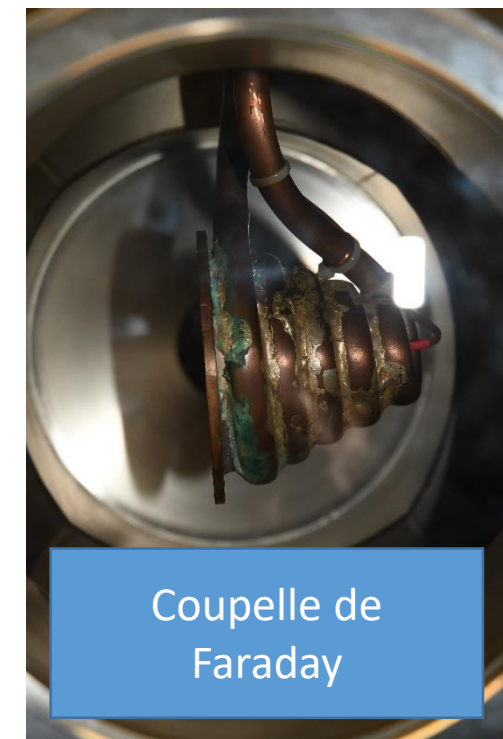
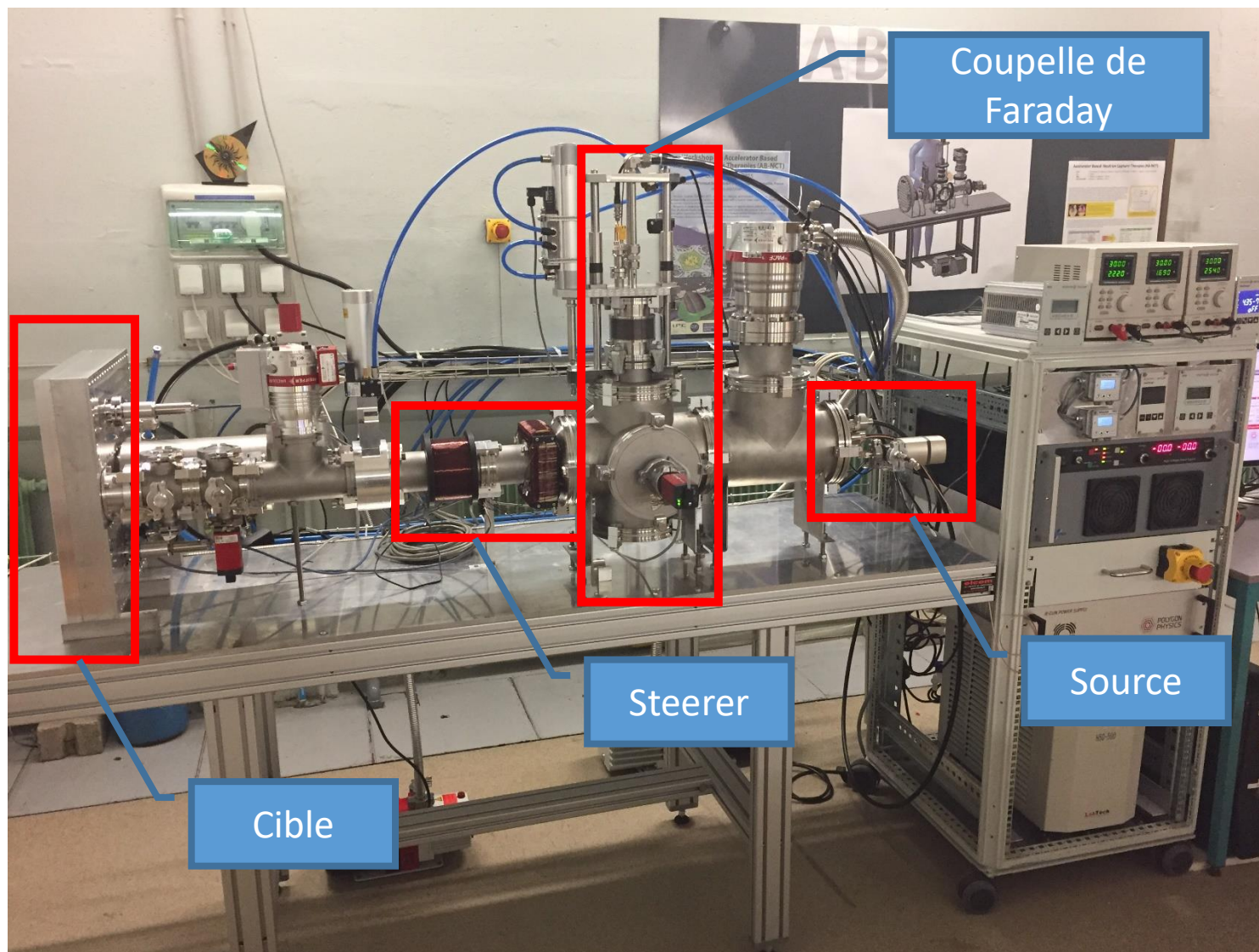


