

# Institut de la Francophonie pour l'informatique

## Vision par ordinateur

### TP1 : Detection et reconnaissance d'objets en utilisant des descripteurs locaux

(Travail en group de 2-3 personnes)

*Date de remise : 23h 18/03/2025*

*Remise : course on smartlink*

## Détecteur de points d'intérêt SIFT et descripteur SIFT

Pour ce TP, vous allez utiliser les descripteurs locaux extraits aux points d'intérêt pour détecter et reconnaître des objets. Vous pouvez utiliser descripteurs SIFT ou d'autres descripteurs locaux comme vous voulez. Bien que cette méthode soit a priori difficile à comprendre d'un point de vue théorique, son utilisation est très simple grâce à la démo disponible sur la page de David Lowe (<http://www.cs.ubc.ca/~lowe/keypoints/>), ou bien vous pouvez utiliser la version de SIFT implémentée dans la bibliothèque OpenCV: [https://docs.opencv.org/4.x/da/df5/tutorial\\_py\\_sift\\_intro.html](https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html). N'hésitez pas à télécharger et à lire les articles de recherche disponibles sur la page de David Lowe. Outre les explications théoriques, on y trouve des explications pratiques et simples à comprendre sur l'utilisation de SIFT pour plusieurs applications.

### 1) Localisation d'objets avec SIFT:

Pour cette partie, on vous demande de détecter l'objet en utilisant les descripteurs locaux:

- L'entrée : une image contenant un objet isolé (O) et une image de scène (I) contenant plusieurs objets.
- La sortie : localisation des objets O dans l'image I.

Vous devez considérer deux situations:

- o L'image I ne contient pas l'objet O
- o L'image I contient seulement un objet O
- o L'image I contient quelques objets O

### Calcul des descripteurs SIFT

Vous allez extraire les descripteurs SIFT de chaque image. Chaque descripteur SIFT consiste en un vecteur de 128 valeurs. Pour une seule image, il est possible d'extraire des dizaines ou même des centaines de points SIFT.

### Mise en correspondance (matching) de points d'intérêts

Étudiez la méthode de mise en correspondance présentée dans la démo SIFT (et aussi présentée en classe). Il s'agit d'une méthode possible, mais pas la seule. [https://docs.opencv.org/4.x/dc/dc3/tutorial\\_py\\_matcher.html](https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html)

La mise en correspondance de deux images se fait en calculant les correspondances entre tous les descripteurs SIFT de ces deux images. La distance entre deux vecteurs SIFT peut se faire simplement en prenant la distance Euclidienne entre les deux vecteurs. On considère la distance la plus courte (petite) entre deux vecteurs comme la meilleure. C'est-à-dire, chaque descripteur  $f_l$  (associé à un

point d'intérêt) dans la 1ere image est apparié avec un descripteur f2 (associé à un autre point d'intérêt dans la 2eme image) qui ont la distance minimale avec f1 en comparant avec d'autres descripteurs dans la 2eme image.

Afin de réduire le nombre de fausses correspondances, une métrique possible est de calculer le ratio entre la distance la plus courte et la deuxième plus courte distance. Si ce ratio est inférieur à un seuil (à déterminer), alors la correspondance peut être considérée comme robuste, sinon elle est rejetée :

$$ratio = \frac{d_{plusproche}}{d_{deuxièmeplusproche}} < seuil$$

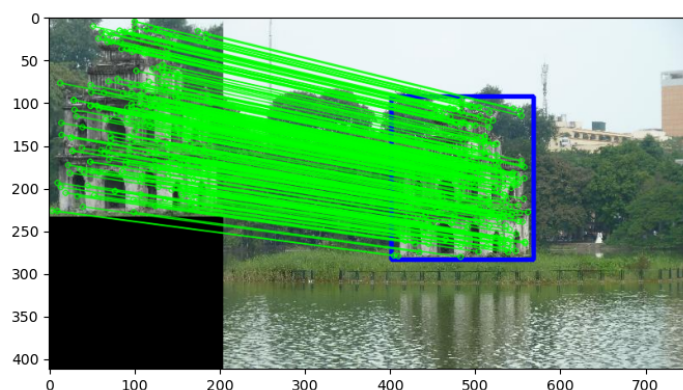


## Localisation d'objets avec SIFT

Supposer que vous avez l'image O qui contient un objet (objet de recherche), et l'image I (image de scène) qui contient peut-être plusieurs différents objets. Vous devez localiser la position de cet objet dans l'image I (comme illustrer dans l'image suivante).

Pour le faire, il faut :

- Mettre en correspondance des descripteurs locaux entre 2 images pour trouver des paires des points d'intérêt associés.
- Chercher la transformation homographie (H) entre l'ensemble des points d'intérêt de l'image O avec l'ensemble des points d'intérêt associés dans l'image I.  
([https://docs.opencv.org/3.4/d1/de0/tutorial\\_py\\_feature\\_homography.html](https://docs.opencv.org/3.4/d1/de0/tutorial_py_feature_homography.html) )
- Calculer les positions transformées dans l'image O des 4 coins de l'objet de requête dans l'image I.



