INSTITUT FRANCOPHONE INTERNATIONAL

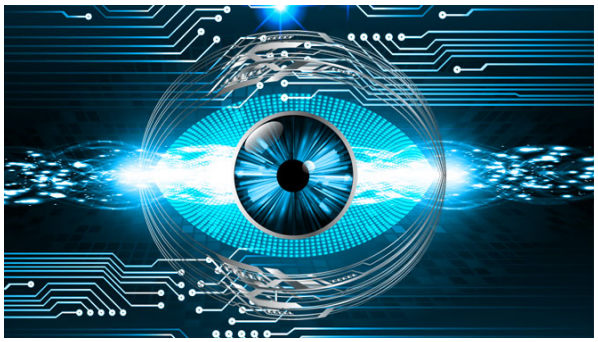
VNU ĐẠI HỌC QUỐC GIA HÀ NỘI Vietnam National University, Hanoi

VISION PAR ORDINATEUR

Rapport de TP1

IFI-Promotion 23

Reconnaissance d'objets avec le descripteur SIFT



Auteurs :

M. LAMAH Richard

M. LAMAH Henry Kpakilé

Encadrants :

Dr. Nguyen Thi Oanh

Table des matières

1 INTRODUCTION

L'utilisation de logiciels pour analyser le contenu visuel mondial est une révolution informatique aussi importante que le mobile il y a 10 ans. Elle offrira aux développeurs et aux entreprises un avantage majeur pour la création de produits étonnants. La vision par ordinateur consiste à utiliser des machines pour comprendre et analyser des images (photos et vidéos). Bien que ces types d'algorithmes existent sous diverses formes depuis les années 1960, les récents progrès en matière d' apprentissage automatique , ainsi que les avancées en matière de stockage de données, de capacités informatiques et de périphériques d'entrée de haute qualité et bon marché, ont permis d'améliorer considérablement la qualité de nos logiciels. peut explorer ce genre de contenu.

1.1 Objectif

L'objectif de ce TP consiste à faire la reconnaissance d'objets à l'aide du descripteur SIFT. Dans un premier temps, nous allons 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.

1.2 Problématique

Naturellement, tout travail de recherche tourne autour de certaines problématique.Cependant, nous n'avons pas faits d'exception 'a cela car nous avons rencontre assez de difficultés qui sont entre autres :

- La mise en correspondance des images avec l'application de la distance
- Le choix des paramètres dans la recherche des correspondants
- Pour l'évaluation de nos résultats, l'utilisation de la matrice de confusion n'a pas ete chose facile.
- etc ...

1.3 Contexte

Le recours aux Nouvelles Technologies de l'information et de la Communication (NTIC) dans le monde du travail est en plein essor. En considérant le nombre d'emplois engendrés

par la conception et la maintenance de ces technologies ou les investissements des entreprises dans ces nouveaux outils comme un indice de l'importance du secteur, on peut souligner l'élévation constante de leur champ d'application et de leur poids dans le monde du travail. Mais nous nous baserons particulièrement sur les données images ici, dans ce cas-ci,cependant, pour notre module de Vision par Ordinateur, il nous a été demandé dans le cadre du TP, d'implémenter un programme de Reconnaissance d'objets avec le descripteur SIFT.

1.4 Fonctionnement Générale du Système

Dans cette section, nous allons décrire le fonctionnement général de notre système qui permet de faire la Reconnaissance d'objets avec le descripteur SIFT. La reconnaissance d'objets, nous permet de détecter la présence d'une instance ou d'une classe d'objets, dans une image ou une scène naturelle. Selon Neisser, notre capacité à reconnaître un objet consiste en deux étapes : un processus de sélection pour extraire les informations les plus pertinentes, et une chaîne complexe des processus pour identifier l'objet. En effet, Pour ce faire avec notre système, nous avons utilisé une base d'images qu'on a divisé en deux grandes parties, le training (4800 images) et le test (2400 images). Quant au training, elle est utilisée pour extraire les descripteurs des images d'entraînement. Ensuite, dans la partition test, on choisit une image comme image d'entrée dans dans le système, son descripteur est ensuite extrait pour déterminer les descripteurs avec les quels il correspond le mieux parmis les descripteurs d'entraînement. Ainsi, le système évalue le pourcentage de correspondance en fonction de sa catégorie.

2 Fonctionnement du programme

Pour le fonctionnement de notre programme, il faut télécharger le Dataset sur ce lien coil-10 et enregistrer dans le répertoire du projet.Ce programme fonctionne sous l'environnement Python incluant OpenCV.Positionnez vous dans le répertoire du projet, ouvrer un terminal et exécute la fonction **python3 ./app.py**.De là,un menu se présente qui vous guidera dans la suite de l'exécution du programme.