# Programme d'alignement de faisceau par caméra avec LabVIEW

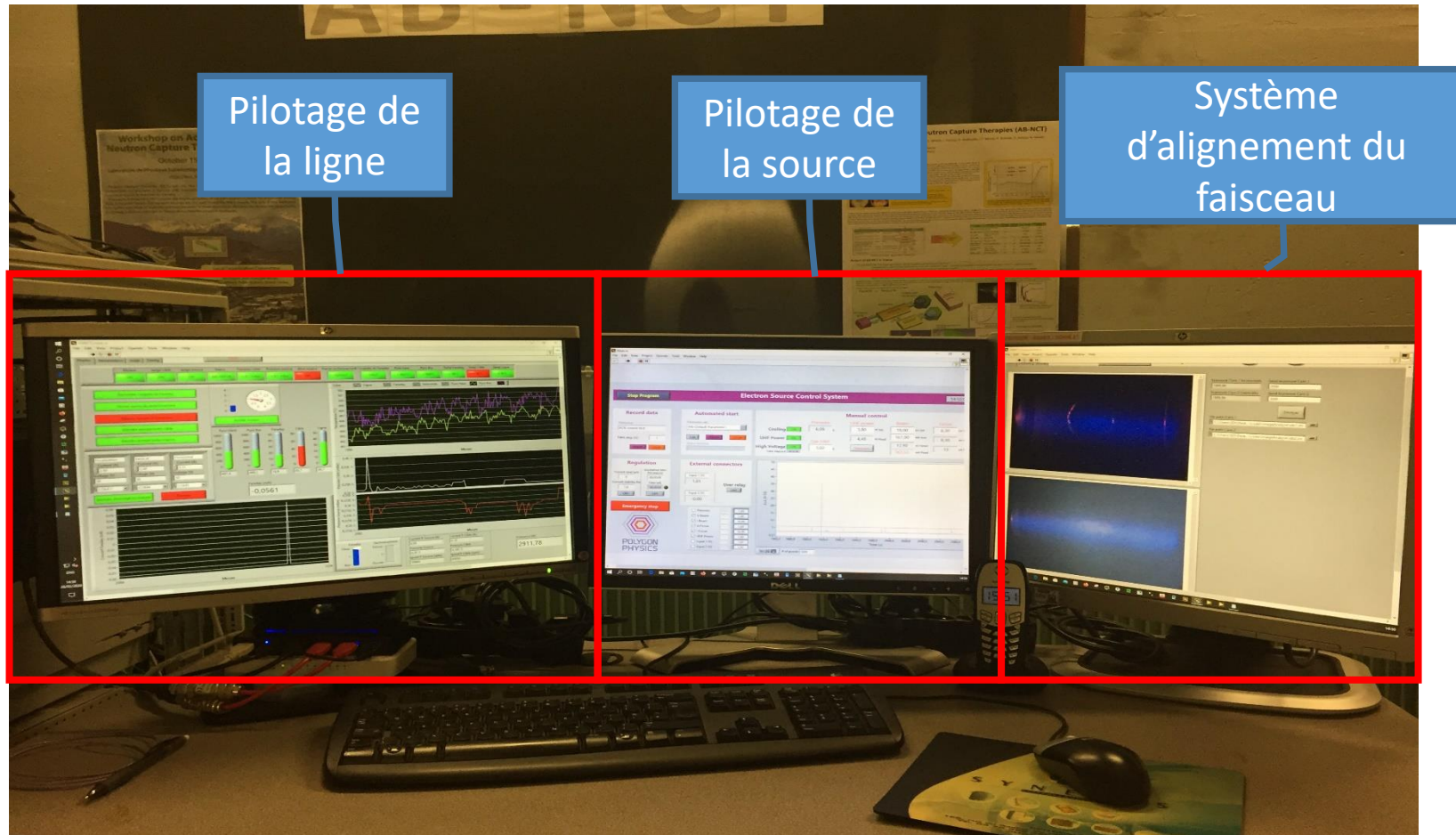
# Contexte AB-nCT (1/3)

- AB-nCT = Accelerator Based Neutron Capture Therapies
- Il s'agit d'un projet de physique médicale dédié à l'étude d'une thérapie anti-cancéreuse originale basée sur l'irradiation du patient par un flux de neutrons
- banc de test thermique pour caractériser les cibles

# Contexte AB-nCT (2/3)

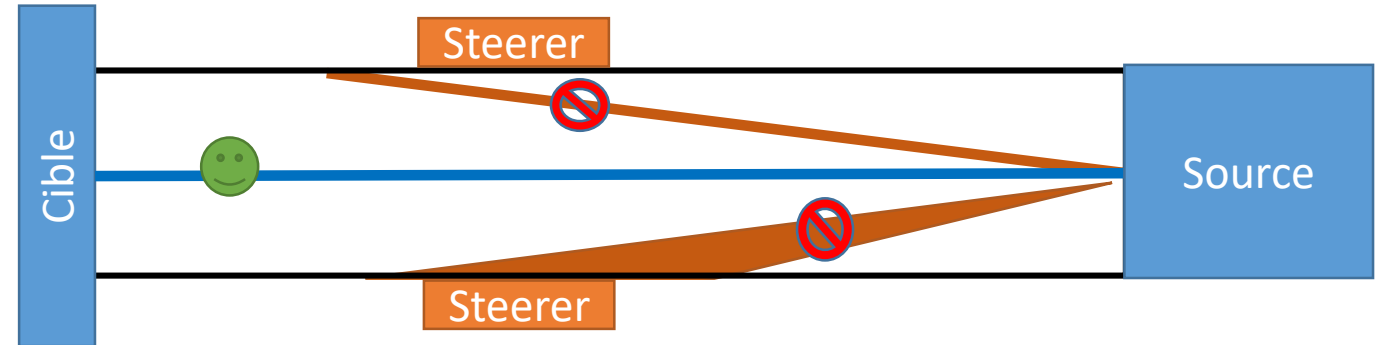
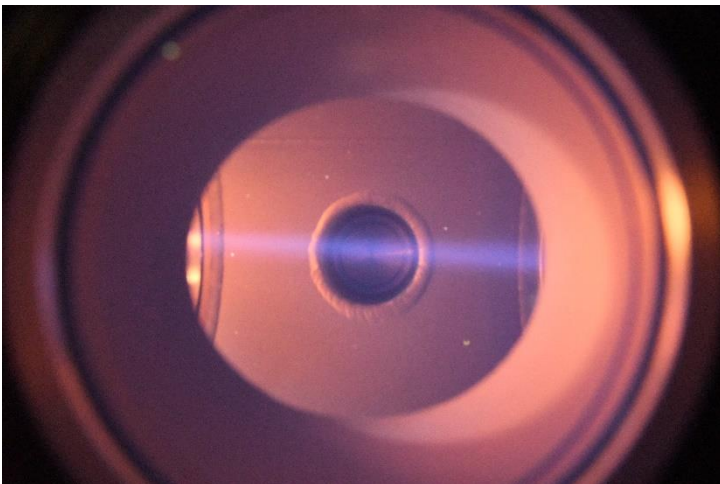
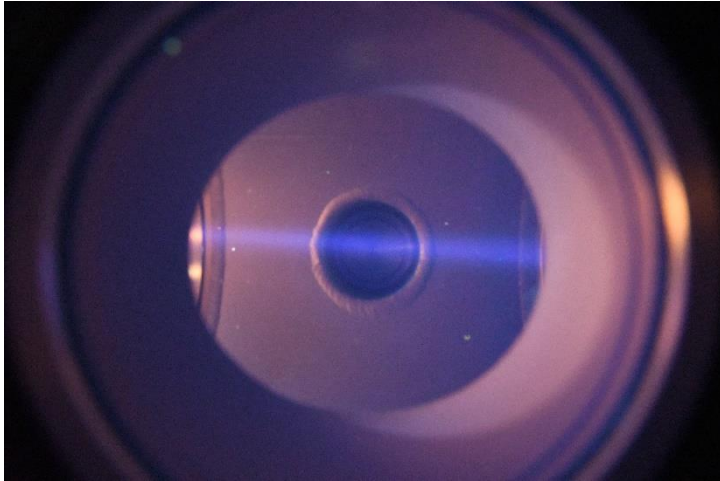


