

1 Séance 2

1.1 Sujet A : Filtrage par TF

Temps conseillé : 4h

A partir du fichier `image_filter_gui.py`, on souhaite pouvoir :

- ouvrir une image
- calculer la TF de cette image (FFT2D)
- créer un masque sur la TF (circulaire par exemple ou rectangulaire)
- générer l'image résultante (par TF inverse) et l'afficher

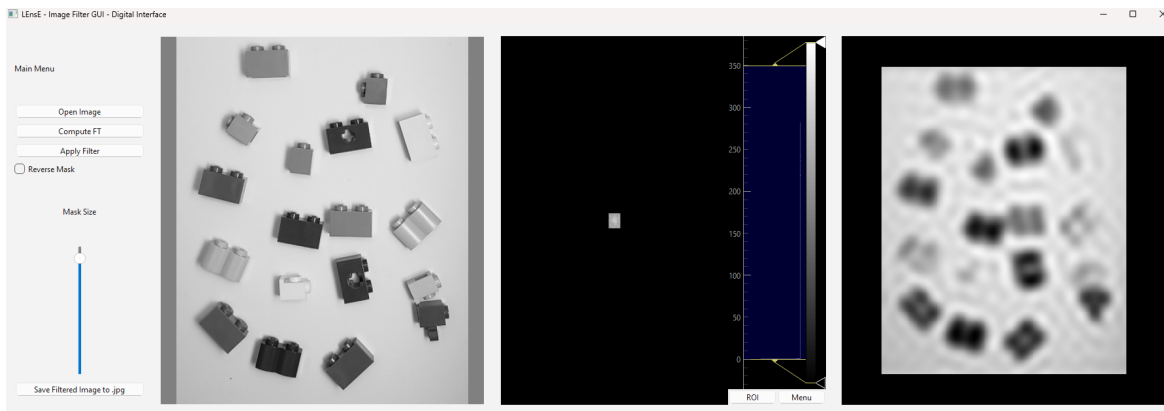


Figure 1: Exemple d'application

Quelques indications :

- Lancer l'application et visualiser le résultat.
- Ajoutez un bouton qui permettra de lancer l'affichage de la TF. Ce bouton doit être actif seulement après chargement de l'image. La TF de l'image sera affichée dans le widget `tf_section`. L'affichage sera réalisé grâce à la méthode `pyqtgraph.image()`. Définissez une méthode de la classe `MainWindow` permettant de calculer la TF de l'image. On utilisera **OpenCV**.
- Définissez un masque qui sera appliqué à la TF de l'image. La forme du masque est libre.
- Ajoutez un bouton qui n'est actif qu'après l'affichage de la TF de l'image. Ce bouton va déclencher l'application du masque sur la TF puis le calcul de la TF inverse. Le résultat sera affiché dans le widget `result_section` à l'aide de `pyqtgraph.image()`.
- Un bouton **Save Filtered Image to .jpg** est défini mais inactif. Rendez ce bouton utilisable seulement après l'application du filtre.

Pour aller plus loin :

- On veut que l'utilisateur puisse facilement changer la taille du masque. On pourra par exemple utiliser un widget **QSlider** ainsi que la méthode `sliderMoved.connect()` afin de mettre à jour la taille du masque en temps réel. On obtient la valeur du widget grâce à la méthode `QSlider.value()`.

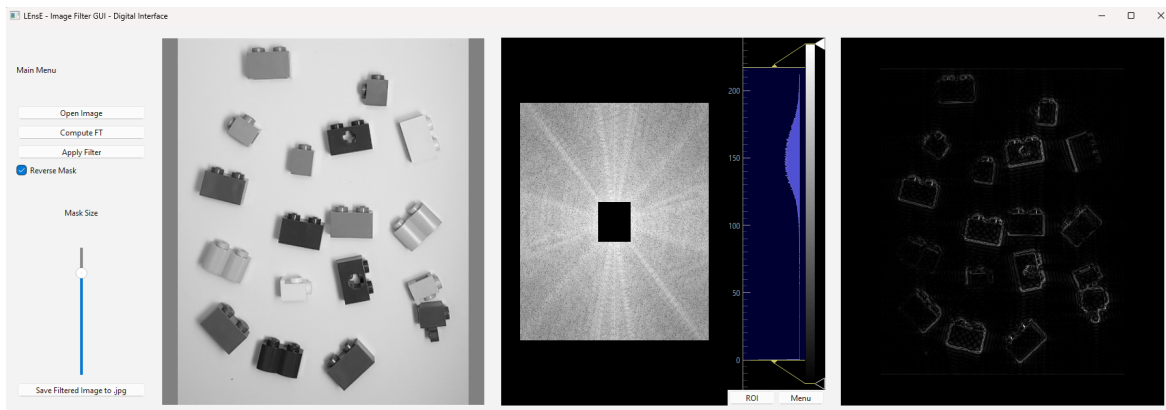


Figure 2: Exemple d'inversion du filtre

- On pourra ajouter un widget du type **QCheckBox** permettant d'inverser d'un clic le sens du filtre (cf. Figure 2)
- Discutez les effets observés des différents filtres