

# ABISM

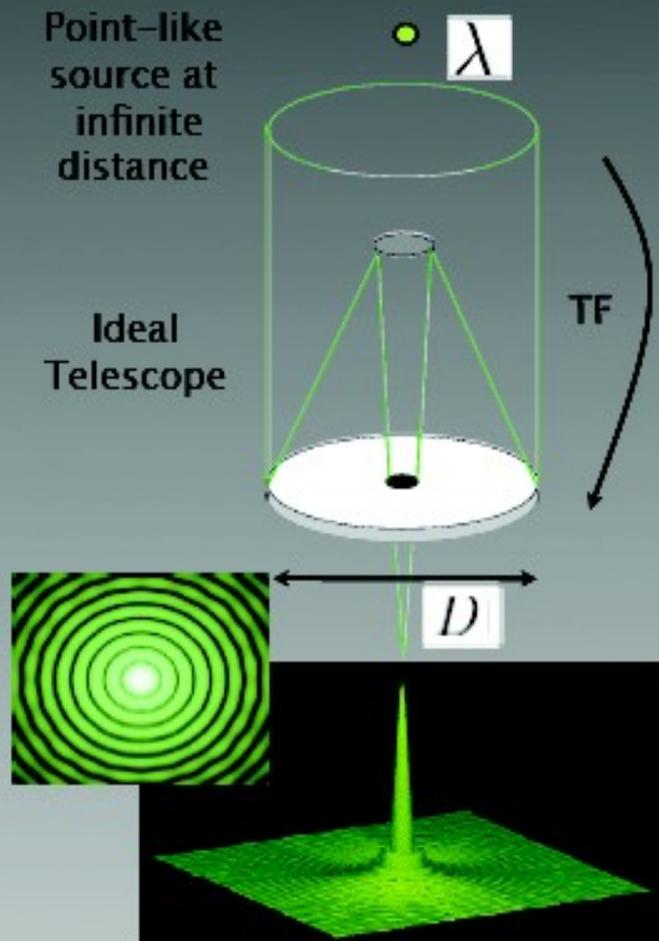


## Adaptive Background Interactive Strehl Meter

Julien Girard, Leonardo Vanzi, Martin Tourehenerboauf

# ABISM

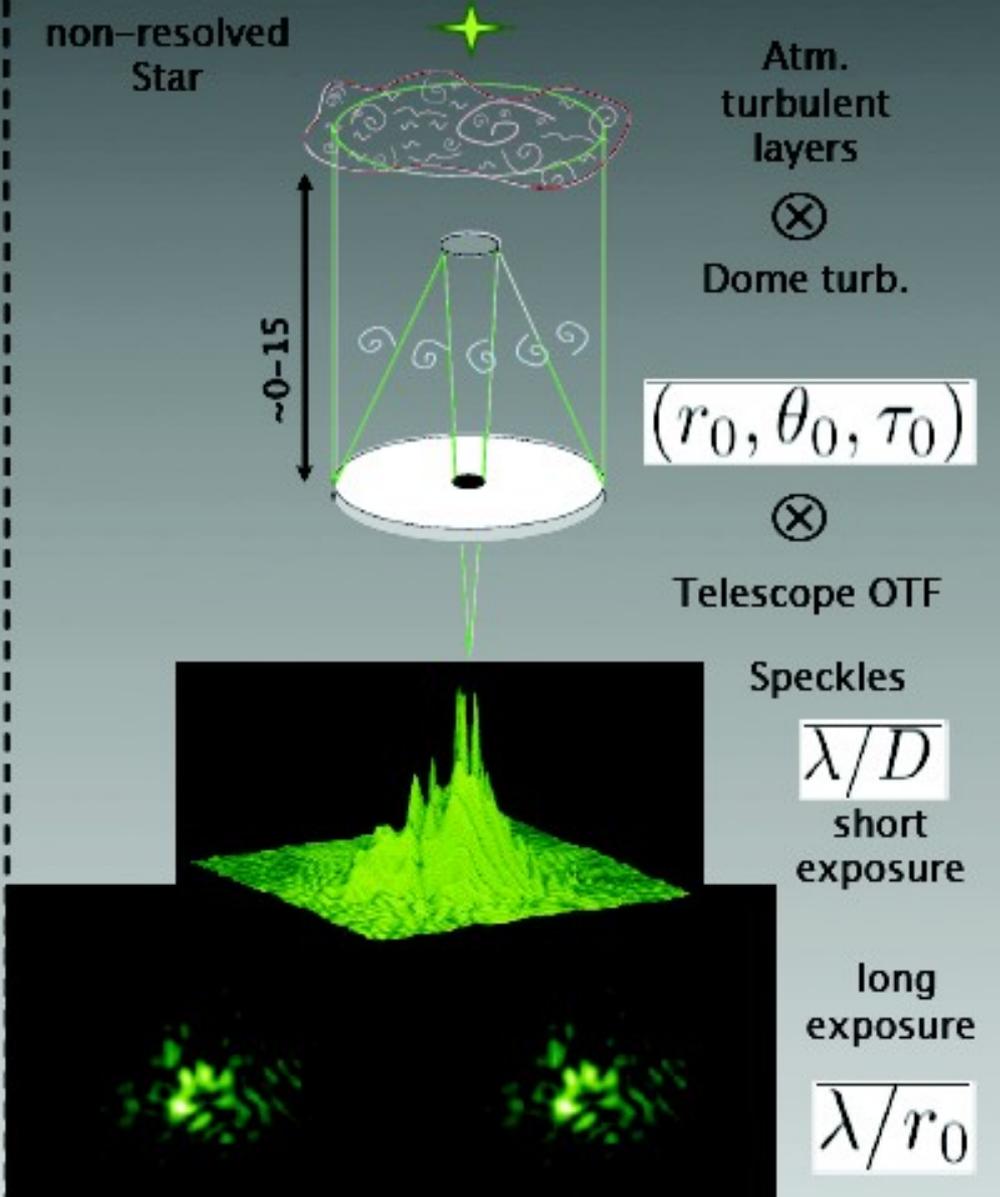
## DIFRACTION



PSF = Airy function

Resolution  $\sim \frac{\lambda}{D}$

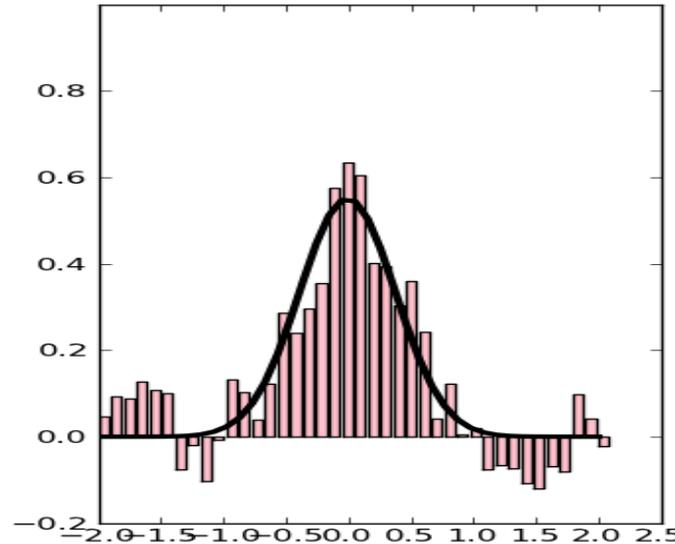