# Contexte AB-nCT (3/3)





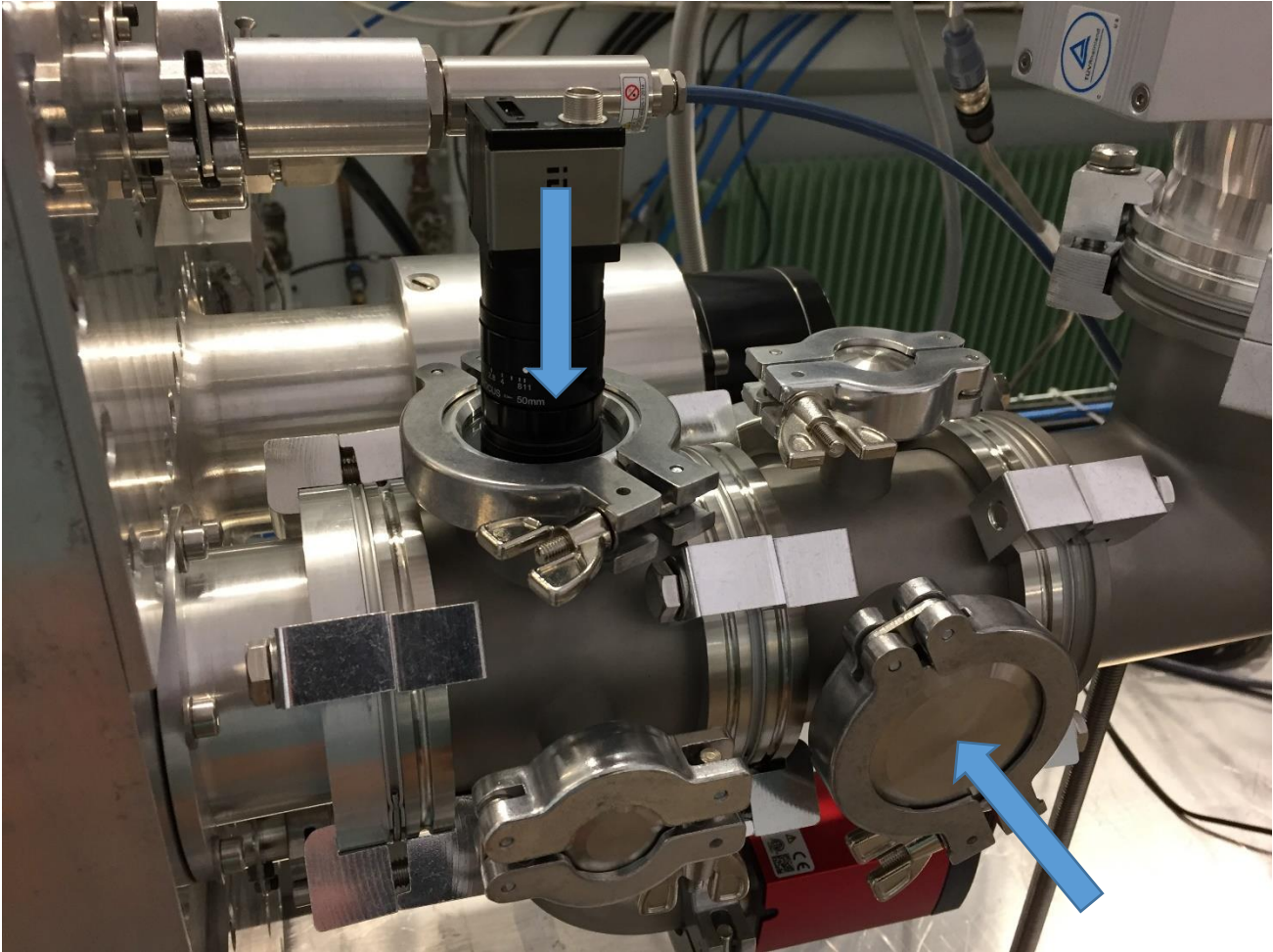
# Problème d'alignement



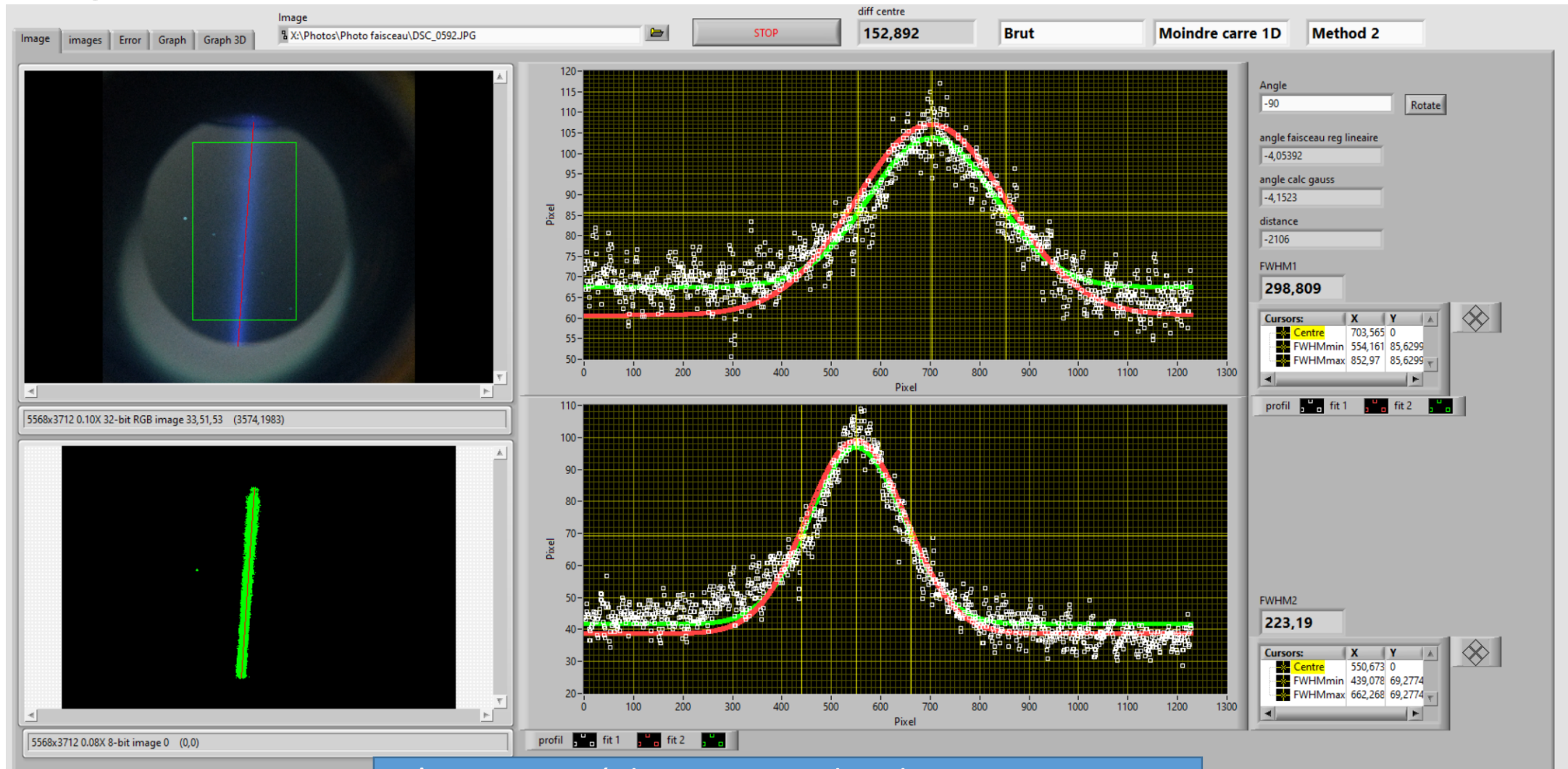
+ regarder dans les hublots pour savoir où est le faisceau

