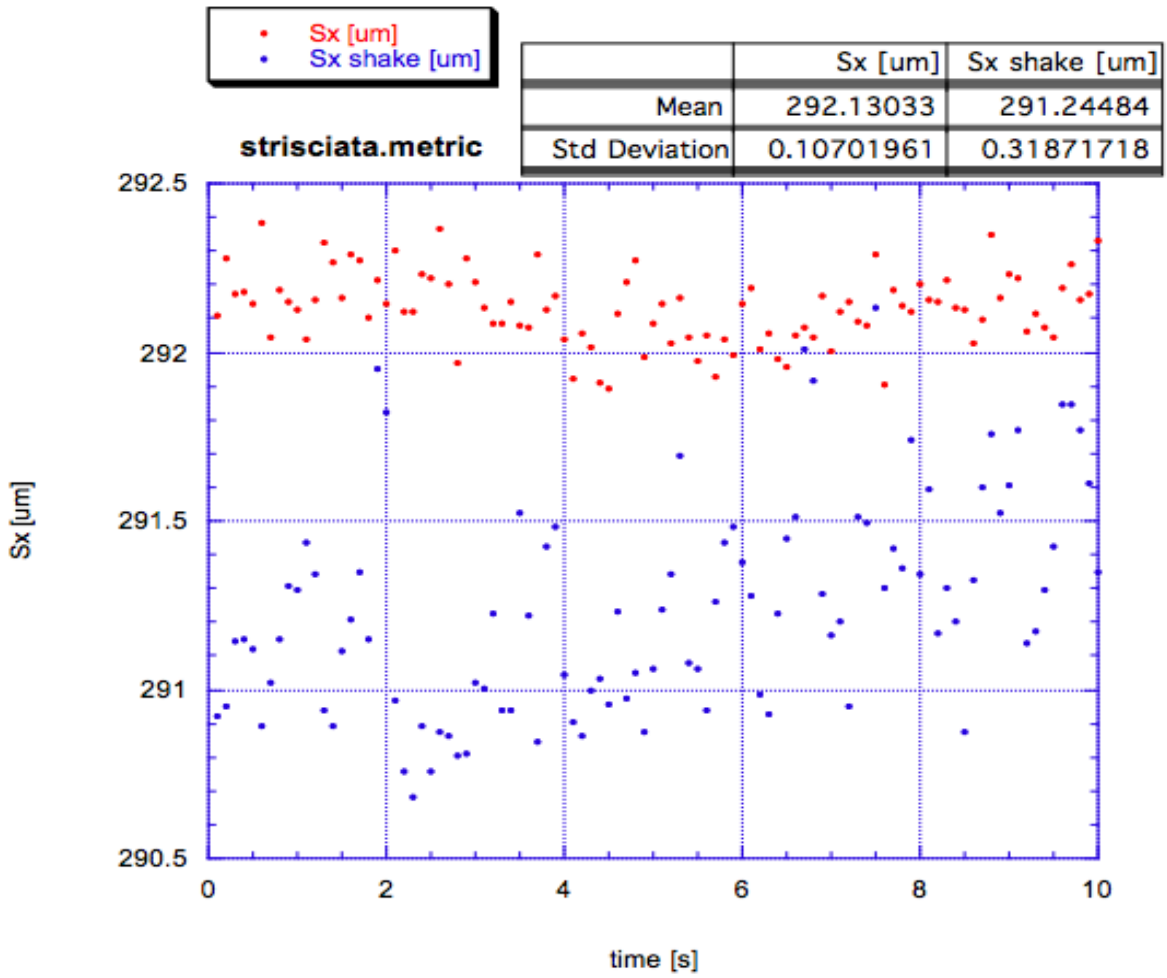
We can also see some change in the histogram:

In fact, the samples are more spread out around the center.