## Imaging through atmospheric turbulence



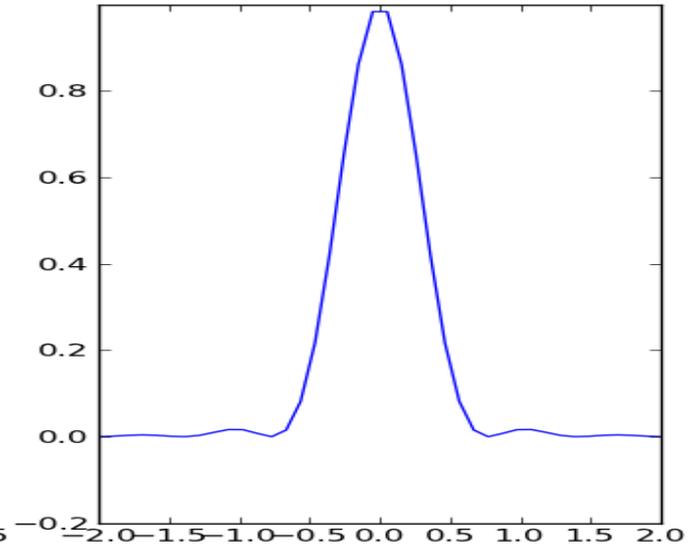
# ABISM

The **Strehl** is a measure of the **quality** of an image used in **Adaptive Optic**.

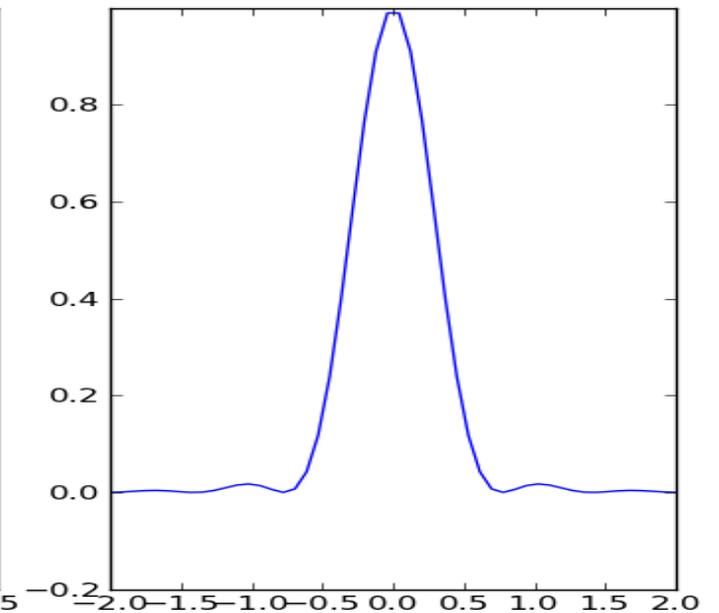
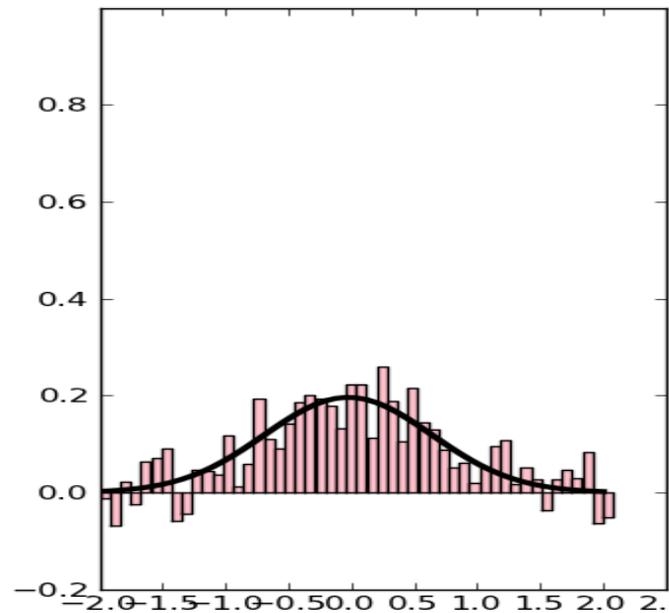
It is comparable to the **Encircled Energy** Or the **FWHM**