On doit jouer sur les steerers pour bouger/centrer/concentrer le faisceau sur la cible

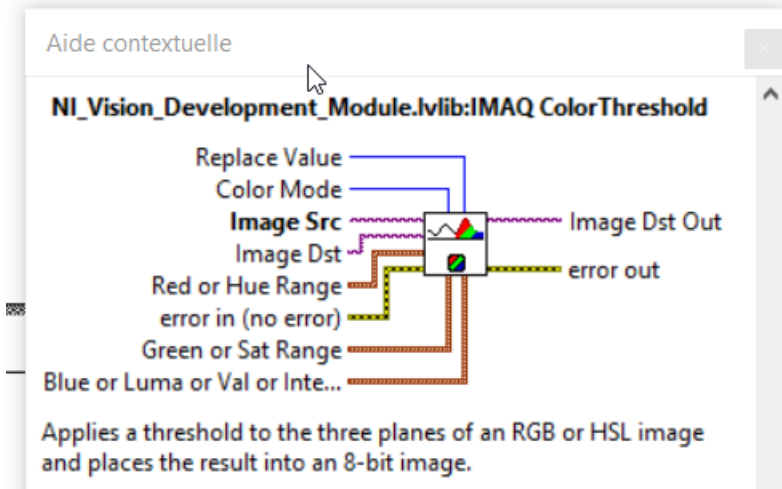
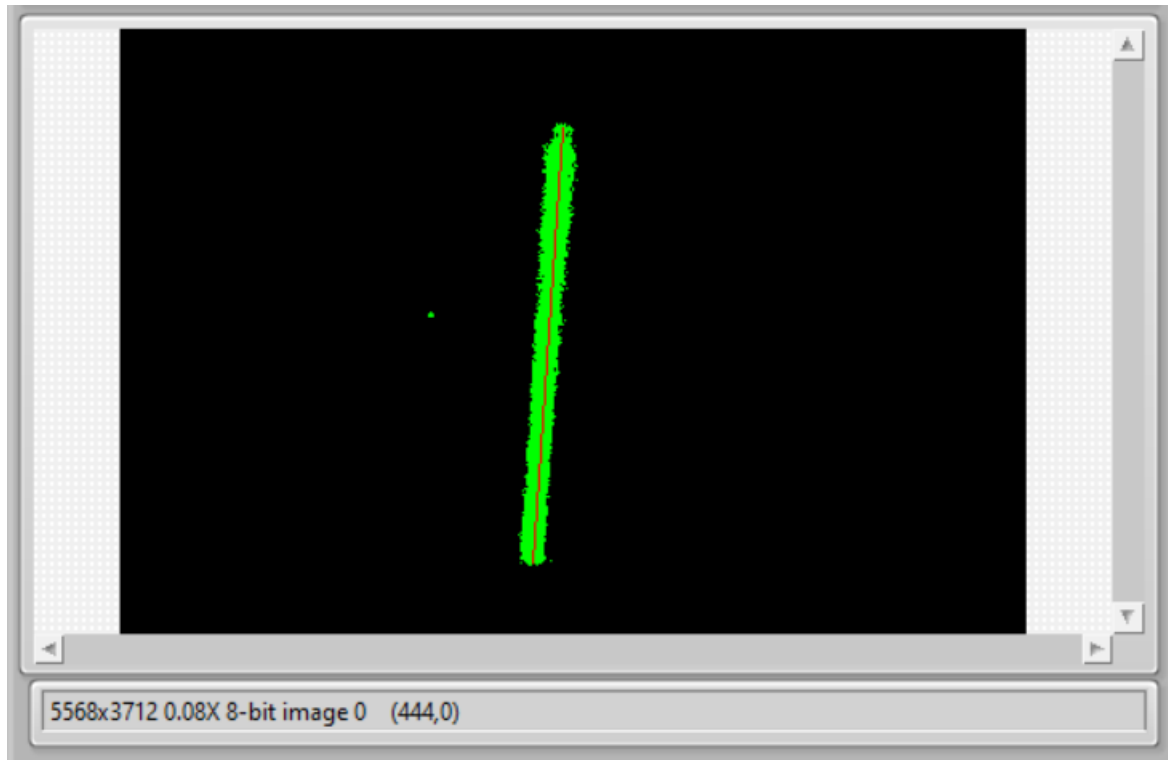
# Systeme de camera