Dans l'image I (l'image de scene), peut-être il n'existe pas l'objet de requête. Dans ce cas, vous devez préciser l'absence de cet objet dans l'image de recherche. Une solution simple est d'calculer le score de correspondances réussies entre 2 images. Plus ce nombre de correspondances réussies, plus les deux images ont des chances de représenter la même classe d'objets. Et puis, vous pouvez mettre un seuillage sur ce score pour décider la présence ou l'absence de l'objet. Si ce score est inférieur à un seuil (*seuil\_score*), l'objet n'est pas apparu dans l'image de recherche. Sinon, localiser-le dans cette image. Le score peut être calculé comme suit :

$$score = \frac{\# \text{ correspondances réussies}}{\# \text{ descripteurs de l'objet de l'objet}} < \text{seuil\_score}$$

**Remarque :** Les méthodes décrites ci-dessous sont simple. Elles ne sont ni complète ni parfaite. Il vous restera plusieurs points à résoudre pour bien réussir ce TP. Je vous encourage fortement à étudier les différentes solutions et idées existantes sur Internet. Dans votre rapport, expliquez-bien les idées que vous avez implémentées, en citant les références extérieures au besoin.

**Analyser:** Dans votre rapport, montrez bon et mauvais resultats. Expliquez les raisons qui donnent des ces resultats.

## 2) Reconnaissance d'objets avec SIFT

Le but de cette partie consiste à faire la reconnaissance d'objets à l'aide du descripteur SIFT. Premièrement, il faut extraire les points SIFT d'images de référence (d'apprentissage) représentant plusieurs classes d'objets. Ensuite, à partir d'une image représentant un objet inconnu, on extrait les points SIFT et on compare avec les classes connues pour identifier l'objet.

- Entree: Ensemble d'objet avec la verite-terrain; objet inconnu.
- Sortie: la classe de l'objet inconnu.

### Bases d'images

Deux bases d'images sont proposées pour ce TP (voir ci-dessous). Vous en choisissez une, celle de votre choix. Quel que soit la base d'images utilisée, séparez-la (environ) 50% base de référence (apprentissage) et 50% base de test.

Chaque base d'images possède plusieurs d'images. Votre programme devra être capable de traiter automatiquement toutes les images. Voici trois conseils pour vous aider :

- Précalculer les vecteurs SIFT à l'avance et travailler ensuite sur les fichiers texte contenant ces descripteurs
- Chaque image comporte déjà sa classe d'appartenance, soit par son nom de fichier, soit par son nom de répertoire. Utilisez ces noms lors de la construction de votre base de référence et lors de la phase de validation.

**Calcul des descripteurs SIFT et Mise en correspondance (matching) de points d'intérêts : voir la 1ere partie.**

### Calcul de la similarite entre 2 images

Une fois la correspondance entre tous les points SIFT calculés, on peut compter le score de correspondances réussies. Plus ce nombre de correspondances réussies, plus les deux images ont des chances de représenter la même classe d'objets. Le score peut être calculé comme suit :

$$score = \frac{\# \text{ correspondances réussies}}{\# \text{ descripteurs de l'image modèle}}$$

« L'image modèle » étant l'image de la base d'apprentissage. On comptera donc ce score de correspondances pour l'image inconnue et pour toutes les « images modèles » de la base d'apprentissage. Les N images ayant le score le plus élevé détermineront la classe de l'objet inconnu (N=1, 3, 5, ...).

## **Analyser :**

Dans votre rapport, montrez quelques résultats de mise en correspondances, bon et mauvais. Expliquez la composition de vos bases d'apprentissage et de test.

Calculez et montrez-la (les) **matrice(s) de confusion** (si vous ne connaissez pas la matrice de confusions, n'hésitez pas à consulter sur wikipédia) sur votre base d'images. Il vaut mieux de faire une représentation par couleur plutôt que par valeur. Expliquez les classes qui fonctionnent bien et celles qui se confondent facilement. Donnez (et surtout expliquez) des exemples qui fonctionnent bien et d'autres qui fonctionnent moins bien. Analysez et expliquez pourquoi vous obtenez ces résultats.

Expliquez bien les idées que vous avez implémentées et qui ne sont pas expliquées dans cet énoncé. Si vous avez d'autres idées pour améliorer les résultats, discutez-les. Donnez les références si nécessaire.

## **Bases d'images pour ce TP**

Choisissez UNE base d'images parmi les suivantes. Ces bases contiennent plusieurs classes (~100), vous pouvez sélectionner un sous-ensemble de 10 à 15 classes afin de faciliter l'analyse des résultats.

- Columbia University Image Library (COIL-100) :  
<https://www.kaggle.com/datasets/jessicali9530/coil100/data>
- Caltech 101: <https://www.vision.caltech.edu/datasets/>

***Bon travail !***