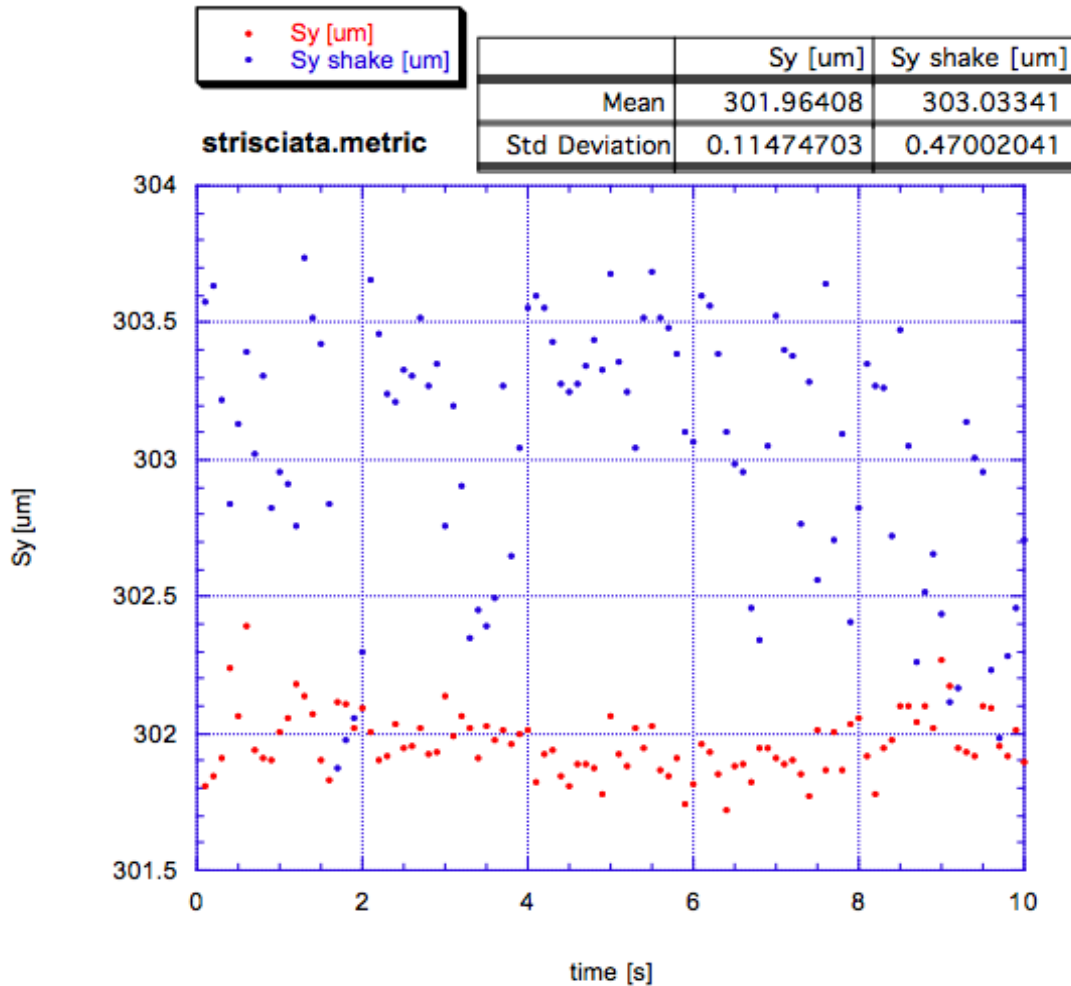
Then, we analyzed the precision with which we can determine the width of the laser spot, comparing the quiet and the “shaken” setup :

width in X quiet and shaken, error  $3.6 \times 10^{-4}$  single shot quiet,  $1.09 \times 10^{-3}$  with shaken fiber.



Averaging over ten images would give us the needed precision to monitor the thermal distortion.

width in Y quiet and shaken, , error  $3.8 \times 10^{-4}$  single shot quiet,  $1.55 \times 10^{-3}$  with shaken fiber.



The error with the shaken fiber is 3 to 4 times the error with the stable fiber.

### 2.3 Wobbling fiber

During our work with the QPD, we discovered that the fiber has a microphonic effect: bending the fiber can cause a shift in position as much as 2.5 radian. Yelling at the fiber produced similar effects.

This is probably due by the cladding mode of the fiber. At first, we thought that we could get rid of this problem by just cropping the image around the center spot. However, from experimenting different setups, we discovered that in most cases the effect is visible even in the center spot of the beam as a change in amplitude. The effect is more likely to be related with the mono mode light.

- Is the vibration transferring some light from the core to the cladding modes?
- We are injecting too much light in the cladding mode of the fiber?

