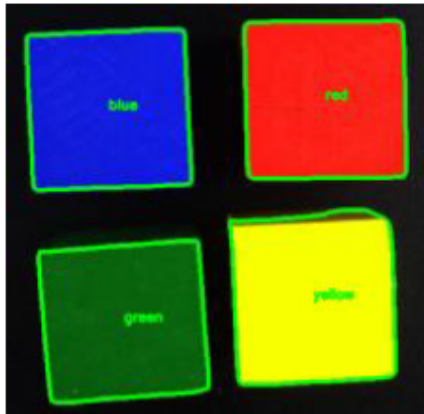


Travaux Pratiques

Semestre 6

Vision Industrielle

TP 1 / Banc de vision industrielle
TP 2 / Bases de traitement d'images sous OpenCV



Ce sujet est disponible au format électronique sur le site du LEnSE - <https://lense.institutoptique.fr/> dans la rubrique Année / Première Année / Opto-Electronique S5 / TP / Bloc Vision Industrielle.



© 2026 by LEnSE-IOGS

Ce bloc de travaux pratiques utilise un **banc de vision industrielle** avec une lampe de type Effi-Ring RGB, une caméra Basler et une interface développée en **Python** (*PyQt6*) et qui utilise des fonctionnalités de la bibliothèque **OpenCV**.

Les documentations de la caméra et de l'éclairage sont disponibles aux adresses suivantes :

- Basler **a2A 1920 - 160ucBAS** : <https://docs.baslerweb.com/a2a1920-160ucbas#specifications>
- **Effi-Ring** : <https://www.ffmpeg.com/fr/produits/annuaire/effi-ring>

INTERFAÇAGE NUMÉRIQUE

Travaux Pratiques

Semestre 6

TP1 - Banc de vision industrielle

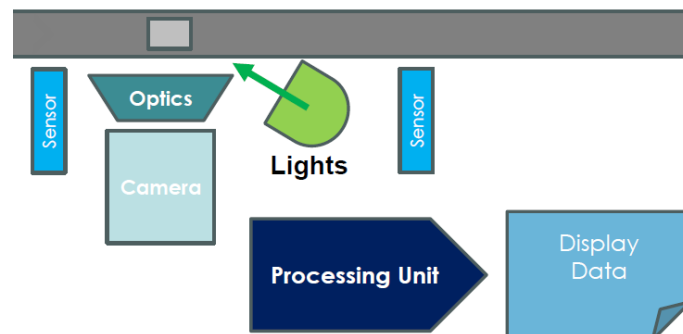
Une feuille de résultats est à remplir et à présenter tout au long de la séance de TP.

Chaîne de vision industrielle

Cette séance se base sur un **banc de vision industrielle** contenant un éclairage annulaire, une caméra et son objectif, des objets à analyser et d'un logiciel de pilotage de la caméra.

Cette séance a pour but :

- d'**analyser l'impact des différents maillons** d'une chaîne d'acquisition sur la qualité de l'image
- de **proposer des méthodes quantitatives de mesure** de la qualité d'image
- de **modéliser cette chaîne** de manière simple



Ce document est complété par un diaporama présentant quelques notions de base de la vision industrielle / Disponible sur le site du LEnsE ou dans le document de ressources.

A - Prise en main de l'interface [30 min]

→ M Allumer l'éclairage annulaire du banc (trois couleurs). Placer un **fond uniforme** sous l'éclairage (feuille blanche, pailleuse...).

→ Q Quelle couleur d'éclairage obtient-on ?

→ M Lancer l'application depuis `S :/_GUI/START_VI.BAT`.

→ M Ouvrir l'onglet `IMAGE OU CAMÉRA`.

Si la caméra est bien connectée en USB à l'ordinateur, vous devriez voir s'afficher son flux.

→ M Ouvrir l'onglet `ZONE D'INTÉRÊT`.

Cet onglet servira par la suite à définir une zone d'acquisition plus restreinte.

Histogramme

→ M Ouvrir l'onglet `HISTOGRAMME` puis sélectionner le mode `RÉPARTITION SPATIALE`.

Dans cette section, vous allez pouvoir :

- visualiser l'histogramme de l'image acquise par la caméra, le sauvegarder
- sauvegarder l'image acquise
- modifier le temps d'exposition et le black level de la caméra

→ Q Que représente l'histogramme d'une image ? A quoi peut-il servir ?

→ M Placer le *black level* à 0. Modifier le **temps d'intégration** de la caméra.

→ Q Que se passe-t-il sur l'image ? Sur l'histogramme ?

→ Q Quelles sont les valeurs minimale et maximale que peuvent prendre les pixels de la caméra ? Quelle est alors la **profondeur binaire** de la caméra utilisée ?

L'objectif monté sur la caméra possède 2 bagues qui permettent de changer : l'**ouverture numérique de l'objectif** et la **mise au point**.

→ M Imposer un temps d'intégration qui ne sature pas le capteur (pour un éclairage blanc). Modifier l'ouverture numérique d'un cran.

→ Q Que se passe-t-il sur l'image ? Sur l'histogramme ?

Mise au point et zone d'intérêt

→ M Placer la mire graduée (zone avec écriture) dans le champ de la caméra. Ajuster la seconde bague de l'objectif pour faire la mise au point sur l'objet.

→ M Dans l'onglet `ZONE D'INTÉRÊT`, ajuster la zone d'intérêt (ou *Area of Interest* - AOI) pour ne sélectionner qu'une partie de l'image autour de l'objet (environ 500 pixels par 500 pixels).