True Image



Diffraction Pattern



# Introduction

- We want to create a Strehl Meter
  - Easy to use (for visitors)
  - Independant
  - Fast
  - Robust (when it's 5am )
  - Can be installed on Paranal computers (old python versions (no wx)
    - Won't make everything bug (no crazy, dangerous loop)
  - Visual
  - Opened (can tell me all what he knows and can be used by everyone)

# Summary

- 1/ Generalities
- 2/ Presentation of the Abism
- 3/ Photometry
- 4/ Difficulties

# How to measure the Strehl (1901)

$$S = \frac{I(\mathbf{x} = \mathbf{0})}{P(\mathbf{x} = \mathbf{0})}$$

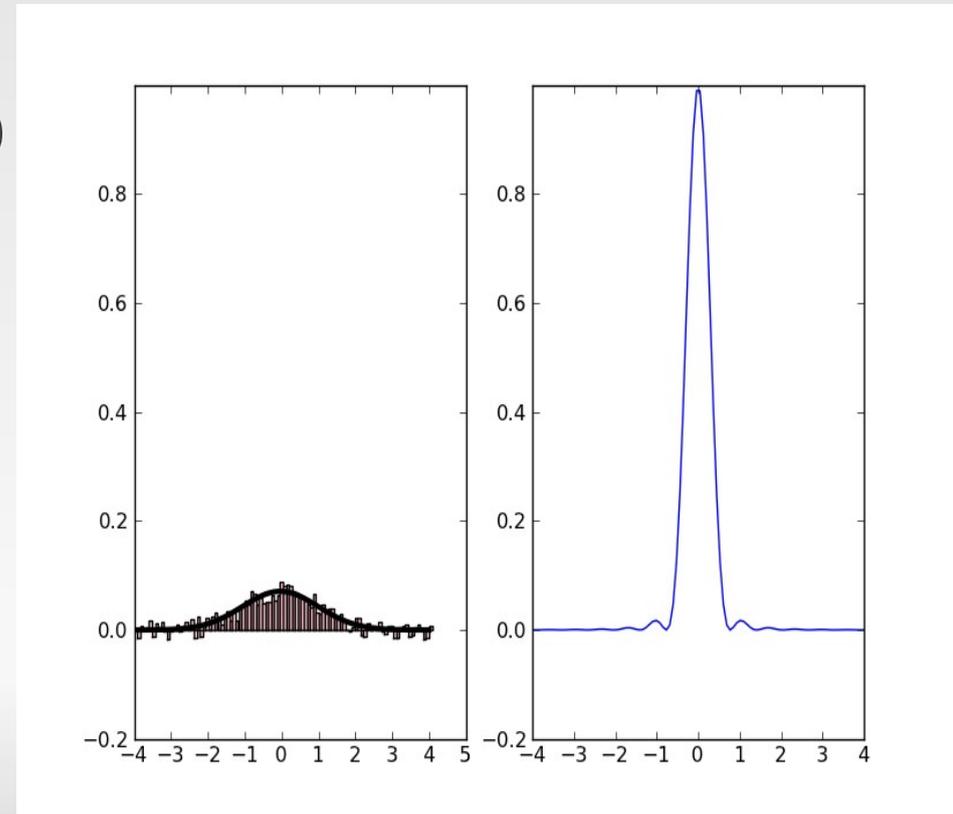
- 1/ Maximum
- 2/ Photometry
- 3/ Maximum of the diffraction pattern

$$S = \frac{\int OTF_{\text{PSF}}(\mathbf{u})d\mathbf{u}}{\int OTF_{\text{diff}}(\mathbf{u})d\mathbf{u}}$$

