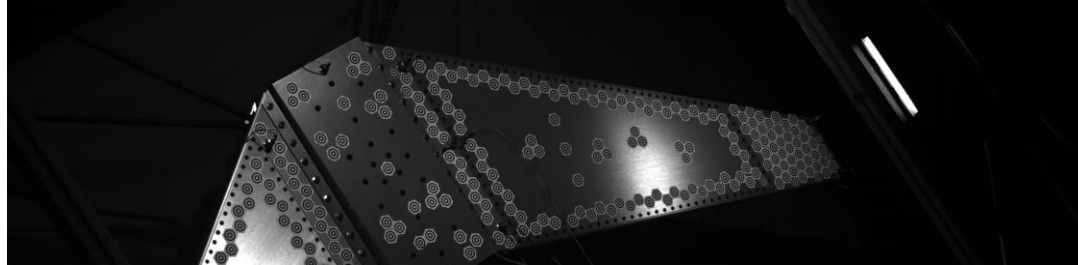
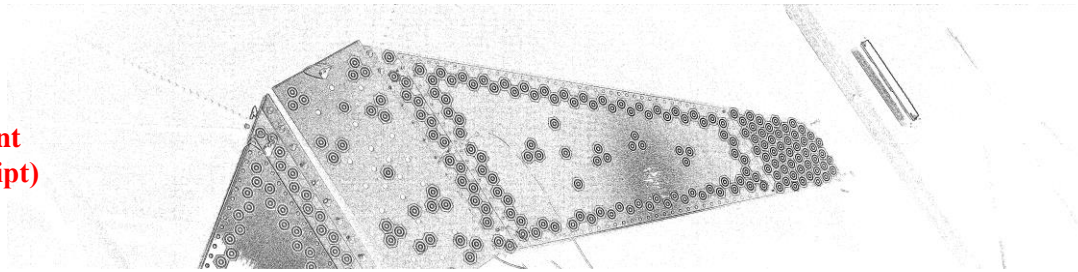


# Présentation: Exercice openCV

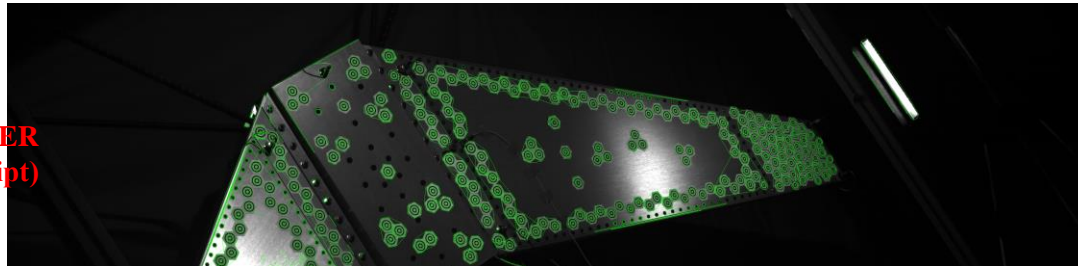
Présentée par: Hamza Ghanmi  
Ingénieur en mécanique et en thermique-énergétique



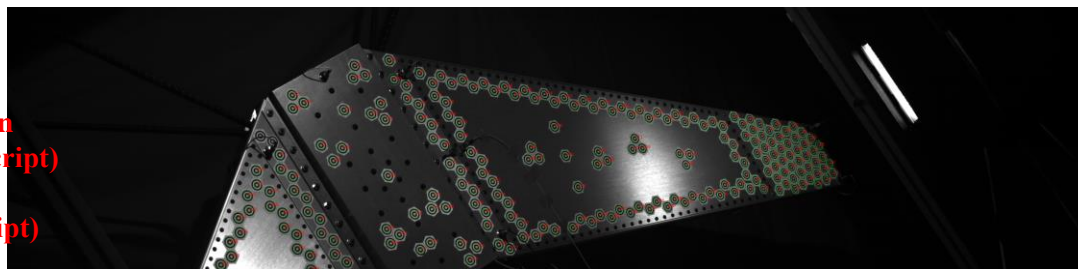
**Partie 1:**  
**Post traitement**  
**L4 → L6 (script)**



**Partie 2:**  
**Détection MSER**  
**L7 → L9 (script)**



**Partie 3:**  
**Filtrage, Dessin**  
**L14 → L33 (script)**  
**Affichage**  
**L35 → 41 (script)**



- Le problème consiste à développer un script Python en utilisant la bibliothèque openCV qui nous permet de détecter et de numéroté des mires collées sur une structure
- L'algorithme se décompose en 3 parties: partie post traitement d'image, partie détection des régions par la méthode MSER, partie filtrage affichage et numérotation des contours superposés aux mires

- Afin de préparer notre image à la phase de détection, un post traitement est nécessaire pour obtenir des meilleurs résultats
- Le post traitement que j'ai prévu est un traitement de binarisation (Noir, Blanc)
- En analysant l'image source, j'ai remarqué une variation de luminosité, ce qui m'a amené d'utiliser une méthode adaptative gaussienne, puis j'ai varié les différents paramètres pour des bons résultats

- La détection des régions est réalisée grâce à la méthode MSER
- On remarque la détection de plusieurs régions non souhaités d'où la nécessité d'un travail de filtrage

- Le filtrage est réalisé selon le nombre des points existant dans les contours, dans le but de garder seulement les formes circulaires (qui sont caractérisées par un grand nombre de points)
- Un second filtrage sur les surfaces pour éliminer les très grandes surfaces et les très faibles surfaces
- Un troisième filtrage sur les formes circulaires qui ont des centres très proches (concentriques), dans le but de garder un seul représentant du mire détecté
- En fin, le dessin et la numérotation des contours sur l'image source