Pour la suite du TP, on s'assurera de prendre une zone d'intérêt à peu près centrée dans l'image et d'une taille d'environ 500 par 500 pixels.

Profil dans l'image

→ M Placer des cubes de couleur dans le champ de la caméra. Sélectionner une zone d'intérêt d'environ 500 pixels par 500 pixels autour des objets à visualiser. Ajuster le temps d'intégration pour obtenir une image non saturée avec un éclairage blanc.

→ Q Commenter l'image et l'histogramme obtenus. Que se passe-t-il avec un cube d'une autre couleur ?

→ M Ouvrir l'onglet `OUTILS POUR IMAGES` puis sélectionner le mode `PROFIL DANS L'IMAGE`.

→ M Déplacer les positions des profils horizontal et vertical. Observer les profils obtenus pour différentes positions.

→ Q Quel est l'intérêt d'un tel outil ?

Echantillonnage et quantification

Conserver les objets dans le champ de la caméra.

→ M Ouvrir l'onglet `QUANTIF./ECHANT.` puis sélectionner le mode `IMPACT DE LA QUANTIFICATION`.

→ M Modifier la profondeur de gris et visualiser l'effet sur l'image et sur l'histogramme après traitement.

→ Q Que peut-on conclure sur l'effet de la quantification sur l'image ?

→ M De la même façon, avec le mode `IMPACT DE L'ÉCHANTILLONNAGE`, modifier le nombre de pixels de sous-échantillonnage.

On parle ici d'un phénomène de **binning**. La résolution de l'image est "dégradée" numériquement dans ce cas et les nouveaux pixels affichés sont la moyenne de $N \times N$ pixels de l'image initiale.

Dans le cas présent, ce phénomène peut simuler le changement de résolution de la caméra sur l'acquisition d'une image numérique.

→ Q Que peut-on conclure sur l'effet de la résolution de la caméra sur l'image ?

B - Outils numériques de base [30 min]

Dans cette section, nous allons nous intéresser à quelques fonctionnalités permettant de **manipuler des images** pour les rendre utilisables : amélioration du contraste, seuillage, suppression du bruit...

Contraste et Luminosité

→ M Placer un cube de couleur dans le champ de la caméra. Sélectionner une zone d'intérêt d'environ 500 pixels par 500 pixels autour de l'objet à visualiser. Ajuster le temps d'intégration pour obtenir un histogramme dont le pixel maximum a une valeur de l'ordre des 2/3 de la valeur maximale de la caméra.

→ M Ouvrir l'onglet `PRÉ-TRAITEMENT` puis sélectionner le mode `CONTRASTE / LUMINOSITÉ`.

→ M Modifier les valeurs de contraste et de luminosité de l'image.

→ Q Quelles sont les opérations mathématiques réalisées sur les pixels par ces deux fonctionnalités ? Vous pourrez vous appuyer sur les histogrammes des images brutes et modifiées pour analyser vos résultats.

→ M Avec le mode `AMÉLIORATION DU CONTRASTE`, tester l'effet des deux curseurs.

- Q Proposer une interprétation de l'opération effectuée sur chacun des pixels.

Seuillage

- M Ouvrir l'onglet `PRÉ-TRAITEMENT` puis sélectionner le mode `SEUILLAGE`.
- M Sélectionner le seuillage *Normal* et modifier la valeur du seuil.
- Q Que pouvez-vous conclure sur l'intérêt du seuillage? Vous pourrez essayer avec des objets de taille, de forme et de couleurs différentes.
- M Tester également le mode *Inversé* et *Double*.
- Q Que pouvez-vous conclure sur ces deux modes?

Filtrage

- M Ouvrir l'onglet `FILTRES` puis sélectionner le mode `FILTRE DE LISSAGE`.
- M Sélectionner le filtre *Blur Moyen* et un noyau de taille 15.
- Q Que se passe-t-il sur l'image? Vous pourrez également vous appuyer sur la différence entre l'image de base et l'image modifiée, en cliquant sur l'option *Image - Effet*, pour analyser les effets sur l'image.
- Q Quel est l'effet de la taille du noyau sur le filtrage?
- Q Qu'en est-il avec les filtres de type *Blur Gaussien* et *Médian*?

Les aspects théoriques liés au filtrage de données (signaux et images) sont abordés dans les modules MATHS ET SIGNAL (semestre 5) et TRAITEMENT DU SIGNAL (semestre 6).

La mise en oeuvre de ces filtres sur des images sera abordée en TD de ce module et également dans des modules de traitement d'images dans vos prochaines années de formation.

C - Contrôle de l'uniformité de l'éclairage [20 min]

L'**éclairage** joue un rôle central dans tout système de vision industrielle, car il conditionne directement la qualité des images et, par conséquent, la fiabilité des algorithmes d'inspection ou de détection. Un choix d'éclairage adapté permet de révéler les caractéristiques pertinentes d'une scène — contrastes, reliefs, défauts de surface, contours — tout en minimisant les reflets indésirables, les ombres ou le bruit visuel.

Un choix raisonné de l'éclairage constitue un élément déterminant pour garantir la robustesse, la répétabilité et la précision du système de vision industrielle.

Uniformité de l'éclairage EFFI-Ring

Nous allons nous intéresser ici à l'éclairage **Effilux EFFI-Ring**, version RGB et en particulier à l'uniformité de celui-ci en fonction de la distance de travail.