# Programme de test



# 1<sup>er</sup> Approche

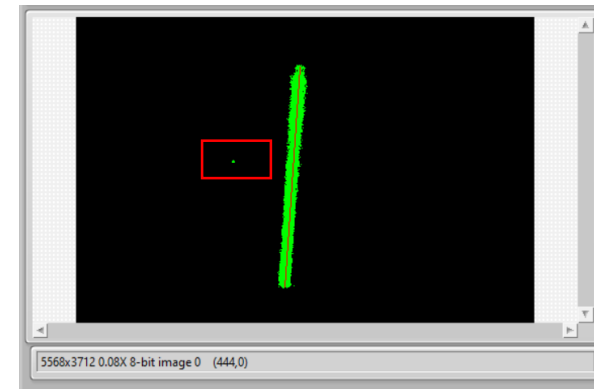


Récupérer les pixels les plus « bleues »

- Dans le RGB, les valeurs vont de 0 – 255
- Quel seuil définir pour prendre les pixels les plus bleues (120 -> 255, 150->255,...)

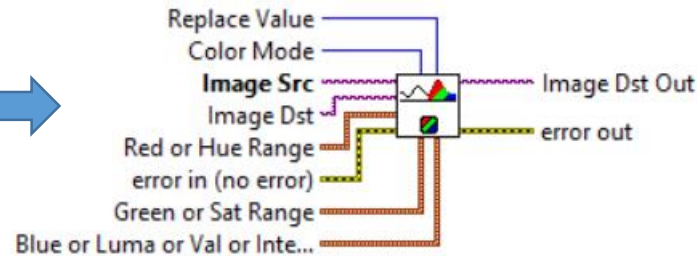


# Parcourir différent seuil

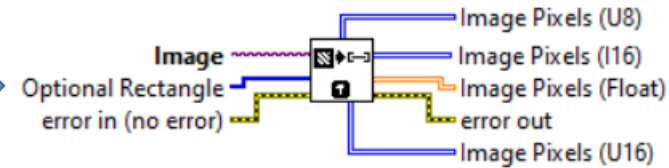


Seuil de 120 à 250  
pas de 10

NI\_Vision\_Development\_Module.lvlib:IMAQ ColorThreshold



IMAQ ImageToArray



J'analyse chaque ligne, soit le pixel est en  
dessous du seuil, soit au dessus

Je construis des tableaux qui indiquent les  
colonnes qui ont un pixel bleue (1 tableau/ligne)

2481	2484	2487	2488	...
------	------	------	------	-----

Moyenne du tableau sans les  
valeurs aberrantes  
(donc 1 valeur/ligne)

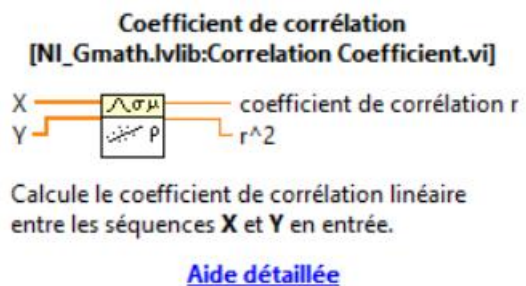
Essayer d'enlever les valeurs aberrantes et génération d'un  
nouveau tableau sans les valeurs aberrantes:

Taille tableau > 100  
 $\bar{x} - 1,96\sigma < \text{val} < \bar{x} + 1,96\sigma$

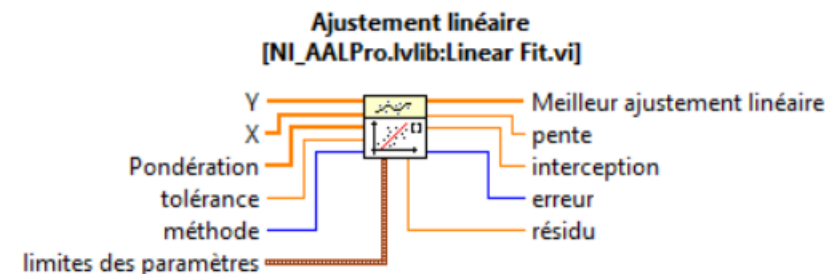
# Tableau final (1<sup>er</sup> approche)

Pour 1 seuil de bleue

Ligne	Moyenne (normalement milieu du faisceau)
1	...
2	...
3	...
4	...
5	...

