

Réalisation d'un analyseur de faisceaux gaussiens

Quentin Cocheril & Marin Cor
M2 photonique

Projet encadré par Julien Fade et Goulc'hen Loas, équipe DOP, institut Foton

Introduction

Objectif du projet : Réaliser un programme Python pour remplacer un programme existant en langage Matlab, en intégrant de nouvelles fonctionnalités et permettant l'utilisation de différentes caméras. Effectuer les tests en conditions réelles pour validation.

Fonctionnalités du programme initial :

- Affichage en temps réel
- Détection du faisceau, de ses paramètres et fit elliptique gaussien
- Fonction d'alignement de faisceaux

Limites du programme initial :

- Compatibilité avec les caméras
- Temps de traitement, instabilité du programme, bugs récurrents
- Licence Matlab payante obligatoire sur l'ordinateur

Pourquoi Python ?

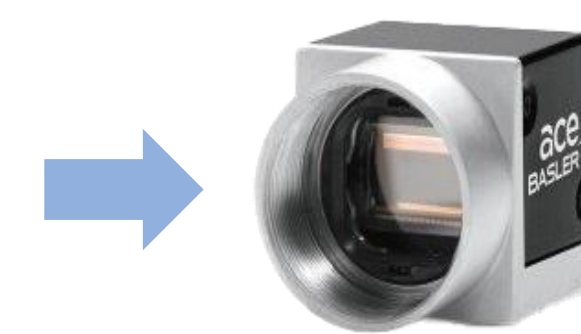
- Open Source, libre → gratuité, multiplateforme
- Grande communauté, forums d'entraide
- Solutions pour presque tous les problèmes que nous pouvions rencontrer
- Bibliothèques complètes, fiables utilisées dans la recherche et l'industrie



Evolutions matérielles

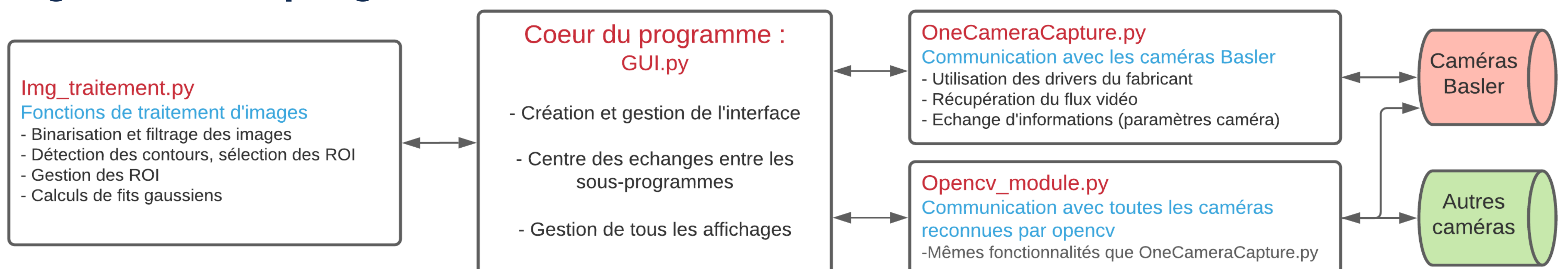
Caméras avec une meilleure dynamique (12 bits)
Pixels plus petits, plus nombreux → meilleure résolution