Quelques données sur cette source sont fournies en annexe de ce document.

→ **M** Allumer l'**éclairage annulaire** du banc (trois couleurs). Placer un **fond uniforme** sous l'éclairage (feuille blanche, paillasse...).

→ **M** Sélectionner l'**ensemble du champ visible** par la caméra. Ajuster le temps d'intégration pour obtenir une image non saturée avec un éclairage blanc. Noter le temps d'intégration choisi.

→ **M** Ouvrir l'onglet **OUTILS POUR IMAGES** puis sélectionner le mode **PROFIL DANS L'IMAGE**. Visualiser les profils vertical et horizontal au centre de l'image (environ).

→ **M** Mesurer les niveaux de gris minimal et maximal obtenus pour cet éclairage.

→ **Q** Que pouvez-vous conclure sur l'éclairage à cette distance de travail ?

On souhaite à présent remplir le tableau XX de la feuille de résultats et mesurer l'écart d'intensité relevé dans la zone centrale du champ de vision pour 3 hauteurs de travail différentes.

→ **M** Sélectionner une zone d'intérêt de 400 x 400 pixels au centre du champ de vision de la caméra.

→ **M** Mesurer les niveaux de gris minimal et maximal obtenus pour **3 hauteurs de travail différentes** : 25cm, 12cm et 6cm (environ) entre le fond et la caméra.

Ne déplacer pas la puissance de la caméra. Déplacer un support uniforme sous la caméra (feuille blanche par exemple).

→ **Q** Retrouve-t-on une courbe de réponse proche de celle du constructeur ?

Autres éclairages

Selon la nature de la pièce à analyser (métallique, transparente, texturée...), de son mouvement et du type de défauts ou d'objets à détecter, différentes stratégies d'illumination (lumière rase, diffuse, coaxiale, structurée, stroboscopique...) peuvent être mises en oeuvre.

Une démonstration est possible! environ 10 min

D - Linéarité du capteur [30 min]

On souhaite vérifier la linéarité du capteur en fonction du temps d'intégration, pour les 3 couleurs d'éclairage ainsi que du blanc. On souhaite alors remplir le tableau XX de la feuille de résultats.

→ **M** Placer un cube de couleur dans le champ de la caméra. Allumer l'éclairage annulaire en blanc. Ajuster la zone d'intérêt pour visualiser une zone quasiment uniforme de l'objet. Placer le *black level* à 0.

→ **M** Ajuster le temps d'intégration le plus élevé de vos relevés de mesure pour obtenir un histogramme dont les valeurs de niveau de gris ne dépassent pas les 2/3 de la valeur maximale de la caméra.

→ **Q** Quel type de profil obtient-on ? Quelle forme d'historgramme ?

→ **M** Pour différentes valeurs de temps d'intégration, relever (graphiquement) sur l'historgramme le niveau de gris du pic le plus élevé.

→ **Q** Quelle relation obtient-on entre ce niveau de gris et le temps d'intégration ? Que peut-on en conclure sur le capteur ?

E - Champ de vision et résolution spatiale [40 min]

Le **champ de vision** (*field of view* en anglais) d'un système de vision (champ transversal à l'axe optique) correspond à l'ensemble des points de la scène imagée à travers le système optique (voir figure ??).

Il dépend de :

- la distance focale de l'objectif,
- de la taille du capteur de la caméra,
- de la distance de travail

Selon la dimension de la scène à visualiser, cette grandeur peut s'exprimer par des dimensions (scène à distance finie) ou par des angles (scène à distance infinie).

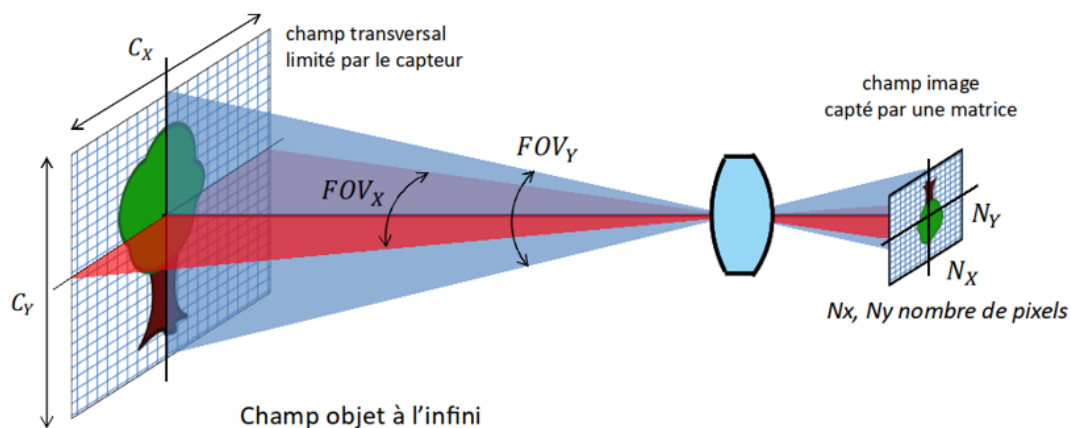


FIGURE 1 – Système optique et champ de vision d'une caméra numérique - illustration provenant du cours d'optique instrumentale de Sébastien de Rossi.

→ **M** Relever sur l'objectif sa focale.

→ **M** Placer une règle graduée sur un fond uniforme dans le champ de la caméra.