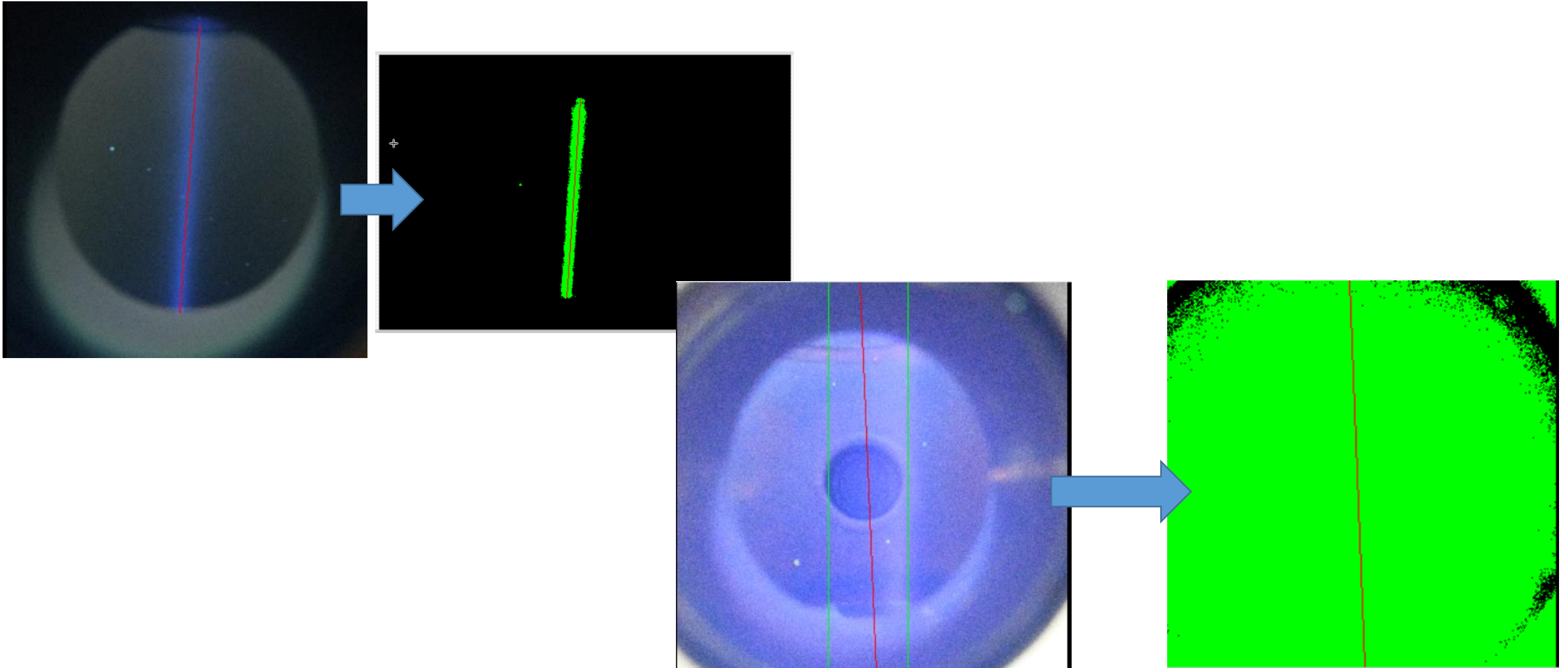


On choisi le seuil avec celui qui a le meilleur coefficient de corrélation

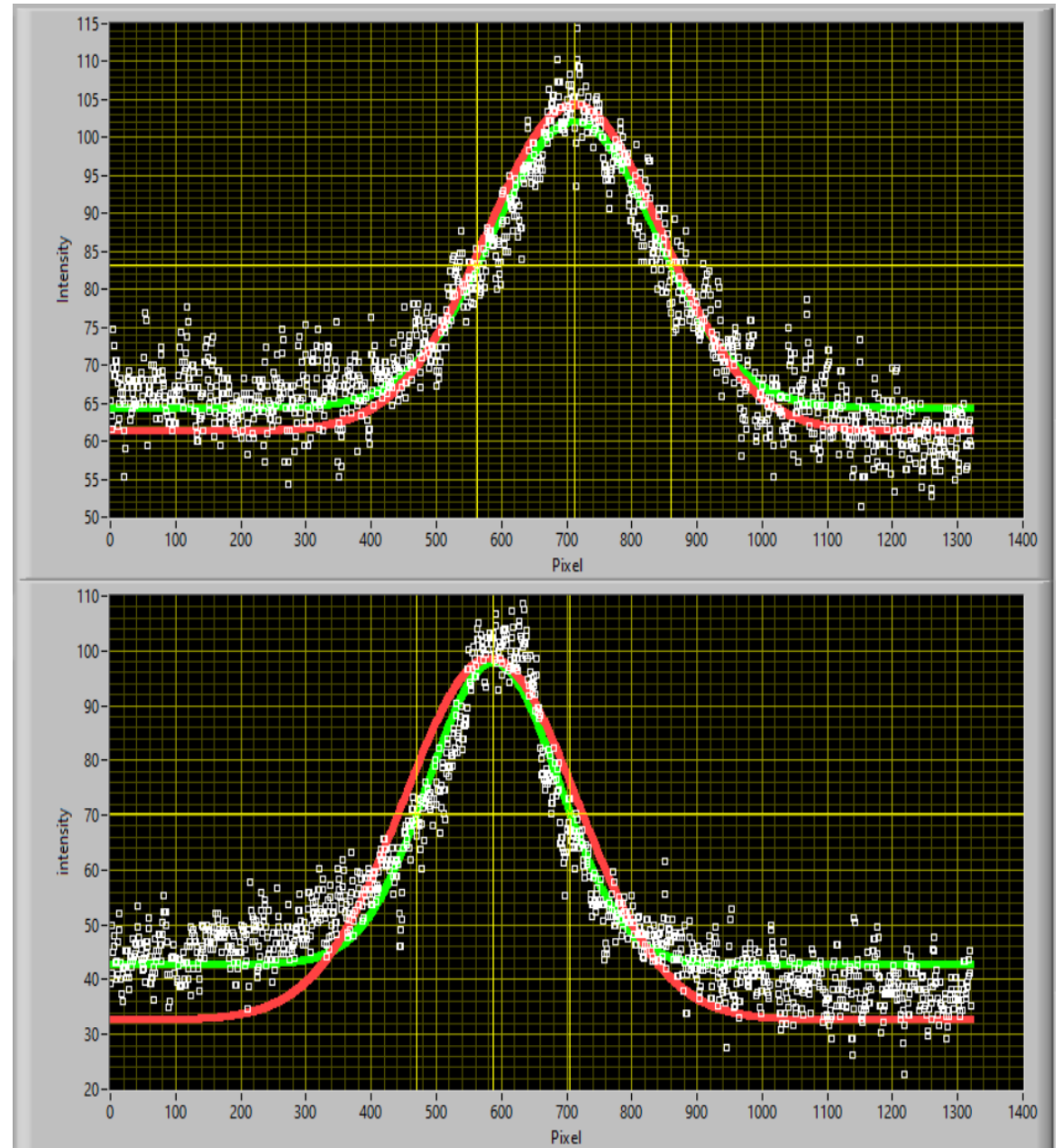
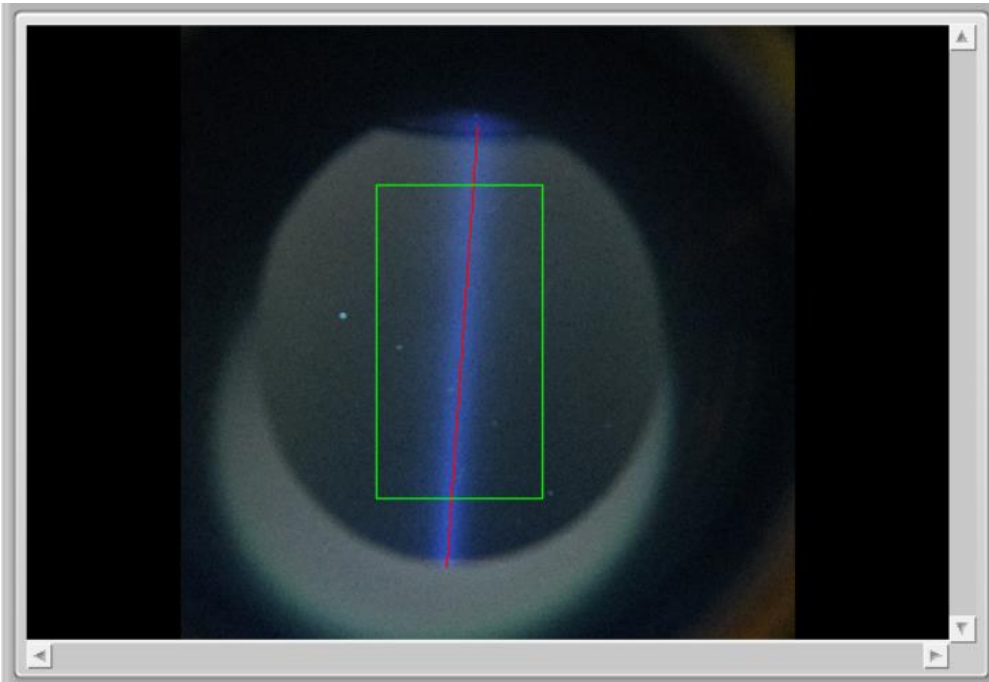


Renvoie l'ajustement linéaire d'un ensemble de données (**X**, **Y**) en utilisant la méthode des moindres carrés, des moindres résidus absolus, ou la méthode bicarrée.