Exemple
modèle Basler



acA 5472-17um :
• 5472*3648 20MP
• 2,4*2,4 µm pixels
• 12 bits, USB

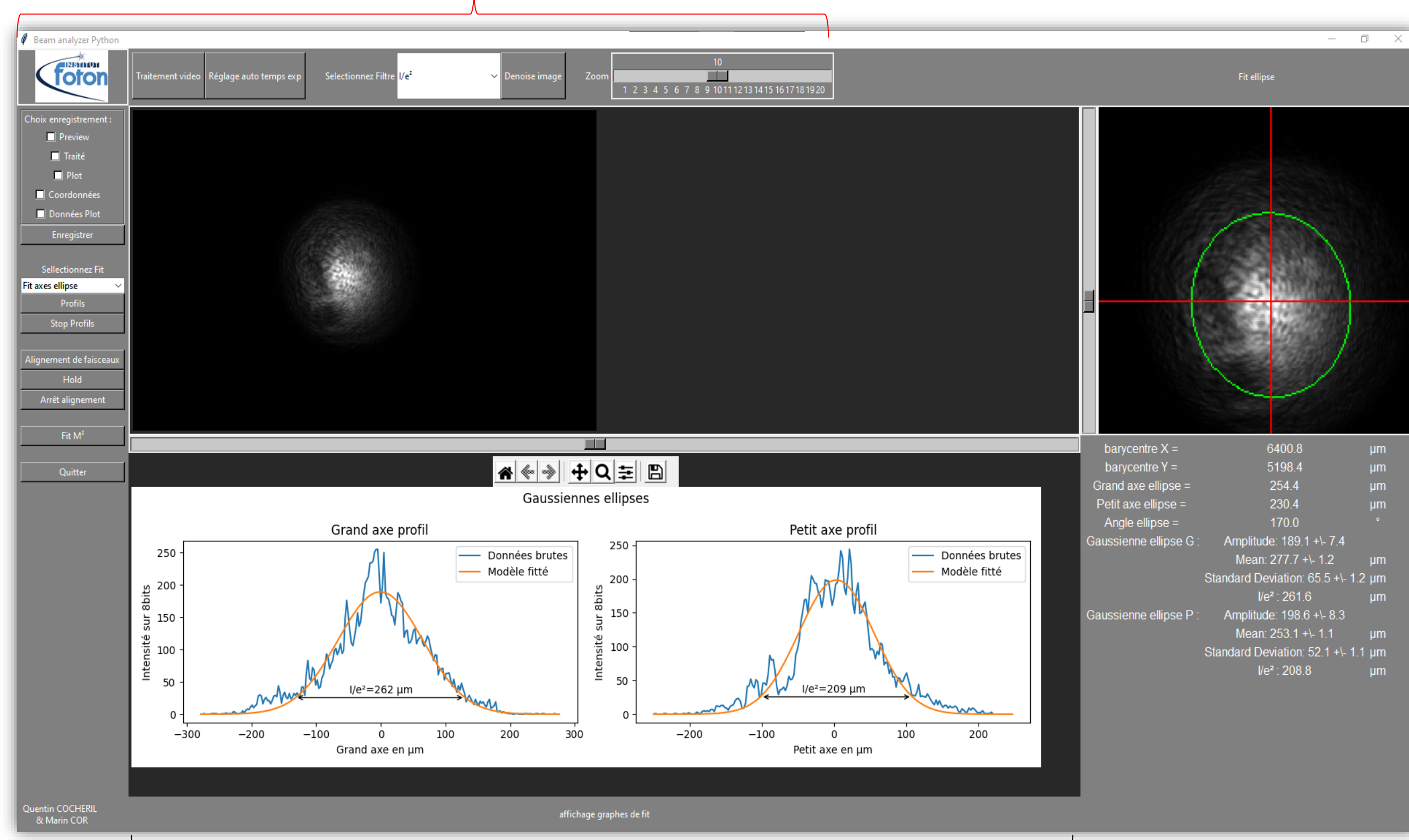
Organigramme du programme



Interface du programme

Commandes de l'interface

(Python 3.8.6 ; Interface : Tkinter)



Preview du traitement d'image

Résultats : paramètres des fit

Affichage des graphes d'intensités spatiales selon différents axes, superposition des fits correspondants

Binarisation et filtrage

3 méthodes de seuillage :

- Otsu** → 2 classes de pixels. Calcul du seuil séparant ces classes, minimisant les variances intra-classes.
- Adaptatif** → Seuil défini en fonction des gradients d'intensité inter-pixels.
- Imax/e²** → Pixels dont l'intensité est inférieure à cette valeur mis à zéro.