- M Sélectionner l'ensemble du champ visible par la caméra. Ajuster le temps d'intégration pour obtenir une image non saturée avec un éclairage blanc. Faire la mise au point sur les graduations de la règle.
- M Mesurer le champ selon la hauteur (C_Y) et la largeur (C_X) de la scène.
- Q A distance de travail équivalente, que devient le champ de vision pour un objectif de focale 2 fois plus courte ?

Quelques **données sur la caméra utilisée sont fournies en annexe** de ce document.

- Q Quelle est la taille d'un pixel ? Quelle est la taille du capteur (hauteur - N_Y - et largeur - N_X) en pixels ? Puis en mm ?
- Q Rappeler les formules de conjugaison et de grandissement en optique instrumentale.
- Q Calculer le grandissement obtenu avec le système optique lorsque l'on mesure un objet positionné sur le plan de travail (distance de travail fixée).
- Q Vérifier par le calcul que la focale mentionnée sur l'objectif est correcte.

-
- Q A partir des dimensions mesurées précédemment (champ de vision) et de la taille du capteur, qu'elle est la taille minimale qu'il est possible de mesurer à l'aide de ce système de vision ?
 - Q Que devient cette valeur si on rapproche l'objet de la caméra en divisant la hauteur de travail par 2 ? Le champ de vision est-il modifié ?

Pour le vérifier, nous allons utiliser des mires de type Foucault. Ces mires sont des répétitions spatiales d'un même motif (succession de bandes noires et blanches) espacé d'une distance connue.

- M Placer une mire dans le champ de la caméra.
- M Ajuster la zone d'intérêt. Ajuster le temps d'intégration pour obtenir une image non saturée avec un éclairage blanc (maximum au 2/3 de l'histogramme). Faire la mise au point sur les graduations de la mire.
- Q Quelle est la plus petite mire que vous arrivez à distinguer ?

-
- M Tracer le profil (horizontal ou vertical) passant par le centre des bandes de la mire.
 - Q Que constatez-vous en fonction du pas de la mire ?
 - M Mesurer la hauteur des variations pour différents espacements des bandes.

En normalisant ces mesures par rapport à la plus grande variation, on peut mesurer le contraste obtenu pour différentes fréquences spatiales de votre système de vision. On peut assimiler à la **fonction de transfert de modulation** (ou FTM). Cette donnée sert à caractériser les systèmes optiques en reliant la luminance de l'espace objet à l'éclairement de l'espace image. Cela permet de modéliser l'influence du système optique sur la distribution de l'énergie lumineuse dans l'espace image.

- M Réduire le temps d'intégration et reprendre les mesures précédentes.
- Q Le contraste est-il dépendant du temps d'intégration de la caméra ?

-
- Q Comment peut-on mesurer des objets à partir d'une image numérique de celui-ci ?
 - M Placer un cube de couleur dont vous aurez au préalable mesuré un des côtés. Tester la méthode proposée pour mesurer numériquement cet objet.
 - Q Quelle est la précision de la mesure ?

F - Première modélisation [50 min]

Dans cette section, nous allons nous intéresser à la visualisation d'objets colorés par l'intermédiaire d'une caméra monochrome et de sources ayant des longueurs d'onde connues et distinctes, afin de donner un premier modèle simplifié d'une chaîne d'acquisition d'image (voir figure ??).

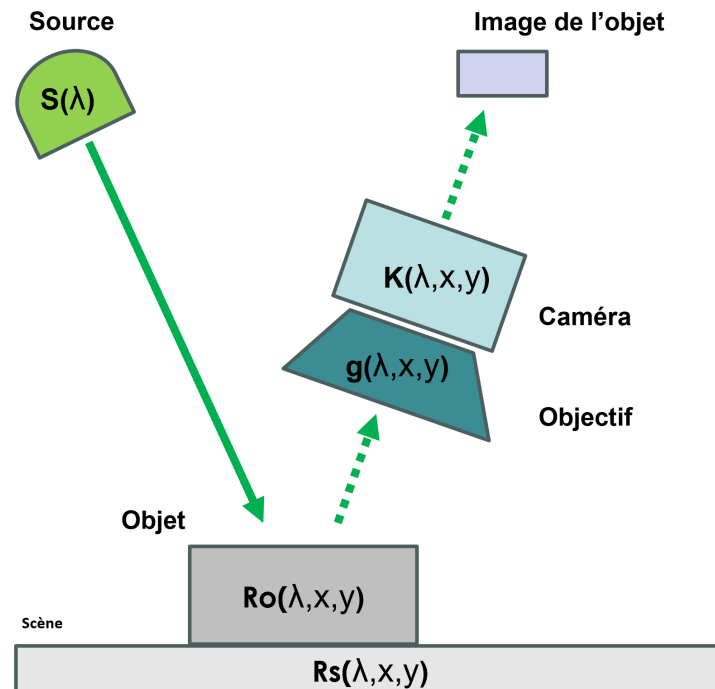


FIGURE 2 – Modélisation simplifiée d'une chaîne d'acquisition d'une image. $S(\lambda)$: source primaire modélisée par son spectre en longueur d'onde (λ). $K(\lambda, x, y)$: le gain de la caméra. $g(\lambda, x, y)$ le facteur de transmission de l'optique (en négligeant pour l'instant les aberrations). $R_o(\lambda, x, y)$: réflectance de l'objet à visualiser. $R_s(\lambda, x, y)$: réflectance des autres objets de la scène (en particulier le fond).

On donne en annexe les réflectances des cubes de couleur que vous avez à votre disposition.