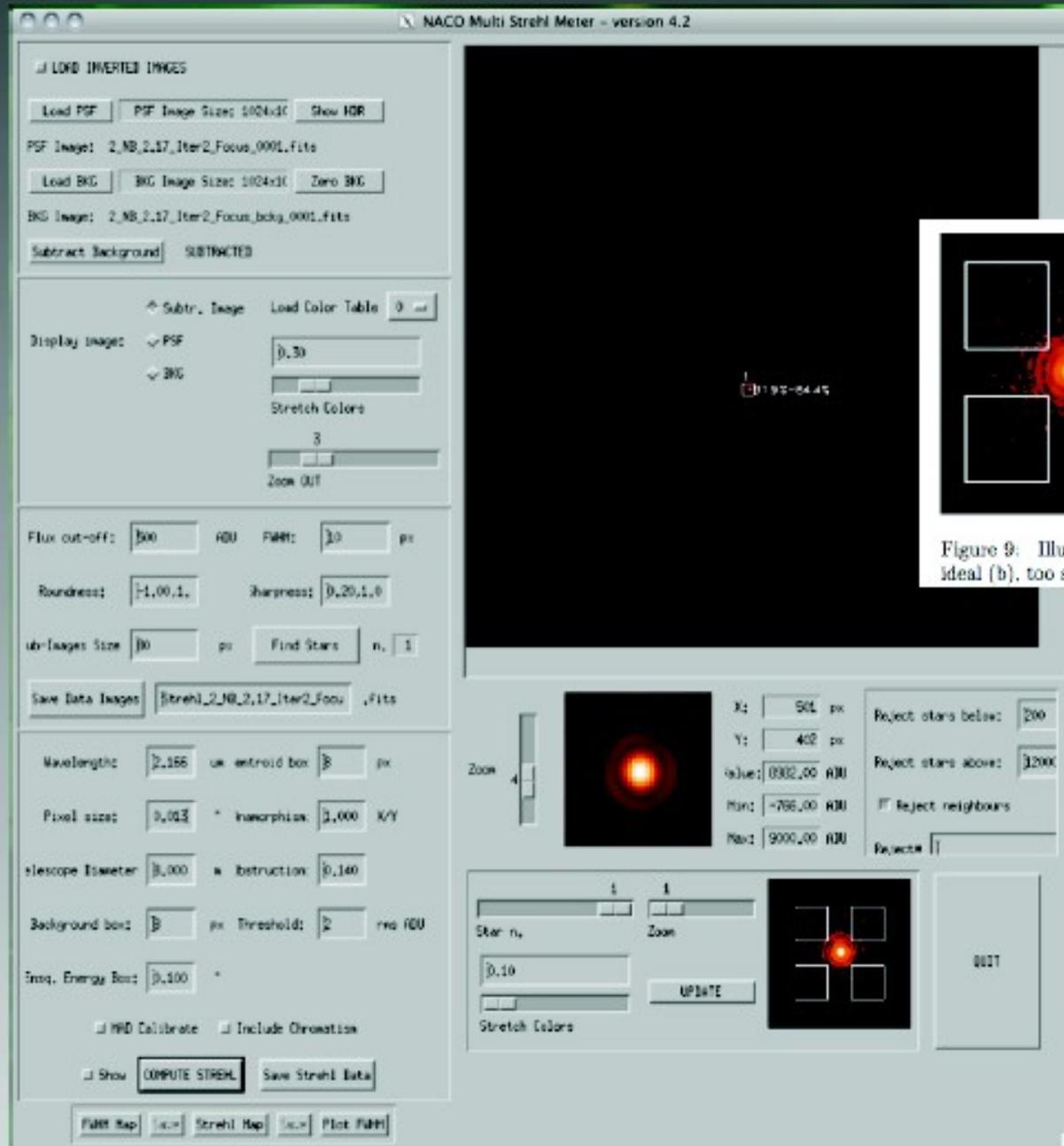
- 1/ Center
- 2/ Fourier transform
- 3/ Aperture
- 4/ Integration

# Generalities

- When the Strehl is too low, better to talk about encircled energy or FWHM. (Interferometry, spectroscopy)
- The Strehl is very dependant on the way to measure it (Lewis 2004).
- Need to know
  - wavelenght
  - diameter & obstruction
  - pixel scale