Après binarisation

Pour améliorer la définition des contours, on réalise

une **fermeture** $A \bullet B = (A \oplus B) \ominus B$

et une **ouverture** $A \circ B = (A \ominus B) \oplus B$

\oplus : une dilatation

\ominus : une érosion

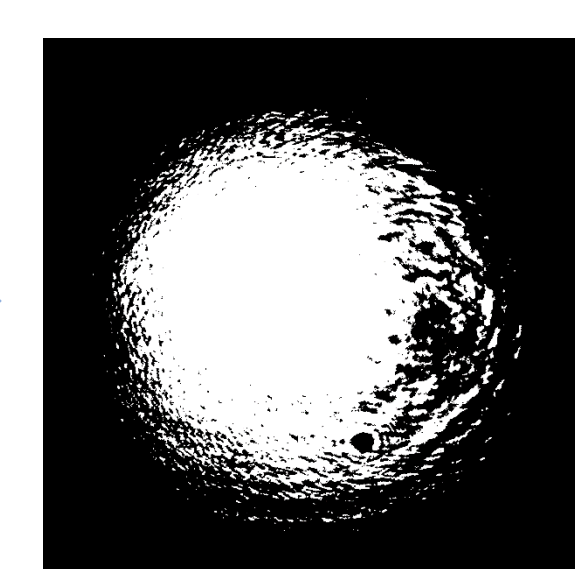
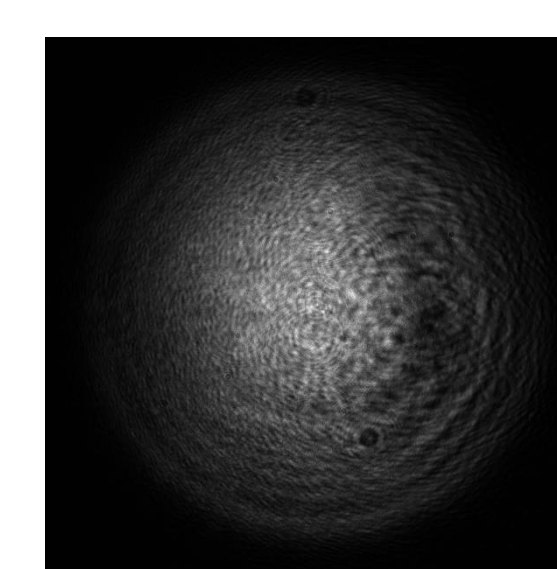
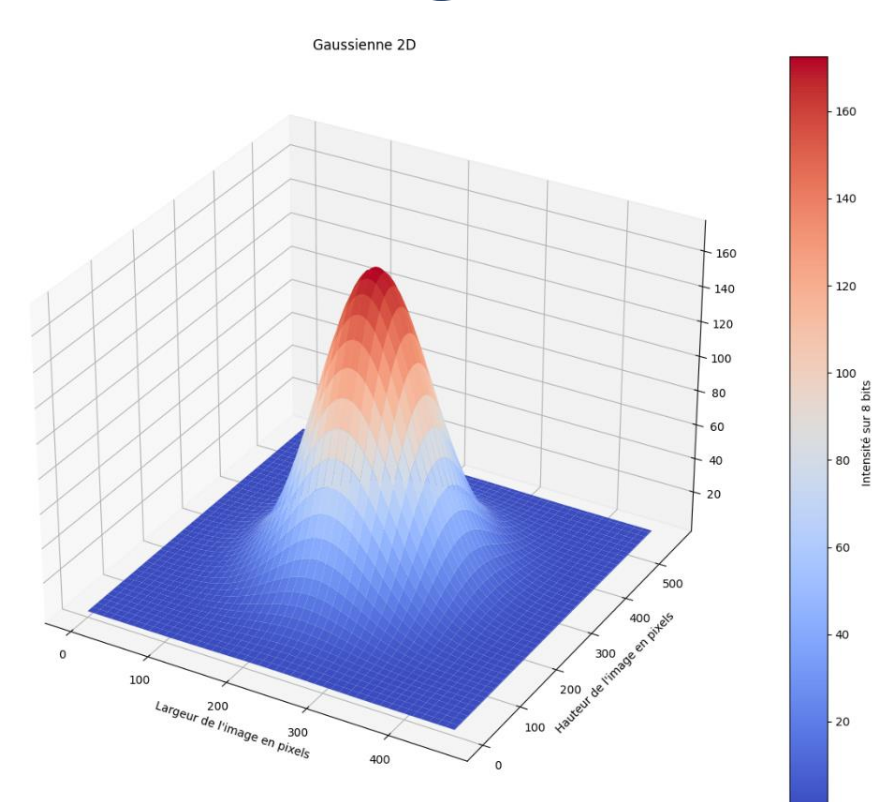


