

# Institut de la Francophonie pour l'informatique

## Vision par ordinateur

### TP 2 : Reconnaissance d'objets avec le descripteur SIFT

**(Binome – Le partenaire doit être différent du TP1)**

*Date de remise : 23h 30/07/2022*

*Remise : sur le serveur du cours en ligne*

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

Pour ce tp, vous allez utiliser les points d'intérêt calculés par la méthode SIFT (voir cours sur les points d'intérêt). 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/>).

Pour commencer, allez sur cette page, télécharger la démo et essayez-là avec les exemples donnés. Avec le code (en C), vous comprendrez comment calculer les points d'intérêt et comment faire la correspondance (*matching*) entre deux points. Tout y est. Cependant, vous n'avez pas accès à tout le code (pour raison de brevet). Mais les exemples donnés vous permettent de calculer les points d'intérêts dans un fichier, puis vous devrez lire ce fichier texte dans votre programme. Ne cherchez pas à tout inclure dans un seul programme, mais procédez en deux étapes comme dans la démo. Vous pouvez appeler le calcul de SIFT par des appels `system` dans votre code, ou en précalculant toutes les valeurs dans des fichiers texte par exemple.

Actuellement, SIFT est aussi implémenté dans OpenCV, donc vous pouvez l'utiliser à la place du fichier exécutable du David Lowe. Donc, c'est à votre choix.

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.

## Reconnaissance d'objets avec SIFT

**Remarque :** La méthode décrite ci-dessous est simple. Elle n'est 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. Il existe une documentation abondante disponible pour ce tp, donc profitez-en. Dans votre rapport, expliquez-bien les idées que vous avez implémentées, en citant les références extérieures au besoin.

Le but de ce tp 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.

### Bases d'images

Deux bases d'images sont proposées pour ce tp (voir ci-dessous). Vous en choisissez une, celle de votre choix. Quelque 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 milliers 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
- Votre programme ne traite qu'une seule image à la fois. Faites un script Linux pour traiter toutes les images de la base. Vous éviterez ainsi bien des problèmes.
- 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

Vous allez extraire les descripteurs SIFT de toutes les images de la base d'images. 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. Pour stocker ce vecteur, vous pouvez les séparer par image (un fichier texte par image avec tous les points SIFT de cette image) ou dans un seul fichier texte pour toutes les images (préfixer chaque ligne par le nom de l'image).

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

Etudiez la méthode de mise en correspondance présenté dans la démo SIFT. Il s'agit d'une méthode possible, mais pas la seule. 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.

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 :

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

**Remarque :** Comme indiqué auparavant, il s'agit d'un exemple de mesure possible (pas forcément la meilleure) pour déterminer les bonnes et les mauvaises correspondances. Il existe d'autres façons de déterminer cela et vous êtes libre d'utiliser une autre méthode, tant que vous l'expliquez dans votre rapport.

### Calcul de correspondance entre 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,...).

**Remarque :** Même remarque que ci-dessus pour la mesure du score.

## Rapport à remettre

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 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 :

- Columbia University Image Library (COIL-100)
- Caltech 101

**Note (1) :** Pour ceux qui veulent tester et comparer leur programme sur deux bases d'images différentes, un bonus sera accordé.

**Note (2) :** Pour convertir les images TIFF, JPEG ou PNG en PGM, je vous suggère l'utilitaire convert de la suite ImageMagick. Vous pouvez aussi utiliser cet utilitaire pour convertir vos images couleurs en niveaux de gris ou modifier la résolution des images.

**Bon travail !**