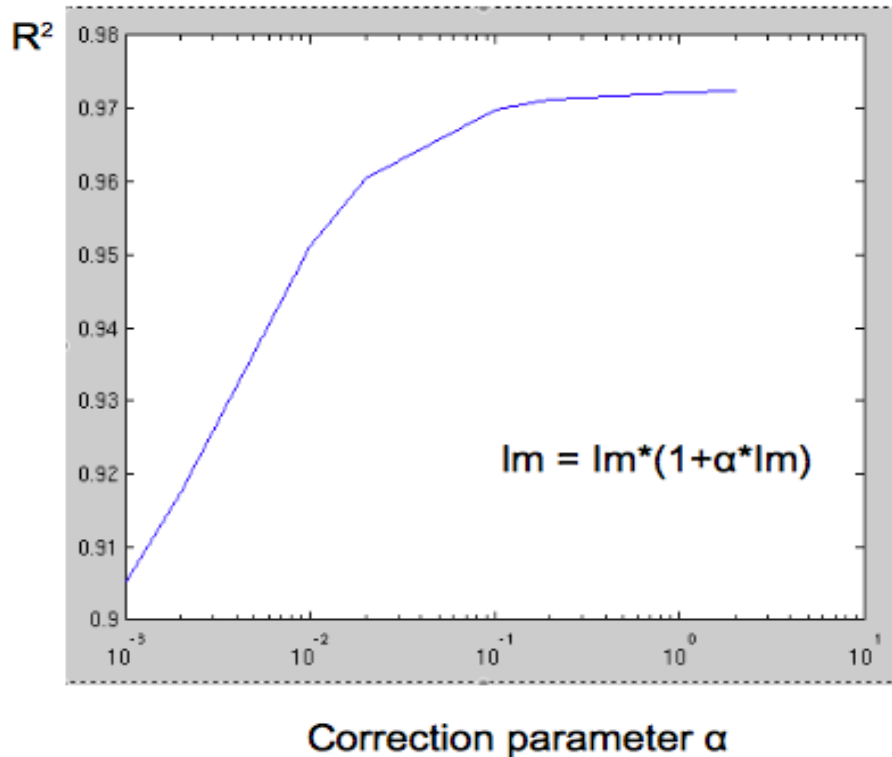
- We have a bad alignment of the optics?

## 2.4 Problems with a camera

We received a new camera (model GC1280) and encountered some problems using the camera.

### 2.4.1 Compensating

We had some doubt about the linearity of the camera in respect with luminosity so we tried to compensate luminosity to see if this affects the fit positively.



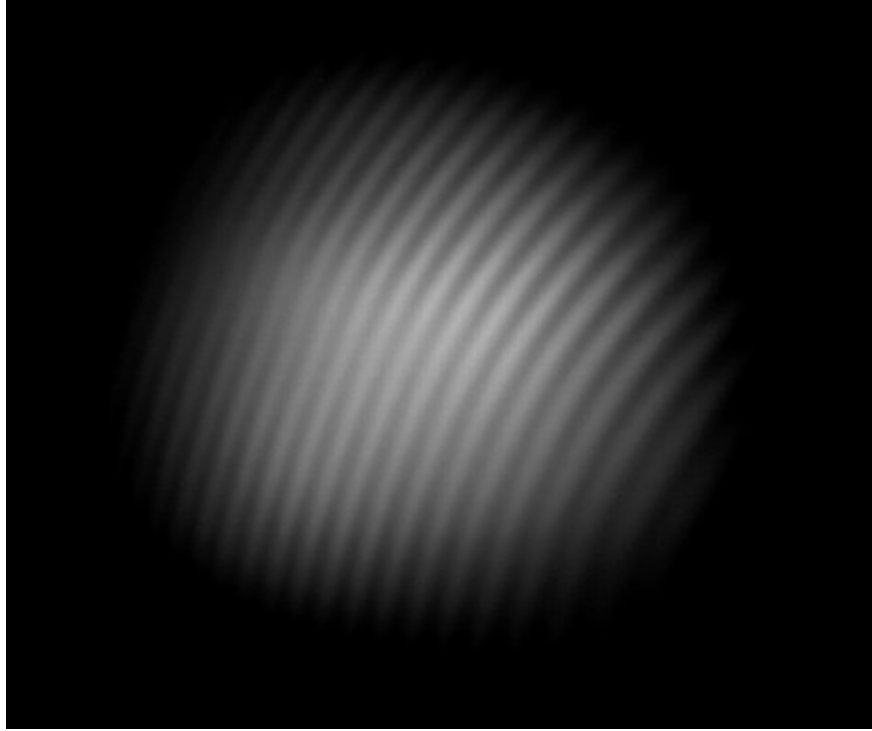
This does affect the fit, but we have to keep in mind that we can't determine an a-priori value for the correction parameter. Also, it seems to work only with the airy fit, which is slower than the gaussian fit.

The apparent non linearity may also be due to optical aberration effects. It is unlikely that the optics produce textbook Airy rings.

Doubts also remain about the behavior of the camera in the different regions of the CCD, but because we expect the laser beam to be centered before operation, this is not a big issue at the moment.