# ESO current Strhel meter MSM (good but not user friendly)



Background boxes

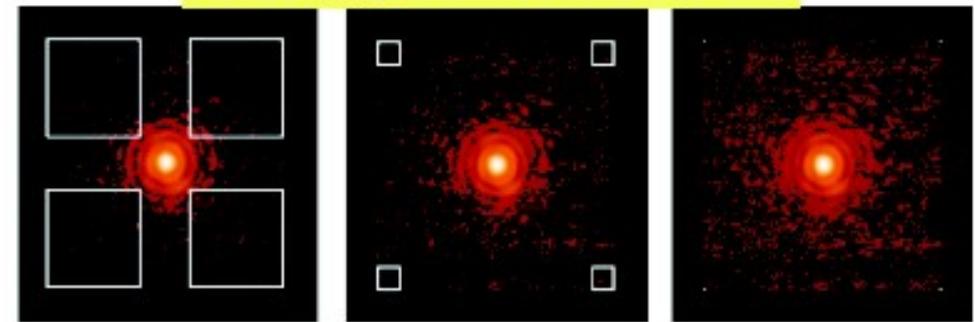
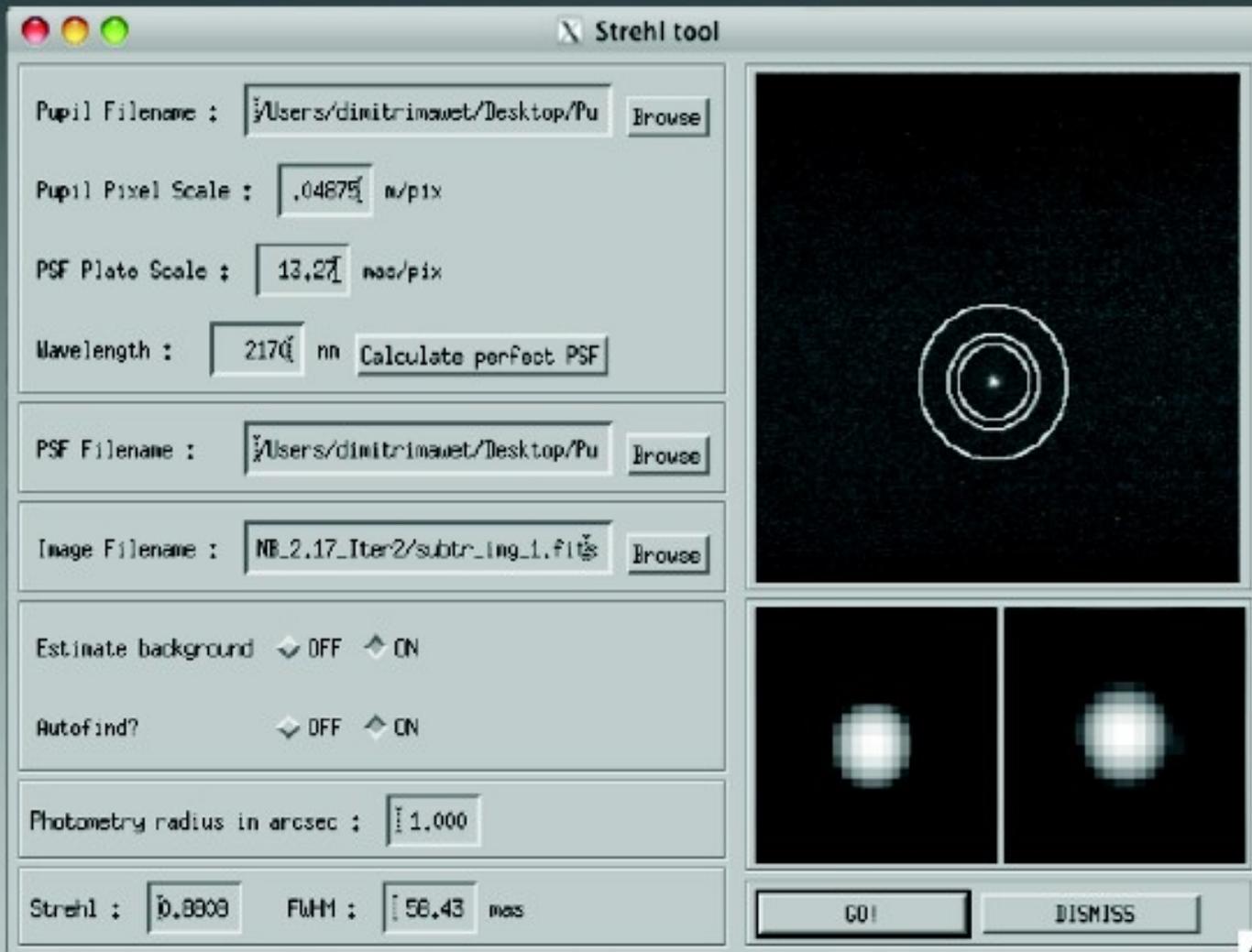


Figure 9: Illustration of three different background box sizes. From left to right: too large (a), ideal (b), too small (c).