- Q En plaçant chacun des cubes de couleurs sous l'éclairage Rouge, lequel sera le plus "lumineux" ?
- Q Classer, théoriquement, les 4 cubes de couleur par ordre de luminosité pour les 3 éclairages disponibles.
- Q Proposer alors une méthode de discrimination des cubes en fonction de leur couleur à l'aide du système de vision industrielle mis à votre disposition. **Cette méthode doit être robuste à un changement de temps d'intégration.**

→ M Placer les cubes de couleur dans le champ de la caméra l'un après l'autre. Sélectionner une zone d'intérêt d'environ 500 pixels par 500 pixels autour des objets à visualiser.

→ M Pour chaque source (Rouge, Vert, Bleu) et pour chaque cube de couleur (pour un temps d'intégration donné et fixé pour l'ensemble des mesures), relever la valeur moyenne d'intensité lumineuse.

→ Q Les résultats sont-ils cohérents avec la théorie ?

→ M Mettre en oeuvre votre méthode de détection des couleurs et présenter vos résultats.

INTERFAÇAGE NUMÉRIQUE

Travaux Pratiques

Semestre 6

TP2 - Bases de traitement d'images sous OpenCV

Lors de cette séance, vous devrez écrire vos propres scripts en **Python** (avec l'IDE PyCharm par exemple) permettant de réaliser des opérations de base de manipulation d'images, à l'aide notamment de la célèbre bibliothèque **OpenCV**.

Une feuille de résultats est à remplir et à présenter tout au long de la séance de TP.

Ressources

Un tutoriel sur les bases d'OpenCV est disponible à l'adresse suivante :

<https://iogs-lense-training.github.io/image-processing/>

Des codes en Python, proposant des exemples à tester, sont disponibles sur le site du LENS E dans la rubrique *Année / Première Année / Interfaçage Numérique S6 / Bloc 2 Vision Industrielle / Répertoire vers codes à tester*.

Un fichier archivé, nommé `_STEP_BY_STEP.ZIP`, regroupe l'ensemble des codes à tester au cours de cette séance, ainsi que les images à traiter.

Un **kit d'images** est disponible sur le site du LENS E dans la rubrique *Année / Première Année / Interfaçage Numérique S6 / Bloc 2 Vision Industrielle / Kit d'images*.

Accumulation de preuves / Méthode de travail

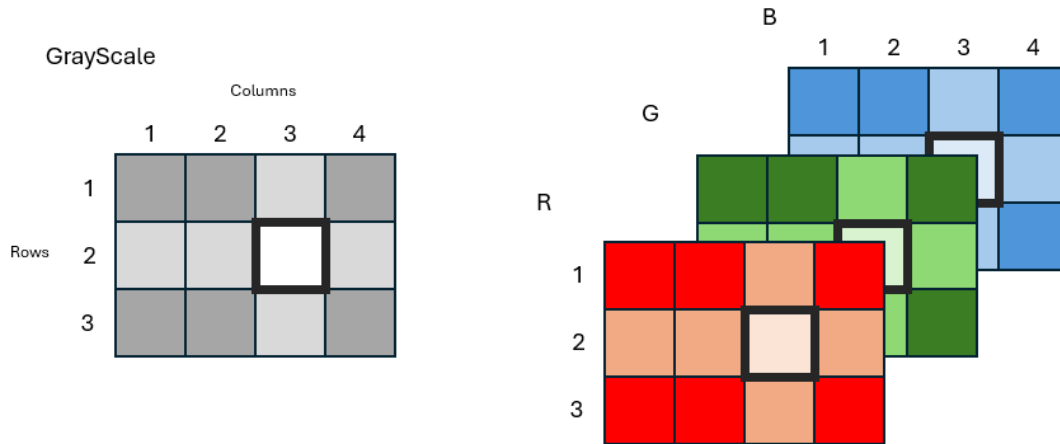
Il est conseillé pour ce TP de **créer un nouveau projet PyCharm** sur votre session (*attention à l'endroit où vous stockerez ce projet - U : sur les sessions Windows de l'IOGS*).

Les différents algorithmes proposés dans ce TP, ainsi que ceux que vous serez amenés à modifier ou créer, pourront vous resservir dans d'autres projets. Nous vous conseillons donc fortement de les **sauvegarder** précieusement et de les **commenter** autant que possible afin de retrouver rapidement les principes mis en jeu derrière les fonctionnalités de OpenCV.

Il serait également pertinent de votre part de rédiger un **journal de bord** sur ce TP en incluant les résultats (images, histogrammes...) et vos analyses des fonctionnalités et de leur intérêt en traitement d'images.

Rappel sur les images numériques

Une image RVB contient 3 canaux (Rouge, Vert, Bleu ou *RGB* en anglais), tandis qu'une image en niveaux de gris n'en a qu'un. Une image en niveau de gris sera **3 fois plus rapide** à analyser qu'une image en couleur RVB mais toute notion de couleur sera alors perdue.



La couleur des objets peut s'avérer inutile lorsqu'on cherche, par exemple, à détecter des formes particulières ou des contours dans une image.

De nombreux algorithmes d'analyse d'image ou de vision par ordinateur travaillent plus efficacement sur des images en niveaux de gris, permettant notamment d'uniformiser l'entrée des algorithmes et de réduire les informations redondantes liées à la couleur.

