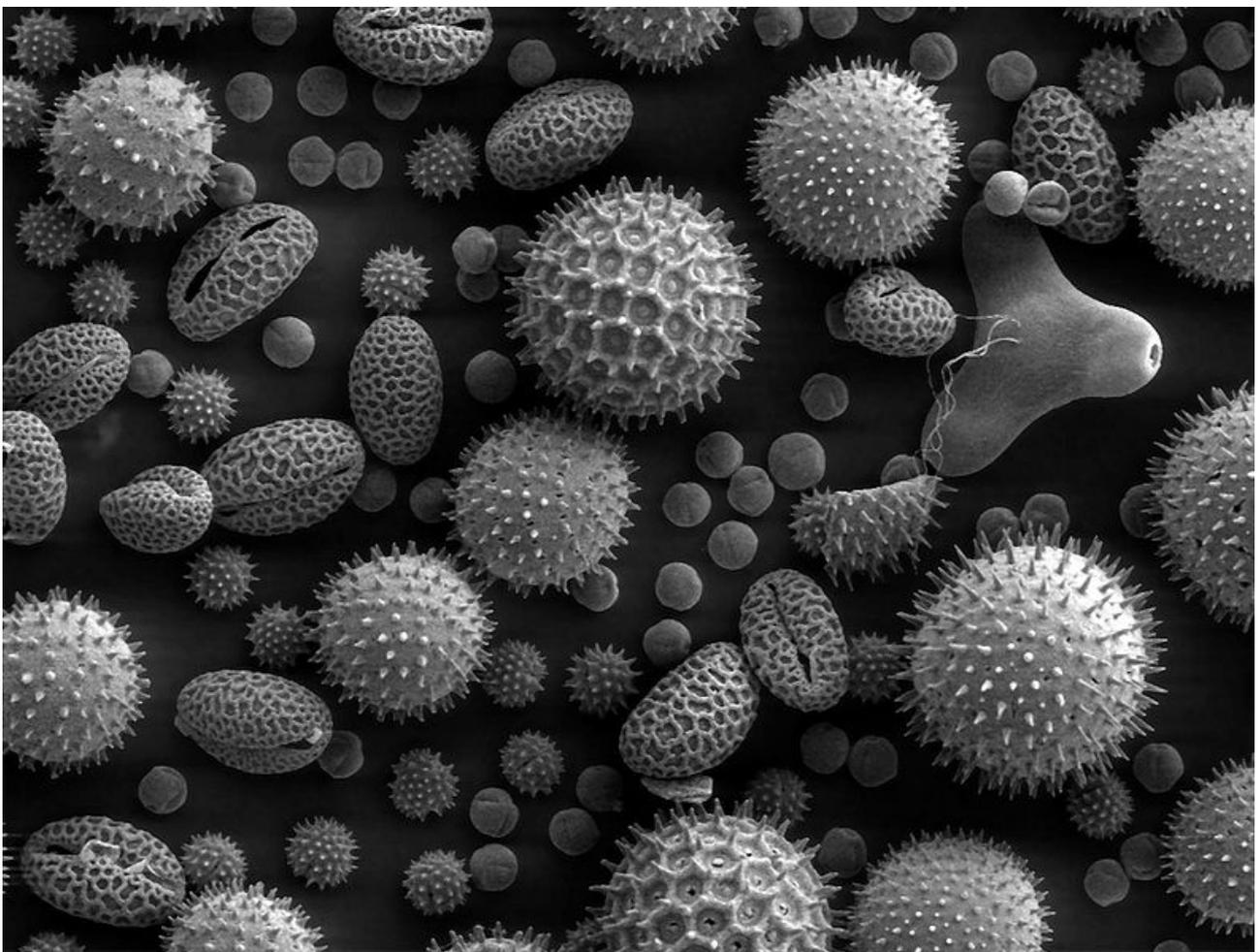


IETI
Microscopie par illumination structurée
Rapport Technique
Institut d'Optique Graduate School
1A 2023



Cohlet Adh mar
Engler Hannah
Pernot Valentine
Pichon--de Lagarde Camille

Table des matières

Introduction

Objectifs

Schéma de principe

Cahier des charges

Notice d'utilisation

Schéma fonctionnel

Description

Fonction main

Librairie caméra

Intégration

Conclusion

Bilan Technique

Bilan Équipe

Introduction

Objectifs

L'objectif du projet est l'automatisation d'un TP de biophotonique réalisé en 3ème année par les élèves de l'IOGS à Palaiseau. Ce TP a pour objectif d'acquérir en 3D un échantillon biologique. en superposant plusieurs plans afin d'obtenir une représentation en 3D. L'image de chaque plan est obtenue par le traitement d'une séquence d'images du même plan illuminée par des mires UV différentes. L'automatisation permet d'éviter la difficulté technique du photoblanchiment, qui intervient lorsque le temps d'exposition est trop long.