### 2.4.2 Interference pattern

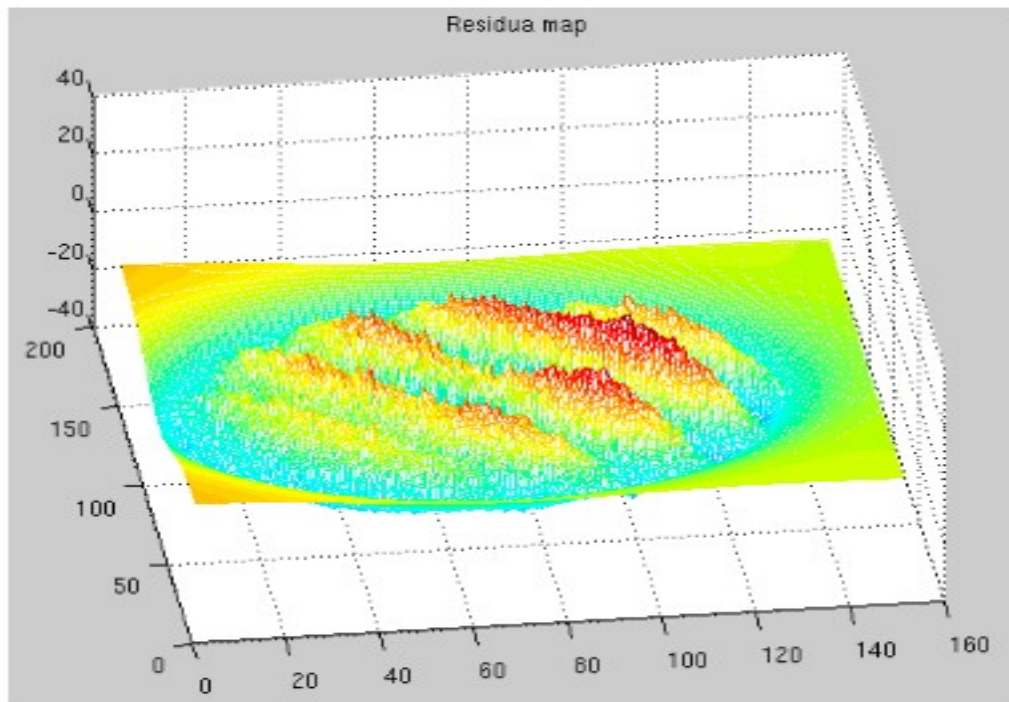
This is the interference pattern that some of the cameras have:



This probably happens because of the protective window glued in front of the CCD. If it is mounted with an angle, it creates the interference pattern shown in the picture above.

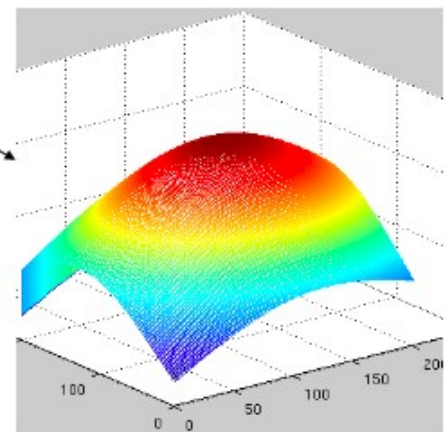
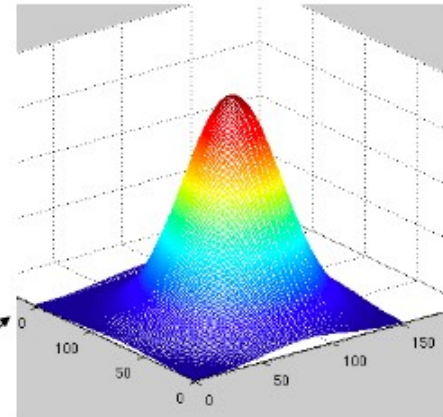
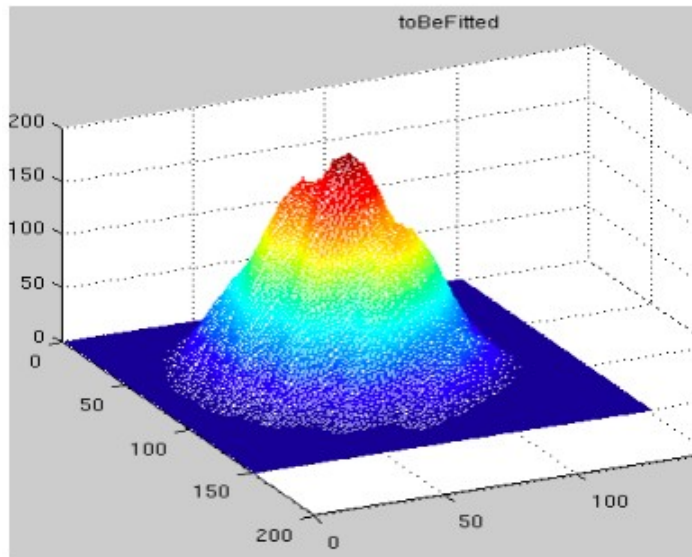
Since we cannot filter high frequency components of the image to remove the diffraction pattern (this will affect the real signal as well), we tried to fit the diffraction pattern itself.

The next picture is the residual map that we obtain from the gaussian fit.



However, sine is not an easy function to fit by itself and in this case, it is nearly impossible.

The diffraction pattern cannot be fitted by  $z = \sin(x + y)$  as we initially thought. The residual map shows a more complicated pattern of overlapping sine functions. If we go ahead and try to force the fit we obtain:



- In the best case the modulation is ignored by the fit.
- In the worst case the whole fit does not converge trying to fit the modulation.