A - Ouvrir une image [20 min]

- M Ouvrir le fichier `01_OPEN_IMAGE.PY` du répertoire des codes à tester.
- M Exécuter ce code.
- Q Que fait ce programme ? Quelle est la taille de chacune des images ? Quel est le type de données d'un élément ?
- Q Que vaut le premier pixel de chacune des images ? A quoi correspondent les données fournies ?

B - Calculer l'histogramme d'une image et l'afficher [20 min]

- M Ouvrir le fichier `02_HISTOGRAM_IMAGE.PY` du répertoire des codes à tester. Exécuter ce code.
- Q Que fait ce programme ? A quoi correspond la valeur affichée pour la ligne `histogram[100]` ?
- Q A quoi correspond la valeur affichée par l'exécution de la ligne `print(np.sum(histogram))` ?

Il peut être intéressant de **créer une fonction qui affiche automatiquement l'histogramme d'une image à partir de ses données**. Elle sera très utile dans la suite du TP pour voir l'impact des effets appliqués sur les images.

C - Améliorer numériquement la qualité d'une image [20 min]

Pour modifier le contraste et la luminosité d'une image, il faut appliquer une transformation linéaire à chaque pixel de l'image pouvant être exprimé mathématiquement comme suit :

$$P_{new} = \alpha \cdot P_{old} + \beta$$

où α est le facteur de contraste. Une valeur supérieure à 1 augmente le contraste, tandis qu'une valeur entre 0 et 1 le réduit.

β est l'offset de luminosité. Une valeur positive rend l'image plus lumineuse, tandis qu'une valeur négative l'assombrit.

On utilise la fonction `cv2.convertScaleAbs()` pour modifier le contraste et la luminosité de l'image.

→ M Ouvrir le fichier `03_ENHANCE_CONTRAST_BRIGHTNESS.PY` du répertoire des codes à tester. Exécuter ce code.

→ M Modifier le contraste de l'image et comparer les histogrammes de l'image originale et de la version modifiée pour différente valeur de α .

→ Q L'affichage de la nouvelle image est-il correct ?

→ M Réafficher l'image avec les options suivantes de `imshow` : `vmin=0` et `vmax=255`

→ M Calculer la moyenne de tous les pixels de l'image originale et comparer à la moyenne des pixels de l'image modifiée.

→ M Modifier la luminosité de l'image et comparer les histogrammes de l'image originale et de la version modifiée pour différente valeur de β .

→ M Calculer la moyenne de tous les pixels de l'image originale et comparer à la moyenne des pixels de l'image modifiée.

→ Q Que pouvez-vous conclure sur les effets du contraste et de la luminosité ?

→ M Ouvrir une image RGB et appliquer les mêmes effets. Que pouvez-vous conclure ?

D - Binariser l'image [20 min]

La binarisation d'une image consiste à transformer une image en niveaux de gris (ou parfois en couleur) en une image ne contenant que deux valeurs possibles : généralement 0 (noir) et 1 (blanc).

L'objectif de la binarisation est de réduire drastiquement les informations à traiter tout en essayant de séparer clairement le premier plan (objets) de l'arrière-plan.

→ M Ouvrir le fichier `04_THRESHOLD.PY` du répertoire des codes à tester. Exécuter ce code. Modifier la valeur du seuil et comparer les images résultantes.

→ Q Que fait ce code ? Est-ce que le choix du seuil a un impact sur la valeur résultante pour la méthode d'Otsu ?

Vous trouverez un peu plus de détails sur les méthode de binarisation à l'adresse suivante :

https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html

→ M Ouvrir une image RGB et appliquer les mêmes effets. Que pouvez-vous conclure ?

E - Opérations de pré-traitement [40 min]

Les opérations de pré-traitement dans le traitement d'images sont essentielles pour améliorer la qualité des images avant d'appliquer des algorithmes plus complexes, comme la segmentation, la détection d'objets ou la classification. Ces étapes de pré-traitement visent à réduire le bruit ou améliorer la structure de l'image.

Parmi les opérations de pré-traitement classiques, on peut citer :

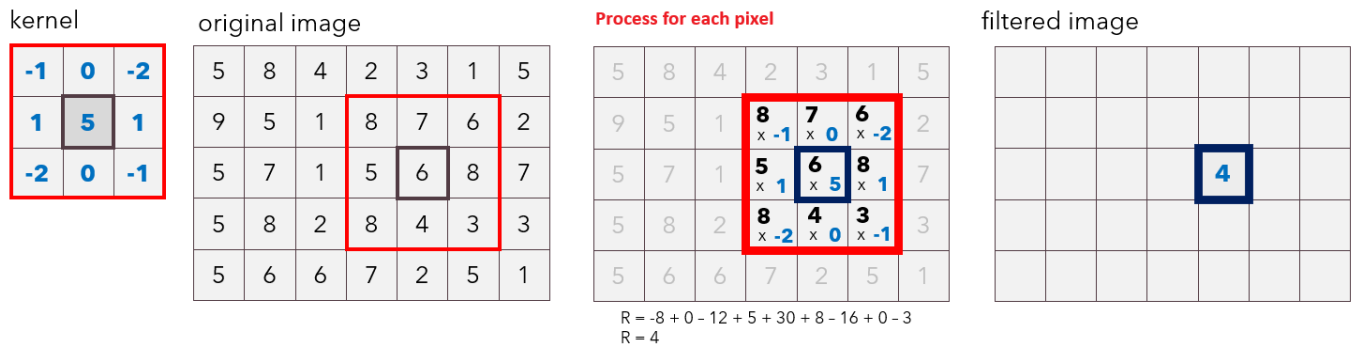
- **Correction des couleurs** : Balance des blancs, Correction gamma, Amélioration de contraste...
- **Réduction de bruit** : Filtrage linéaire pour atténuer les bruits sans trop affecter les détails importants de l'image, Filtrage non linéaire pour éliminer les bruits impulsionsnels, Filtrage anisotrope...
- **Opérations morphologiques** : érosion pour éliminer du bruit, dilatation pour combler des lacunes dans les objets, ouverture et fermeture pour enlever les petites anomalies ou remplir les petits trous dans une image
- **Filtrage fréquentiel** pour éliminer ou atténuer des fréquences particulières (comme des motifs de bruit répétitifs)