MAD Multi Strehl Meter  
by E. Marchetti

Good for strehl maps

# An other Strehl meter goog but not user friendly either

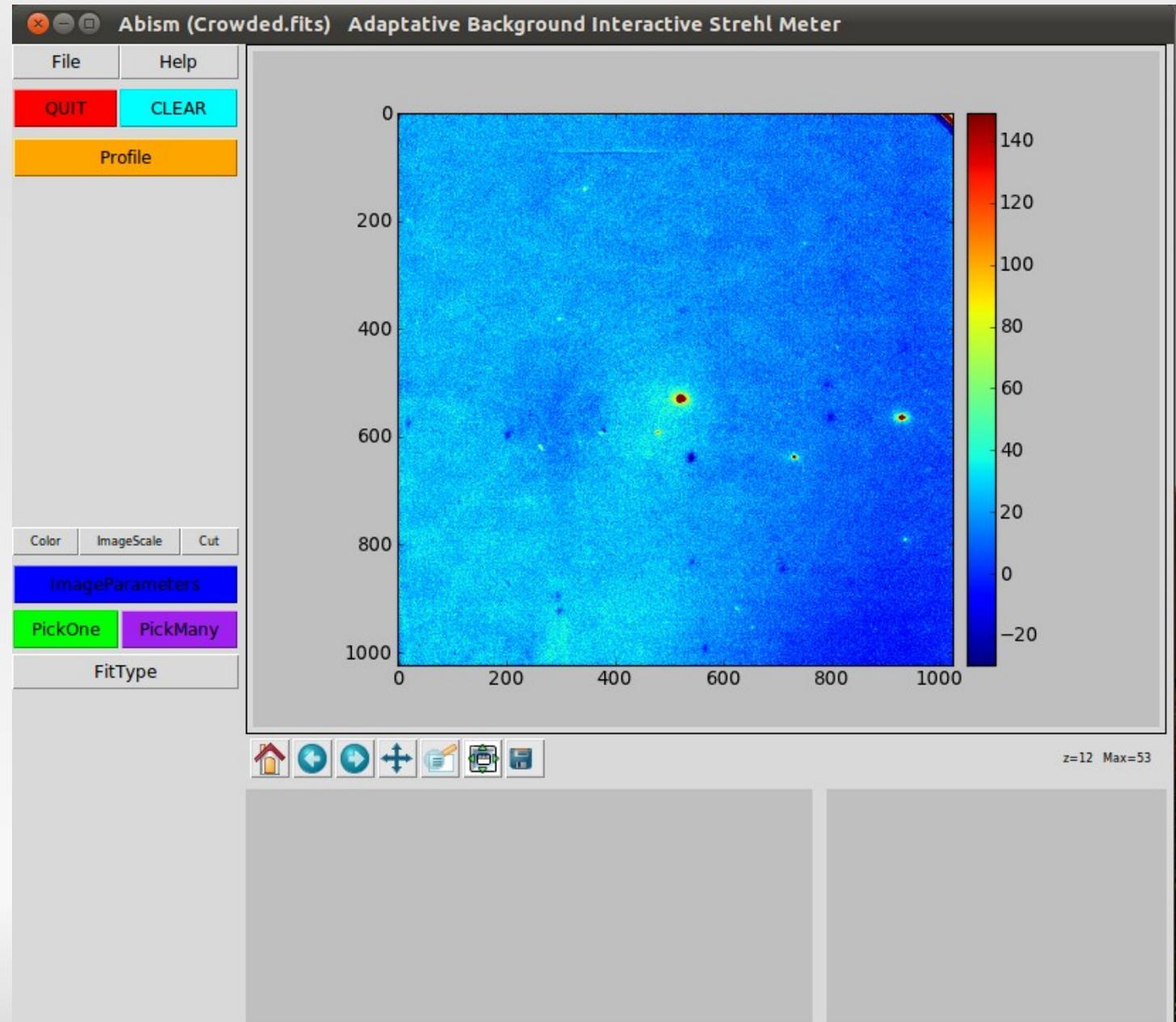


M. Van Dam

accurate but slow..  
adapted to Keck's complex  
aperture

# Our Strehl Meter

Just make a  
rectangle  
around your  
star



# Summary

- 1/ Generalities
- 2/ Presentation of the Abism
- 3/ Photometry
- 4/ Difficulties

# Presentation of ABISM

- Language : Python
- GUI package : Tkinter
- For : ESO : NaCo (can be used for every image.fits but won't read Header)
- Caracteristiques :
  - Can read data cube (by default take the last image)
  - Return an error and many easy way to compute the strehl
  - Return also : Strehl equivalent 2.2  $\mu\text{m}$  , FWHM, photometry, star center, background, Encircled energy (not implemented yet)

# Presentation of ABISM

- Can be opened by command line :
  - `python Abism.py [image.fits]`
  - Or by a script (Gasgano)
- A GUI allows the user to check and adjust parameters
- The user choose one or many stars and get his strehl.