We can try with more complex functions, or with an iterative procedure (over the fit itself) that fit independently the gaussian beam and the sine modulation (using the residual map). This adds a lot of complexity to the algorithm and thus affects its performance.

We also have to keep in mind that only one of the cameras that we used (the GC1280) showed such dramatic effect. The others have this effect, but in a really negligible amount.

We borrowed other cameras from the manufacturer and not all had this effect. If CCD camera are used, they should be selected to avoid this problem.

### 2.4.3 CCD vs CMOS

We compared many of cameras so we can make some kind of comparison between these two types. We started using CMOS cameras because they are less prone to saturate, at the end of the summer we switched to CCD cameras. In our work conditions, the difference between these two technologies is negligible and we have a better signal using CCD cameras just because they use newer circuits and have a better SNR.



## 2.5 Computational

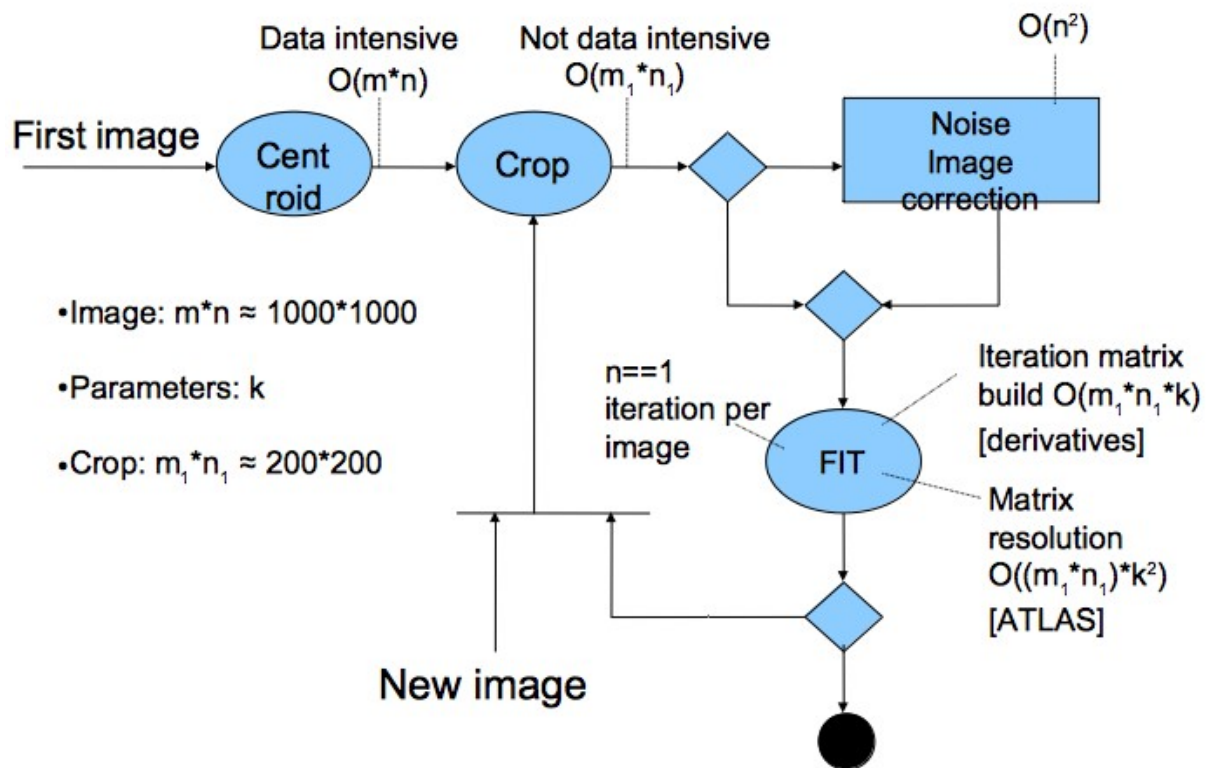
The final algorithm that we produced in C++ showed a really good performance. This is mainly because of the trust region that we introduced. The amount of data that we have to process after the cookie-cutter algorithm is small enough to fit in the cache of a commercial CPUs.

Using a macbook 2GHz, the average time to fit an image is between 100 and 200ms.

This performance can likely be improved with further work and using specialized processors.

We also used parallel libraries to solve the linear system created at each step. This increased performance immensely and we expect that every other optimization would have less impact on performance.

Here it is the activity diagram of the algorithm:



As we can see the centroid algorithm is used only to determine the initial parameters. We use the results from the previous fit to determine the parameters for the next one. We can keep the number of iteration per fit low (just one in most cases) and have a high frame rate. Another crucial point is the crop before the fit because it reduces the amount of data to be processed.