Éléments structurants d'une opération morphologique (noyau)

Notions : *Éléments structurants*

Les **transformations dites morphologiques** se basent sur l'application d'un **élément structurant** (ou noyau) que l'on va superposer sur chaque pixel de l'image.

Un **élément structurant** (ou noyau) est une petite matrice (généralement de taille et de forme prédéfinies, comme un carré, un disque, une ligne, etc.) qui sert de sonde pour inspecter et modifier les pixels d'une image.



Les éléments structurants jouent un rôle clé en traitement d'image, notamment dans les opérations de morphologie mathématique. Ces opérations sont principalement utilisées pour analyser et traiter des images binaires ou en niveaux de gris en modifiant leurs formes ou en extrayant des structures spécifiques.

- M Ouvrir le fichier 05_KERNEL.PY du répertoire des codes à tester. Exécuter ce code.
- Q Quel est le type de des objets noyaux résultants ?
- M Générer un noyau en forme de croix de taille 3 par 3 pixels et afficher ce noyau.
- M Générer un second noyau en forme de cercle (ellipse) de taille 5 par 5 pixels et afficher ce noyau.
- M Générer une ellipse de taille 31 par 21 pixels et afficher ce noyau.

Opérations d'ouverture et de fermeture

Notions : *Opening - Closing*

- M Ouvrir le fichier 07_OPENING_CLOSING.PY du répertoire des codes à tester. Exécuter ce code.
 - Q Quelles sont les opérations effectuées sur l'image *a_letter_noise.jpg*? Quels sont les noyaux utilisés pour ces opérations?
 - Q De quel type est l'image utilisée pour ces opérations?
 - M Modifier la taille des noyaux et tester à nouveau ces opérations.
 - Q Que pouvez-vous conclure sur l'utilité des opérations d'ouverture et de fermeture sur une image? Sur l'impact de la forme et de la taille du noyau?
-

- M Ouvrir une image RGB (*formes_bruit.png* par exemple) et appliquer les mêmes effets. Que pouvez-vous conclure?
- M Ouvrir cette même image en niveau de gris et appliquer les mêmes effets. Que pouvez-vous conclure?

Opération de gradient

- M Ouvrir le fichier 08_GRADIENT.PY du répertoire des codes à tester. Exécuter ce code.
- Q Quelle est l'opération réalisée sur l'image *grad_image_cross_3*? Quelle est la suite d'opérations réalisée sur l'image *grad_sub_image_cross_3*?
- Q Que pouvez-vous conclure sur le lien entre ces deux méthodes?
- M Proposer une méthode "fiable" pour comparer facilement ces deux images. Mettre en oeuvre cette méthode.
- M Modifier la taille des noyaux et tester à nouveau ces opérations.
- Q Que pouvez-vous conclure sur l'utilité de l'opération de gradient sur une image? Sur l'impact de la forme et de la taille du noyau?

F - Calculer la FFT d'une image [30 min]

→ M Ouvrir le fichier 09_FFT.PY du répertoire des codes à tester. Exécuter ce code.

→ Q Quelle est la taille de l'image initiale? Quel est le type et quelle est la taille de l'objet *fft_image*? De quel type sont ses éléments?

→ Q Expliquer l'intérêt de l'affichage du logarithme de la valeur absolue de la FFT.

→ Q Que permet de faire la fonction *circular_mask* sur la FFT? Commenter le résultat obtenu sur l'image recréée à partir de cette FFT modifiée.

→ Q Quel est l'impact de la taille du masque? A quoi sont dues les "vaguelettes" que l'on aperçoit sur l'image résultante?

→ Q Que se passe-t-il sur une image RGB? Quelle est la taille de la FFT?

→ M Générer une trame sinusoïdale en deux dimensions (200 x 300 pixels) à l'aide de la fonction *sine_trame* de la bibliothèque *images_manipulation*. Calculer sa FFT et afficher les résultats.

→ Q Commenter le résultat obtenu pour différentes périodes spatiales et différents angles.

G - Appliquer un filtre moyenneur sur une image [30 min]

L'utilisation de filtres permet de prendre en compte les pixels voisins en exploitant les relations locales dans une image, ce qui est crucial pour des tâches comme la suppression de bruit, la détection de contours, l'extraction de caractéristiques, et l'amélioration de la qualité visuelle. Travailler sur des pixels isolés limite l'analyse à des informations ponctuelles, tandis que considérer les voisins permet une compréhension plus riche et contextuelle de l'image.

→ M Ouvrir le fichier 10_BLUR_MEAN.PY du répertoire des codes à tester. Exécuter ce code.

→ Q Commenter les effets des deux méthodes utilisées pour lisser l'image. Quelle méthode est utilisée ici pour montrer de manière objective les modifications apportées à l'image?