**Gasgano**  
Eso software

**Abism.sh**  
Compatible with bash

**Abism.py**  
Order the text entrance  
(20 lines)

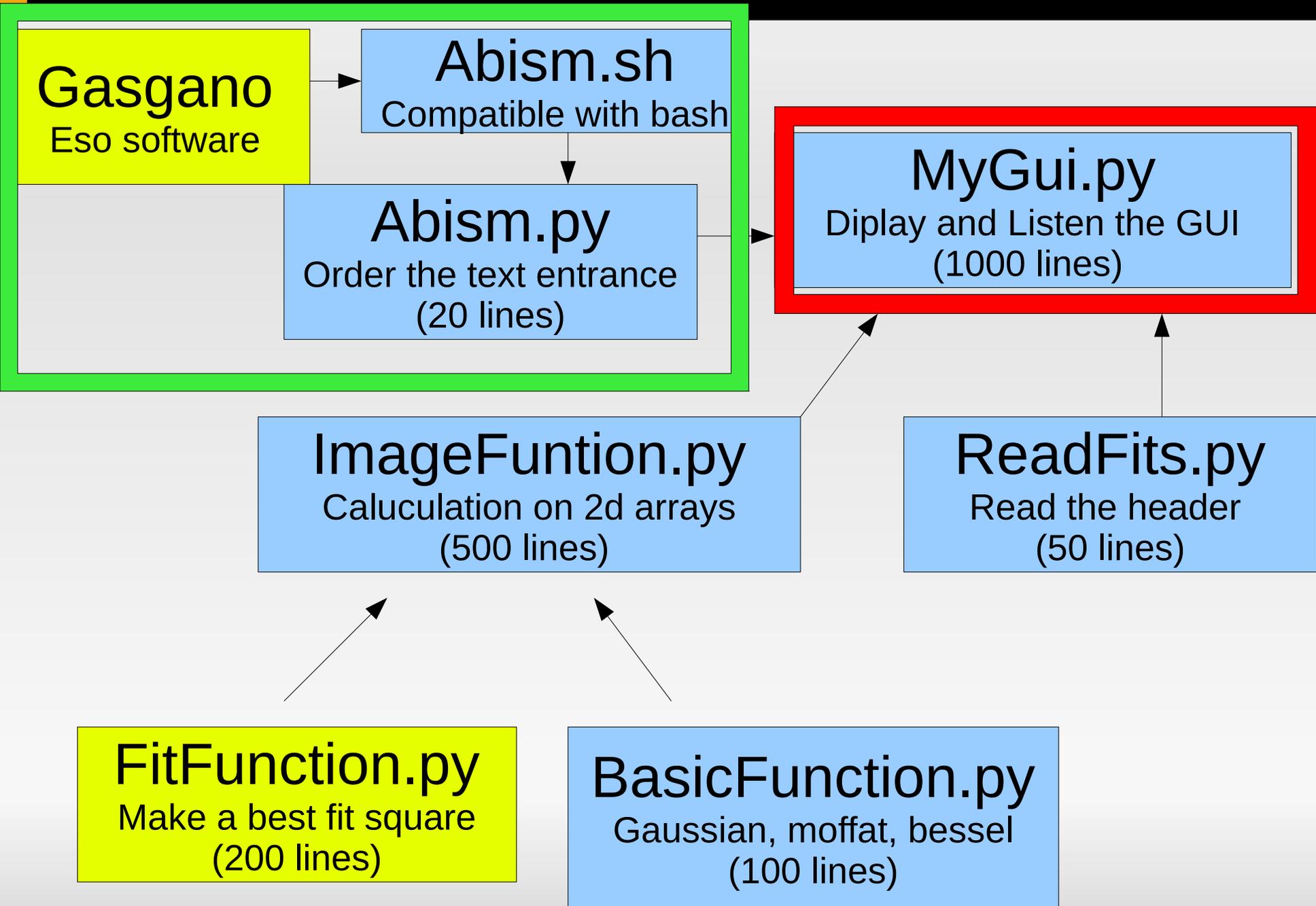
**MyGui.py**  
Diplay and Listen the GUI  
(1000 lines)

**ImageFuntion.py**  
Caluculation on 2d arrays  
(500 lines)

**ReadFits.py**  
Read the header  
(50 lines)

**FitFunction.py**  
Make a best fit square  
(200 lines)

**BasicFunction.py**  
Gaussian, moffat, bessel  
(100 lines)



# Summary

- 1/ Generalities
- 2/ Presentation of the Abism
- 3/ Photometry
- 4/ Difficulties

# Photometry

- How to measure an accurate photometry ?

## PSF fit

- What is the form ?
- Fall in a local optimum ?
- Give initial parameters
- Where to stop

## Aperture

- Where to stop ?
- Bad Pixels ?
- Circular ?
- How to interpolate ?
- What is the error ?

Astigmatism ? Binary ? Stepped Background ?

How to find the star ?

# Photometry

## Photometry : Our Solution

### 1/ Pre-Parameters

- a) Center : User defined
- b) FWHM : We parcurate the pixels

### 2/ Fit

- a) Type : User defined (But we'll do Gaussian → Moffat → Bessel )
- b) Local Optimum, to avoid it , we show the fit

### 3/ Aperture photometry

- a) Rectangle stop at 99% encircled energy
- b) Bad pixels destroyed by median filter

### 4/ Error

- a) Difference between my photometry and the fit that has been done
- b) Can be infered by S/N

#### PSF fit

- What is the form ?
- Fall in a local optimum ?
- Give initial parameters

#### Aperture

- Where to stop ?
- Bad Pixels ?
- Circular ?
- How to interplolate ?
- What is the error ?

Astigatism ? Binary ? Stepped Background ?

How to find the star ?

# Photometry (background)

- How to estimate background ?

## Previous PSF fit

## New, clever Aperture

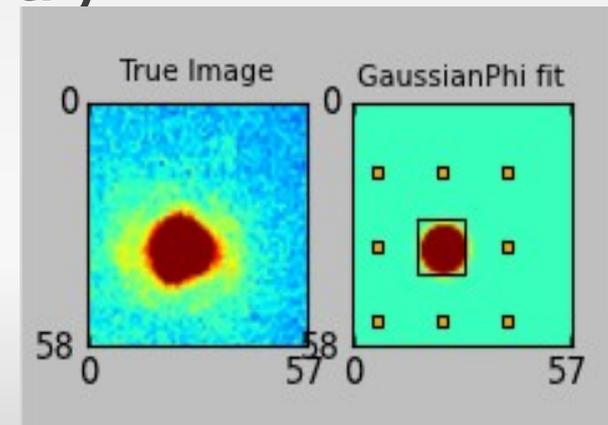
- Limits ?
- Degenerated ?
- Where ?
- How ?