The noise reduction stage was usually skipped during our tests. It is not a computational intensive algorithm but it did not affect significantly the precision of the fit.

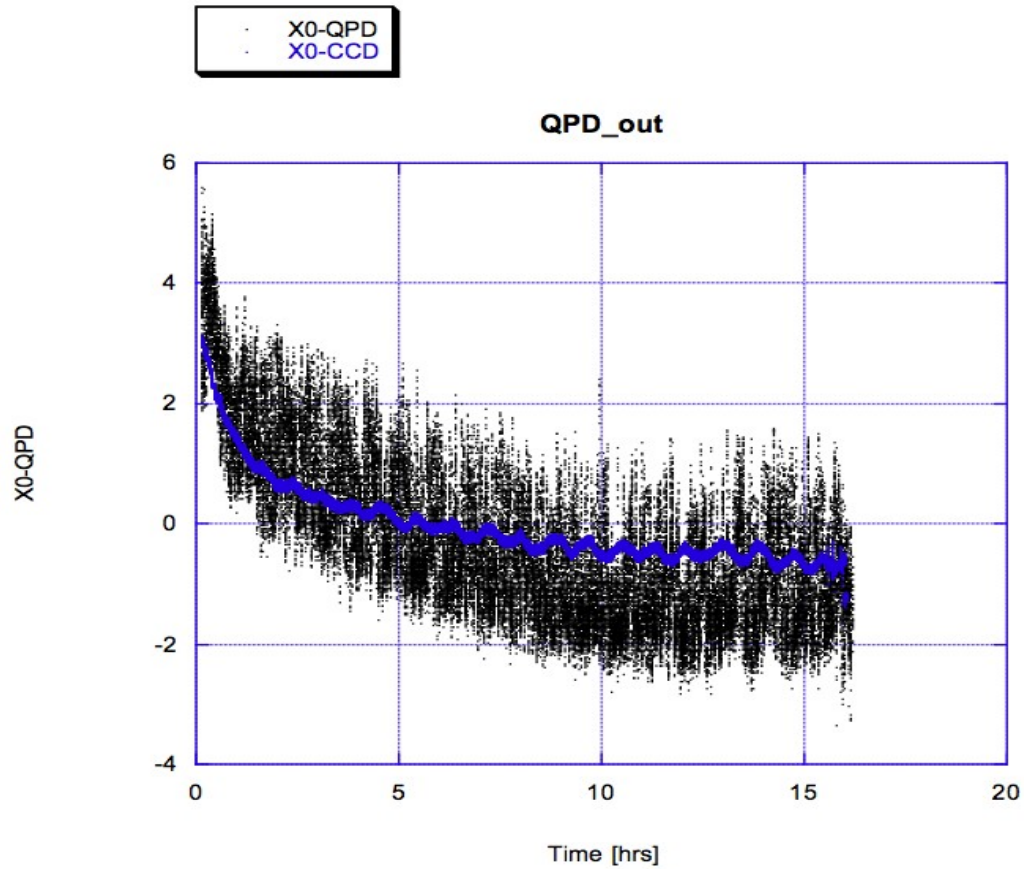
## 2.6 Comparison QPD vs CCD

We managed to make a comparison in real time of the two systems dividing the light on a cube beam splitter and reading it at the same time on a QPD and a CCD. The timing between the two plots is only approximate because acquisition is performed on two different computers:

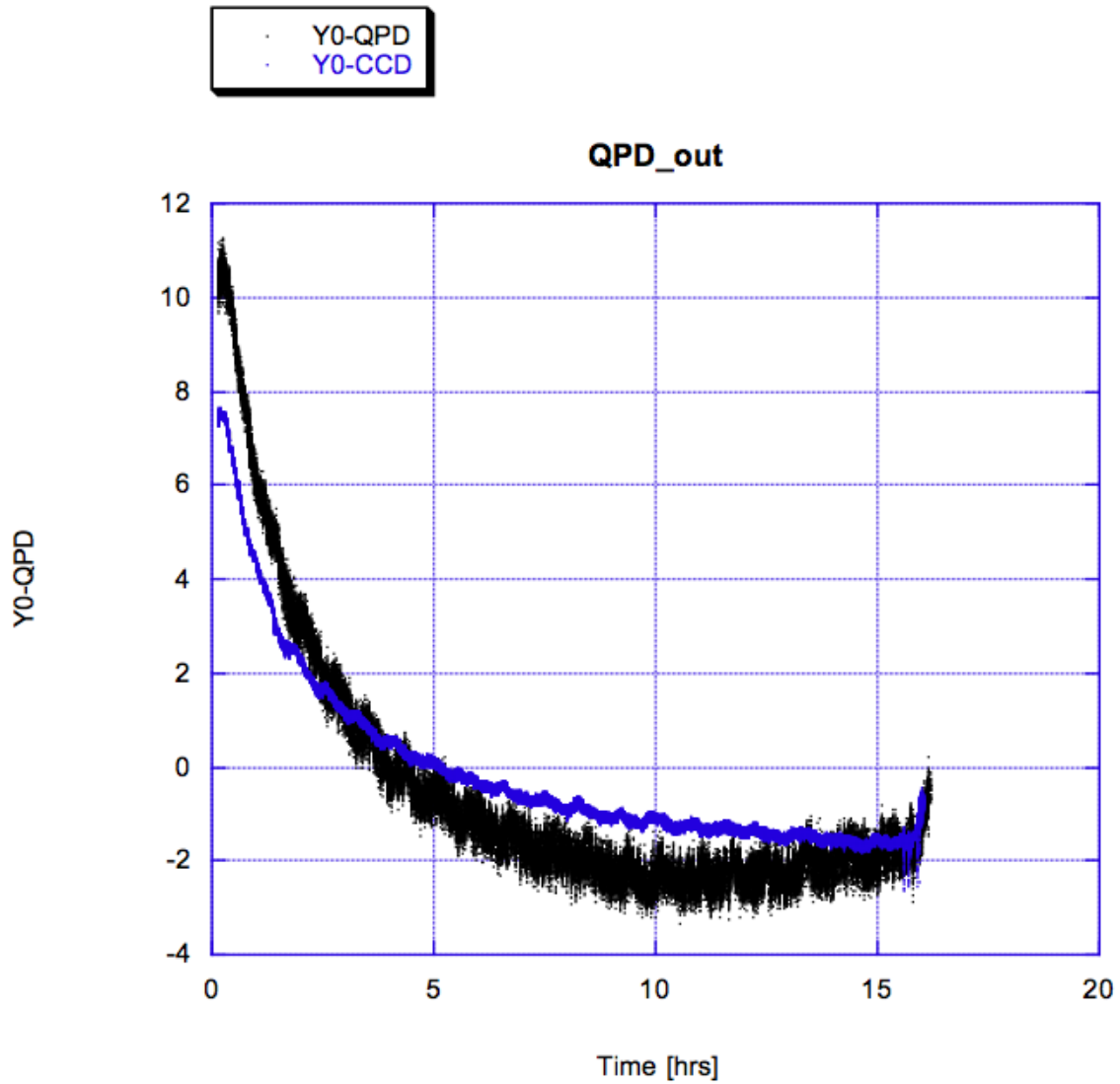
X position over time.

All the positions are relative to the average point, so every value is around zero.

All values are in microradian.

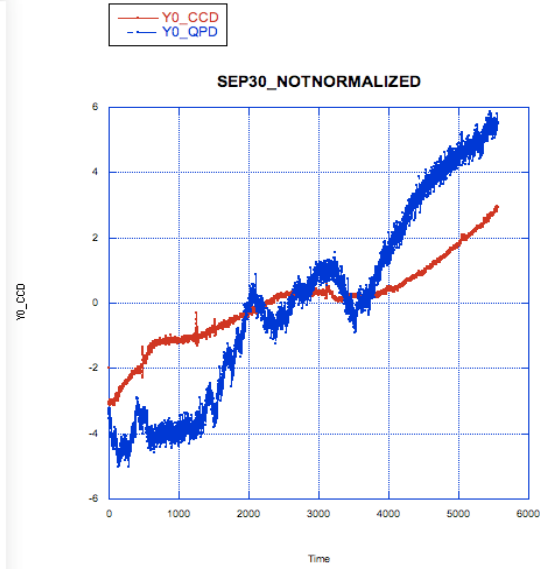
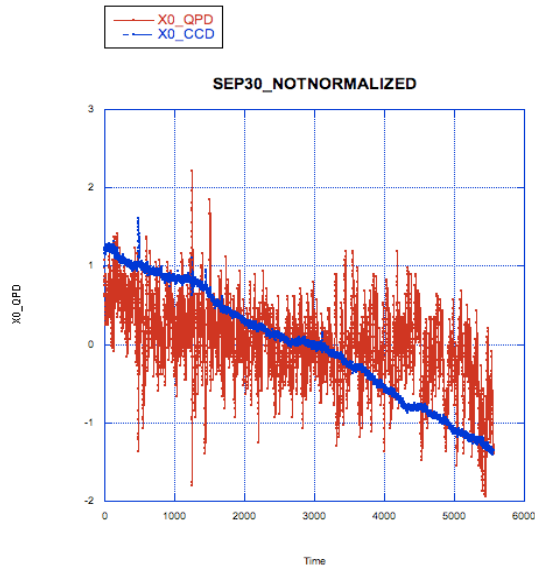


And Y position over time.



We can definitely see a correlation between the long term drift of the two systems.