→ Q Que se passe-t-il sur une image RGB?

Le fichier 10A_BLUR_MEAN_SLICE.PY propose une autre méthode pour comparer les effets du lissage.

→ M Ouvrir le fichier 10A_BLUR_MEAN_SLICE.PY du répertoire des codes à tester. Exécuter ce code.

→ Q Quelle est cette méthode?

→ M Calculer et afficher la FFT de chacune des images précédentes (originale et modifiées).

→ Q Quelle est la fonction réalisée par les deux filtres précédents?

On cherche à présent à voir l'impact de la taille du noyau sur l'image finale.

→ M Ouvrir le fichier 10B_BLUR_MEAN_KERNEL.PY du répertoire des codes à tester. Exécuter ce code.

→ Q Que pouvez-vous conclure sur l'impact de la taille du noyau? Vous pourrez également comparer les FFT...

H - Appliquer un filtre passe-haut sur une image [40 min]

Le **filtre moyenneur** (vu précédemment) permet de conserver les **éléments à basse fréquence spatiale** dans l'image. C'est une méthode intéressante pour supprimer des bruits ponctuels (des éléments isolés et donc "rapides"). Il est également possible en choisissant un autre élément structurant de réaliser l'opération complémentaire qui supprime le fond continu et ne conserve que les transitions de fréquence spatiale élevée (bords d'un objet par exemple).

Il est possible d'utiliser la fonction `cv2.filter2D()` pour appliquer un noyau particulier sur une image.

Opérateur de Roberts

L'**opérateur de Roberts** est l'un des premiers filtres de **détection de contours**. Il repose sur la convolution avec deux petits noyaux 2x2, conçus pour approximer les dérivées en diagonale de l'image.

Les noyaux de convolution sont les suivants :

$$K_x = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}, \quad K_y = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}.$$

Il est alors possible de calculer l'amplitude du gradient par l'opération suivante :

$$\text{Amplitude} = \sqrt{G_x^2 + G_y^2}$$

où G_x est le résultat de la convolution de l'image par le noyau K_x et G_y le résultat de la convolution de l'image par le noyau K_y .

Une forte amplitude indique un contour ou un bord marqué. Une faible amplitude indique une région où l'intensité est relativement constante.

- M Ouvrir le fichier `12_ROBERTS.PY` du répertoire des codes à tester. Exécuter ce code.
- Q Commenter les résultats. Que pouvez-vous conclure sur l'effet de ce filtre ?
- M Appliquer cet effet sur l'image `noise_vi_small.png`.
- Q La détection de contours est-elle optimale ?

→ M Appliquer un filtre gaussien à l'aide d'un noyau de taille 11 x 11 et d'écart-type de 2. Appliquer l'effet précédent à l'image résultante.

- Q La détection de contours est-elle améliorée ?

Opérateur de Sobel

L'**opérateur de Sobel** permet de réaliser une opération similaire à celui de Roberts, mais en étant moins sensible aux bruits dans l'image, puisqu'il se base sur un noyau plus large et ainsi lisse l'image dans la direction perpendiculaire au gradient mesuré.

Les noyaux de convolution sont les suivants :

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

De la même façon que précédemment, on peut calculer l'amplitude du gradient par l'opération suivante :

$$\text{Amplitude} = \sqrt{G_x^2 + G_y^2}$$

où G_x est le résultat de la convolution de l'image par le noyau K_x et G_y le résultat de la convolution de l'image par le noyau K_y .

- M Ouvrir le fichier 12_SOBEL.PY du répertoire des codes à tester. Exécuter ce code.
- M Visualiser l'image originale, le résultat du filtrage selon X et le résultat du filtrage selon Y.
- Q Commenter les résultats. Que pouvez-vous conclure sur l'effet de ce filtre ?
- M Appliquer cet effet sur l'image *noise_vi_small.png*.
- Q La détection de contours est-elle optimale ?

→ M Appliquer un filtre gaussien à l'aide d'un noyau de taille 11 x 11 et d'écart-type de 2. Appliquer l'effet précédent à l'image résultante.

- Q La détection de contours est-elle améliorée ?

-
- Q Que se passe-t-il sur une image RGB ?

Méthodes de Harris et de Canny

L'**algorithme de Canny** est une technique de détection qui a pour objectif d'identifier les bords significatifs d'une image tout en minimisant le bruit.

Cet algorithme comprend les étapes suivantes : **lissage** (réduction du bruit avec un filtre gaussien), **calcul du gradient** (obtention de la magnitude et de la direction des contours), **suppression non maximale** (affinage des contours pour ne garder que les crêtes locales) et **seuils double** (application de deux seuils pour conserver les contours forts et connecter les contours faibles pertinents).

L'**opérateur de Harris** permet de détecter des coins ou points d'intérêt dans une image, utiles pour la reconnaissance d'objets, le suivi de mouvement ou la reconstruction 3D.

Il est basé sur la mesure de la variation de l'intensité d'image dans des zones de l'image (par convolution avec un noyau). Les coins correspondent à des régions où la variation de l'intensité est forte dans plusieurs directions. Enfin un calcul d'une matrice de structure et d'une fonction de réponse R pour identifier les coins significatifs est réalisé.

- M Ouvrir le fichier 13_HARRIS_CANNY.PY du répertoire des codes à tester. Exécuter ce code.
- Q Commenter les résultats.
- M Appliquer cet effet sur d'autres images.