# Résultat 1<sup>er</sup> approche



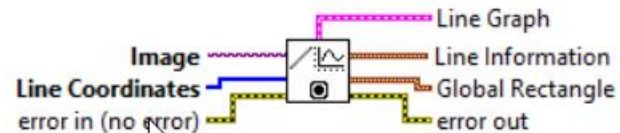
# 2<sup>ème</sup> Approche





# Pixel RGB en Intensité

NI\_Vision\_Development\_Module.lvlib:IMAQ LineProfile



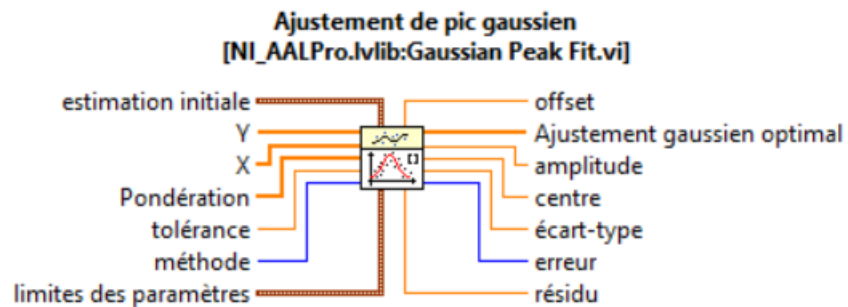
Calculates the profile of a line of pixels. This VI returns a data type (cluster) compatible with a LabVIEW graph. The relevant pixel information is taken from the specified vector (line).