Le schéma du montage du TP est donné ci-dessous:

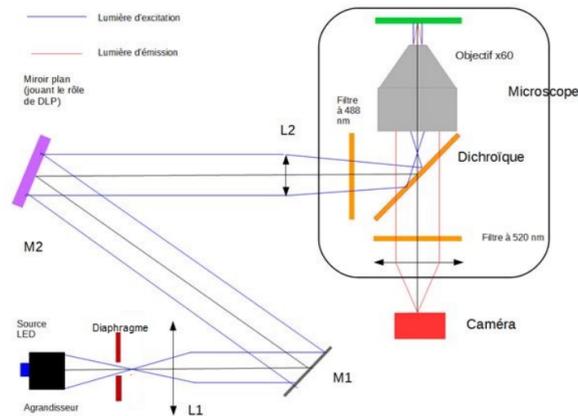
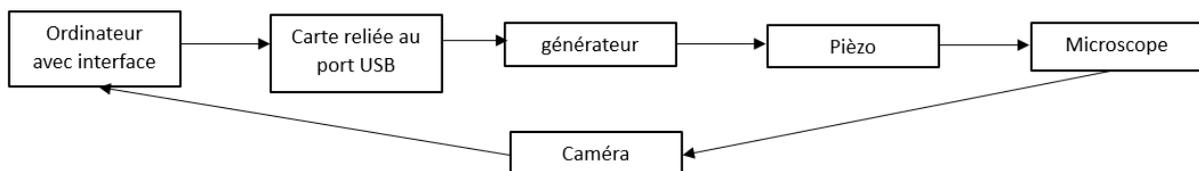


Schéma du montage du TP

Schéma de principe



Cahier des charges

Processus automatisé :

- Déplacement du Piézo automatisé
- Récupération et traitement des images

Interface utilisateur :

Le projet doit disposer d'une interface utilisateur conviviale et facile à utiliser pour les utilisateurs finaux. L'interface doit permettre à l'utilisateur de contrôler le déplacement du Piézo à l'aide d'un

bouton ou d'un autre élément interactif. L'interface doit être intuitive et facile à comprendre, même pour les utilisateurs novices.

Fonctionnalités :

- Contrôle automatique du déplacement du Piézo
- Interface utilisateur intuitive et facile à utiliser
- Communication avec le port USB pour le contrôle du Piézo
- Capacité à enregistrer et à exporter les données de déplacement du Piézo
- Traiter des images

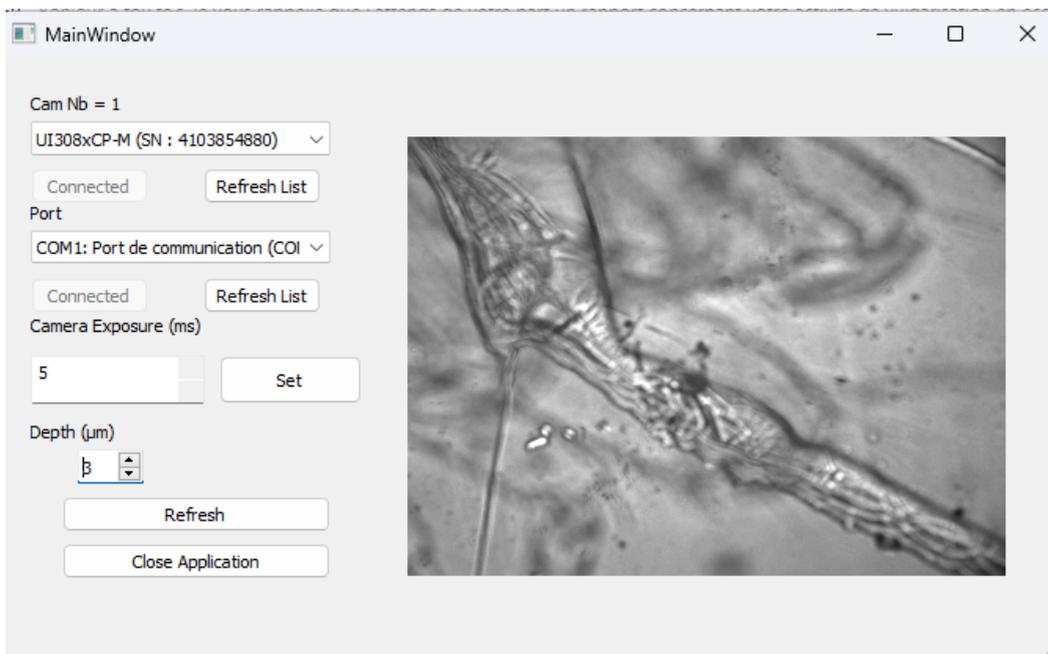
Exigences techniques :

- Utilisation d'un port USB pour communiquer avec le Piézo
- Utilisation d'un langage de programmation approprié pour contrôler le Piézo
- Utilisation d'une caméra pour récupérer des images
- Conception d'une interface utilisateur conviviale et facile à utiliser

Contraintes de temps :

- Le projet doit être achevé dans les délais impartis (pour le 16 mai)

Notice d'utilisation



Une fois l'interface affichée (capture d'écran ci-dessus), connectez la caméra en la sélectionnant dans le menu déroulant "connect camera" puis en cliquant sur "connect".