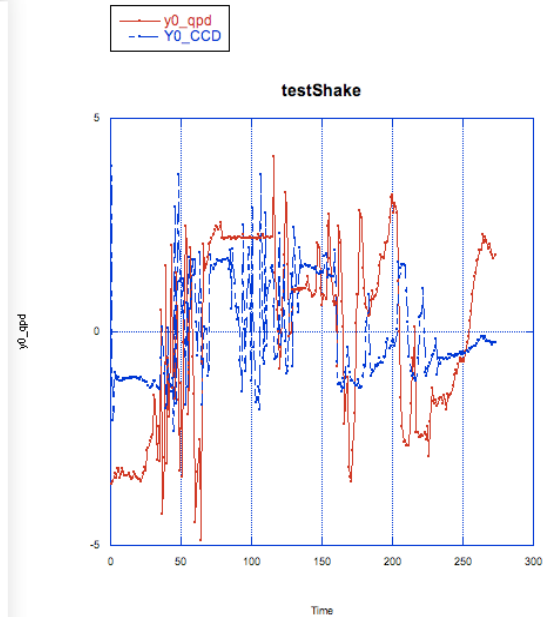
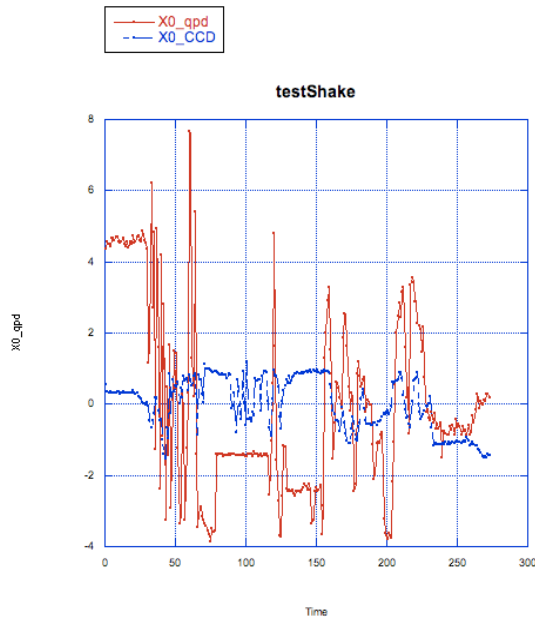
The waviness of the CCD image is attributed to the Air conditioning cycles warping the optical bench. The slow trend is probably thermal as well. The QPD signal shows much more noise, probably due to fluctuating Airy ring or cladding light arbitrarily cut by the outer edges of the QPD. Then we compared with the not-normalized QPD.



As expected we have a lot less correlation. By the way we tested the behavior of the system only for a relatively short time and we have to make more tests.

The normalized QPD was discarded because of the feedback signal used to normalize the output. With too low or too high amplitude of the signal the feedback stopped working and we have a spike in the output.

As a last test we tried to perform a comparison of the two systems while shaking the fiber.



During the test we tried to shake the fiber at various frequency. The effect is noticeable on the CCD, but still much less than the QPD.

Also, we noticed a really different behavior in the x and y axis for the CCD. This can be an indicator that the precision of the algorithm can be increased and that maybe we have a bad conditioning of the matrix in respect to the x axis calculation.

We had no time to study optimizations of the algorithm to further reduce its sensitivity to shaking effects. It appears though that normal acoustic vibrations are below threshold and affect little the stability of the fit's result.

### 3 Future Work

It is possible that the two systems can behave equivalently if we can get rid of noise from the QPD. At present though the CCD seems more reliable, less confusing to use even with the crude algorithm that we used. Also the CCD allows us to monitor the width of the laser beam with sufficient accuracy to track the residual thermal lensing after application of Thermal Compensation to the test masses.

Here is a list of tests and thing that needs to be done:

- Have a better understanding of the optics so that we can get rid of most of the diffraction and instability of the setup.
- Rewrite the code in a parallel fashion so that we can have the highest frame-rate. At the moment only the matrix-resolution (which is the most computing intensive) use parallel libraries but we can substantially increase performance.
- Use a new camera with better resolution and less noise.
- Study the conditioning of the matrix that we use for the Gauss-Newton algorithm so we can be sure that the algorithm itself does not introduce small drifts.
- Use an external normalization box for the QPD.
- Make sure that the QPD does not pick up too much noise because the sampling rate of labView is too low (Hopefully don't use labView at all).
- We can increase the precision of the fitting algorithm just using double precision (16bit image). Small portions of the code needs to be rewritten.
- After that we have acquired a really good understanding of the optics we can try to use better functions to fit the image and make a strict comparison of the different methods. Probably the gaussian fit that we produced will remain the best choice. A laser beam without diffraction has a gaussian shape and that is he reason why the gaussian fit would give us the best results.
- Also we never tested the program in the real site and with the optics that they currently use.