[Aide détaillée](#)

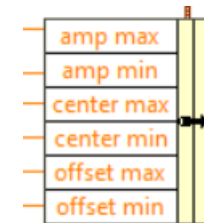
Pour chaque pixels, LabVIEW utilise une formule de ce type:

$$0.2126 * R + 0.7152 * G + 0.0722 * B$$

# Interpolation de la gaussienne



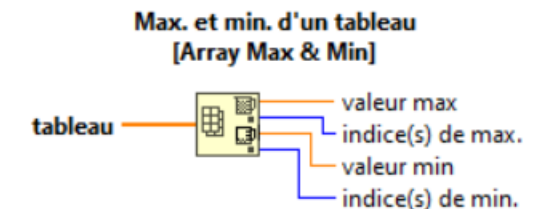
Renvoie l'ajustement gaussien d'un ensemble de données (**X**, **Y**) en utilisant la méthode des moindres carrés, des moindres résidus absolus, ou la méthode bicarrée.



Limites des paramètres

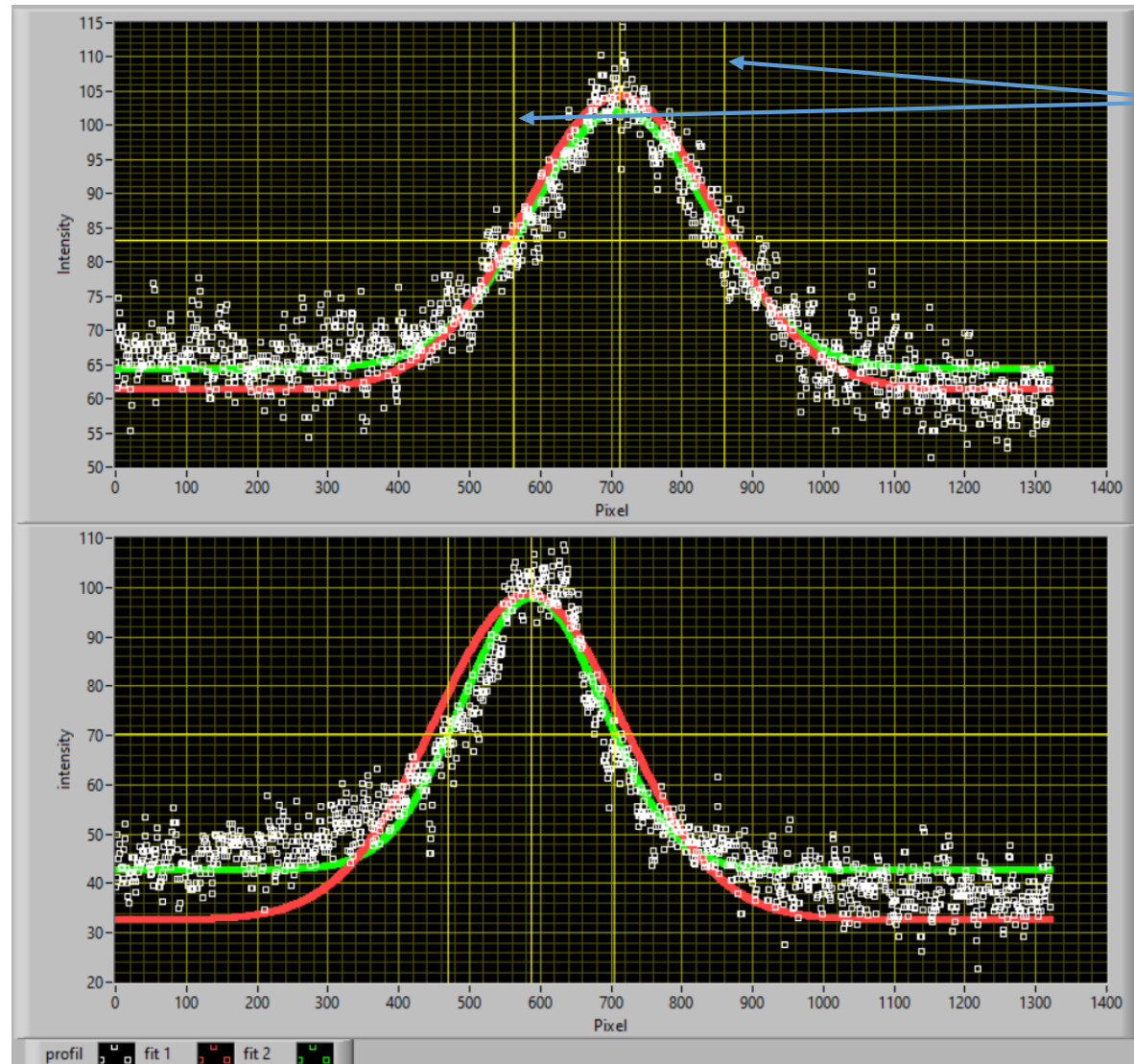
Méthode 1: Pour aider la gaussienne, je lui donne une approximation de l'endroit où se trouve le centre (indice au maximum de la courbe), l'offset le minimum (+ ou - 10) et pour l'amplitude max, le maximum (+ ou - 20)

Méthode 2: Pour aider la gaussienne, je lui donne une approximation de l'endroit où se trouve le centre (indice au maximum de la courbe), peu de limite sur l'offset et l'amplitude



Renvoie les valeurs maximum et minimum trouvées dans le **tableau**, ainsi que l'indice de ces valeurs.

# Résultat 2<sup>ème</sup> Approche



Largeur à mi hauteur  
(FWHM)

# Au final

Au final, on a utilisé la 2<sup>ème</sup> approche qui fonctionne mieux.

La 1<sup>ère</sup> approche fonctionnait bien avec les images tests mais une fois en condition réel avec la caméra, j'avais des problèmes pour trouver le centre du faisceau...