Ensuite, connectez le piezo en sélectionnant le port USB sur lequel le piezo est connecté puis cliquez sur "connect".

Entrer ensuite la durée d'exposition souhaitée dans le menu "exposure", puis cliquer sur "Set".

L'exposition idéale dépend aussi de l'éclairage de l'échantillon par le microscope, il faut donc y faire attention.

Finalement, dans le menu "Depth (µm)" sélectionner le nombre d'images souhaitées (entre 1 et 11).

Ensuite cliquer sur "Refresh".

Dans le dossier sur lequel se trouve le code Python vous trouverez vos photos enregistrées.

Pour quitter l'interface, appuyez sur "Close Application".

Exemple d'une image enregistrée par ce processus :

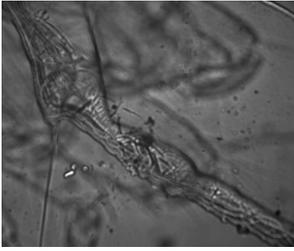
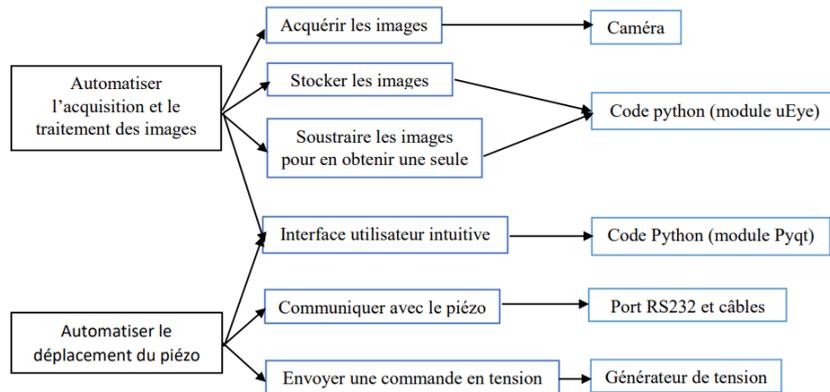


Schéma fonctionnel



Description

Cette partie a comme objectif de décrire les différentes fonctions de notre code informatique.

Librairie main

init(self)

refreshGraph(self)

refreshExposure(self)

initListCamera(self)

initListPort(self)

connectPort(self)

connectCamera(self)

closeApp(self)

closeEvent(self)

value_changed(self)

Cette fonction calcule les différentes positions prises par le piézo, déplace le piézo et capture les images et les enregistre.

On note N le nombre de valeurs prises par le piézo, puis on calcule l'intervalle entre chaque valeur, en sachant qu'on peut faire varier le piézo de 0 à 5 Volts. On calcule toutes les valeurs en volts prises par le piézo $valeur(k) = valeur(k - 1) + intervalle$ avec $valeur(0) = 0$ à l'aide d'une boucle et on met toutes les valeurs dans la liste valeurs.

On utilise ensuite le module nidaqmx qui permet de communiquer avec un port USB. On connecte ce port USB à un driver. Le piézo est connecté à la borne ao0 du driver. Pour chaque valeur de la liste on déplace le piézo puis on capture l'image et on l'enregistre grâce aux fonctions du module caméra.

A la fin de la boucle on obtient les N images des N plans de la cellule.

Librairie caméra

```
get_nb_of_cam()
get_cam_list()
_init_(self,cam_id,0)
init(self)
get_cam_info(self)
get_sensor_info(self)
get_sensor_max_width(self)
get_sensor_max_height(self)
set_display_mode(self,mode)
capture_video(self)
stop_video(self)
alloc(self)
un_alloc(self)
stop_camero(self)
get_mem_info(self)
get_image(self)
get_aoi(self)
set_aoi(self,x,y,w,h)
get_colormode(self)
get_exposure(self)
get_exposure_range(self)
set_exposure(self,exposure)
get_frame_rate(self)
get_frame_time_range(self)
set_framerate(self,fps)
get_pixel_clock(self)
get_bits_per_pixel(color_mode)
```

Ces lignes nous permettent de récupérer la liste des caméras disponibles et ensuite de choisir celle voulue via l'interface. Cela permet également de modifier l'exposition après avoir récupéré les images de la caméra. On récupère une image actualisée dès que l'on appuie sur "refresh" sur l'interface.

Intégration

On appelle la classe main et on l'exécute grâce à la commande main.show()

Conclusion

Bilan Technique

Le programme fonctionne correctement, mais nous pouvons imaginer plusieurs axes d'améliorations (pour une future équipe ?) :

- Remplacer "depth" par "nombre d'images souhaitées"
- Continuer d'automatiser le TP en automatisant le DMD
- Après avoir automatisé le DMD, faire la soustraction des images obtenues par l'ordinateur afin d'automatiser complètement le TP.

Bilan Équipe

Bon travail en équipe. Lorsque l'équipe était au complet, le projet avançait à une vitesse satisfaisante. Néanmoins, la difficulté du projet et le fait que seulement un encadrant savait faire ce projet ont limité considérablement l'équipe.