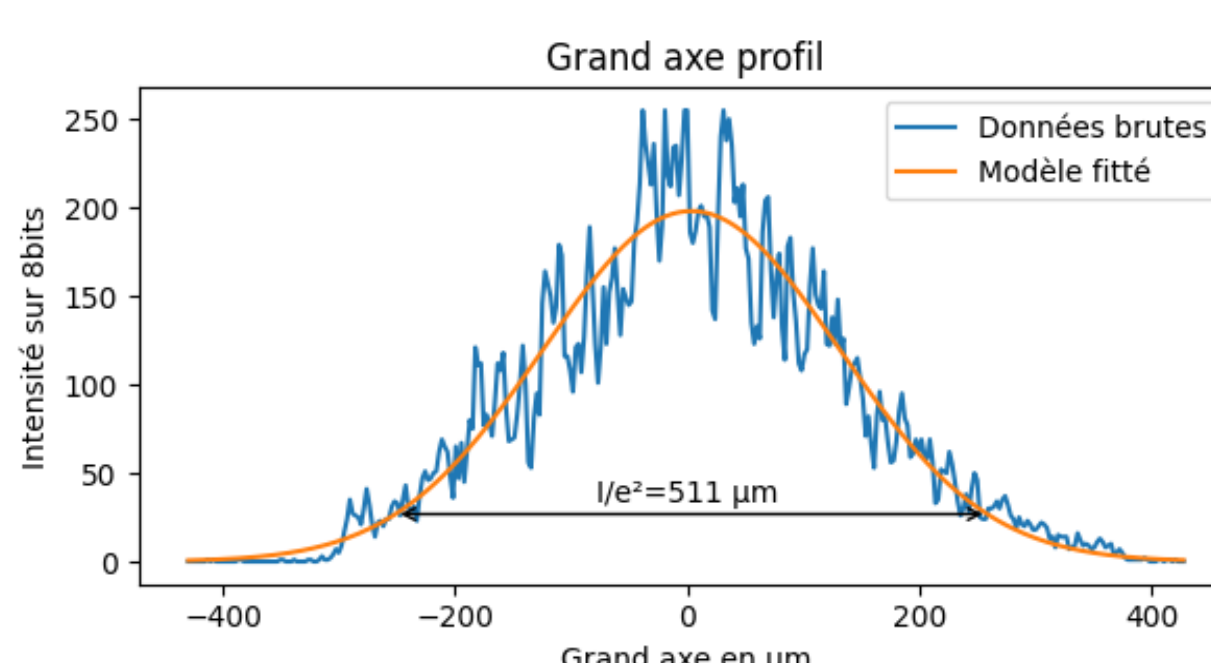
Image après binarisation puis fermeture et ouverture

Faisceaux gaussiens



Fit gaussien de l'intensité selon X puis Y

(Fit possible selon le petit et grand axe si faisceau elliptique)



$$E(r, z) = E_0 \frac{w_0}{w(z)} \exp\left(-\frac{r^2}{w^2(z)}\right) \exp\left(-ikz - ik\frac{r^2}{2R(z)} + i\zeta(z)\right)$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Conclusion

Objectifs initiaux remplis.

Fonctionnalités du programme dans l'état actuel :

- Preview flux vidéo
- Fits gaussiens (axes XY, petit et grand axes ellipse, 2D)
- Alignement de faisceaux
- Enregistrement des résultats
- Temps d'exposition automatique
- Tests opérationnels concluants.

Principe mesure M² :

Plusieurs mesures de w le long de l'axe z

Fit non linéaire : obtention de l'équation $w(z) = Az^2 + Bz + C$