- We do a 8 rectangles photometry

:) And we take the median

:) In the futur we aim to give

the slope of the background



# Summary

- 1/ Generalities
- 2/ Presentation of the Abism
- 3/ Photometry
- 4/ Difficulties

# Difficulties

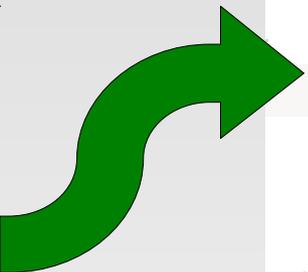
The screenshot displays a software interface with three main windows:

- Python command:** A terminal window showing the execution of a Python script. The output includes FITS header information for a file named 'Abism9', such as 'SIMPLE = T', 'DATE = '2010-08-02T02:03:10'', and 'OBJECT = 'Pistol\_star\_and\_neb''. It also shows the results of a 'diff-max' operation, including coordinates like (530, 521) and various photometric parameters.
- Abism (Crowded.fits):** A window displaying a 2D plot of a crowded field. The plot shows a color map with a color bar on the right ranging from 0 to 140. The plot axes are labeled from 0 to 400. Below the plot are several control buttons: 'File', 'Help', 'QUIT', 'CLEAR', 'Profile', 'Color', 'ImageScale', 'Cut', 'ImageParameters', 'PickOne', and 'PickMany'.
- Python command (smaller window):** A smaller terminal window showing the execution of a Python script with the following code:

```
print self.psf_fit[0][0]
print " This is for helping the programmer to debug"
```

The 'Run' button is highlighted with a mouse cursor.

# Difficulties



```
def Run(self,String):
    String = String.get("1.0",END)
    → exec String in globals(),locals() ←
def RunPython(self):
    window_run = Tk()
    window_run.title('Python command')
    #window_run.geometry("+0+0")
    frame=Frame(window_run)
    frame.pack()
    text_user = Text(frame)
    text_user.insert(INSERT, 'print self.psf_fit[0][0]')
    text_user.pack()
    text_user.focus_force()
    button=Button(frame,text="Run", command=lambda : self.Run(text_user))
    button.pack()
    window_run.mainloop()
    return
```

# Difficulties (Find the star)

- Histogram (brightest pixels)
- Near the center
- Iterative gravity center
- **Human eye** (Draw a rectangle)
  - Can change color, scale, (min,max)
  - Can subtract the image of the noise

# Difficulties

## Science

- Noise
  - Median Filter (when necessary)
- Error estimation
  - (Strehl is in %)
- Moffat
  - 2 degree of freedom
  - 99% energy

## Computer

- Compatibility
  - from  
`matplotlib.backends.backend_tkagg`  
`import FigureCanvasTkAgg as`  
`FigureCanvas`
- Data cube
  - automatically take the last
- Going fast with array
  - `od[d<0.25] = array[d<0.25]`

# Difficulties

## Science

- Binary
  - (background, photometry)
- Cut in log scale
- Aperture Fourier transform
  - (Analytical value but for Keck ?)

## Computer

- Wich click do I want
  - Color of the buttons
- Destroy, clear
- Passing Variable
  - (class, dictionary)

# Conclusion

- The error in strehl mesure (4%)
  - we can do an automatic method
    - phot, noise
    - read header
    - select star (ESO Eclipse)
- For low Strehl, better to speak about encircled energy, FWHM
- Return checks for the human.
  - psf fit, global variables ...

# Questions ?

The screenshot displays the Abism software interface. On the left, a FITS header is visible with the following parameters:

```
SIMPLE = T / f
BITPIX = -32 / n
NAXIS = 2 / n
NAXIS1 = 1024 / l
NAXIS2 = 1024 / l
EXTEND = T / FITS datas
DATE = '2010-08-02T02:03:10' / f
EXPTIME = 20. / I
AIRMASS = 1.004 / A
CTYPE1 = 'RA---TAN' / C
CTYPE2 = 'DEC--TAN' / C
CRVAL1 = 266.56411 / C
CRVAL2 = -28.83474 / C
CRPIX1 = 473.999999 / R
CRPIX2 = 499.410032 / R
ALARM = / A
CD1_1 = -7.55278E-06 / T
CD1_2 = 0. / T
CD2_1 = 0. / T
CD2_2 = 7.55278E-06 / T
ORIGIN = 'ESO-PARANAL' / R
TELESCOP = 'ESO-VLT-U4' / E
INSTRUME = 'NAOS+CONICA' / I
OBJECT = 'Pistol_star_and_neb' / T
print " This is for helping the programmer to
```

The main window shows a 2D image plot of a star field. A 'Profile' window is open, displaying the following parameters:

- Equivalent Strehl: 2.2 : 1.87
- GaussianPhi(530.3,520.5)
- Strehl: 59.1 +/- 2.6
- FWHM: 139 y= 34 (mas)
- Photometry: 160004.2
- Background: 38.0
- Intensity: 3460.2 (adu)

The 'Profile' window also includes buttons for 'SubtractBackground', 'Noise', '8Rects', 'Fit', 'Manual', 'InRectangle', 'None', 'Photometry', 'EncircledEnergy', 'Fit', 'Manual', and 'Close'. A checkbox for 'Anisoplanetism' is checked.

At the bottom, there are two smaller plots. The left one is a line plot showing the 'Real Profile' (black), 'Encircled Energy' (dashed), 'Fitted PSF' (purple), and 'Diffraction Pattern' (blue) against wavelength (490-560 nm). The right one shows a 'True Image' and a 'GaussianPhi fit' as a grid of squares.

HOW WOULD YOU CALCULATE A STEHL RATIO?