2.1 Création de la Base de Données

Nous avons choisi la base d'images sur le lien suivant : coil-10.Après avoir télécharger cette base nous avons diviser en deux sous base nommé "test" et "training" respectivement pour les données de test et d'entraînement grâce à la fonction **shutil.copy()** du Python.

2.2 Détection des points d'intérêts

La détection des points d'intérêts (ou coins) est, au même titre que la détection de contours, une étape préliminaire à de nombreux processus de vision par ordinateur . Les points d'intérêts, dans une image, correspondent à des doubles discontinuités de la fonction d'intérêts. Celles-ci peuvent être provoquées, comme pour les contours, par des discontinuités de la fonction de reflectance ou des discontinuités de profondeur . ci-dessous des exemple : les jonctions en T (première image),les coins (deuxième image) ou les points de fortes variations de texture (troisième image). De façon générale, un détecteur de point d'intérêt consiste à calculer une valeur de réponse représentative de l'intérêt pour chaque pixel de l'image et ensuite à sélectionner les meilleurs.

FIGURE 1 – Exemples des poits d'intérêts

Il existe une quantité importante de détecteurs de points d'intérêt. Nous n'en citons que quelques uns notamment détecteur de Harris, les variantes de **Harris**(Variante de **Plessey**,Variante de **Shi et Tomasi**,Variante de **Schmid**, ...),détecteur **SIFT de Lowe**, detecteur de **Kadir**, ... Comme suggérer dans l'énonce nous utilisons celui de **SIFT** pour notre travail. Le but du détecteur **SIFT** (scale invariant feature transform) présenté dans [**Lowe 99**] par **Lowe** est de localiser des points cles avec un vecteur descripteur afin de pouvoir caractériser un objet et ˆetre capable de le reconnaıtre en comparant les caractéristiques des

points trouvés à une base de données. Nous avons utilisé les fonctions de la librairie Opencv **xfeatures2d.SIFT create()** pour créer l'objet **SIFT** afin d'avoir les points d'intérêts et **drawKeypoints()** pour afficher les points.

FIGURE 2 – les points d'intérêts

Nous constatons que plusieurs points d'intérêts sont bien déterminés ce qui nous rassure que l'algorithme est efficace 'a ce niveau et l'image choisit est aussi bon pour la détermination des points Par contre sur l'image suivante, peu des points d'intérêts ont été déterminer mais on ne met pas en cause l'algorithme car pour avoir des points d'intérêts ,il faut que l'image ne soit pas homogène ce qui fait que l'image de la figure suivante a moins des points d'intérêts.

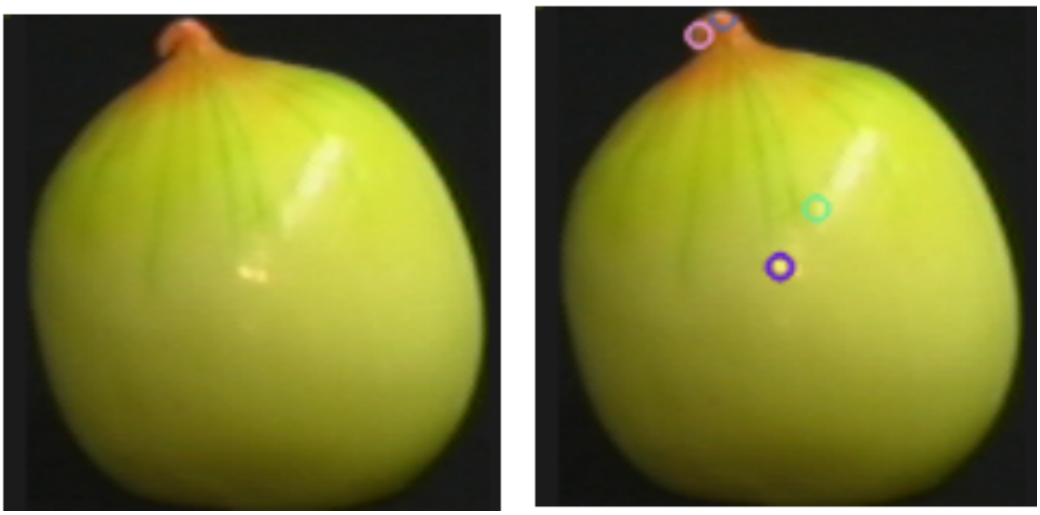


FIGURE 3 – les points d'intérêts

2.3 Calculs des descripteurs SIFT

Le descripteur d'une région d'intérêt autour d'un point détecté représente l'histogramme des orientations des gradients dans la région.Maintenant, pour calculer le descripteur, OpenCV fournit deux méthodes.

1. la fonction **sift.compute()** qui calcule les descripteurs à partir des points clés que nous avons trouves.
2. la fonction **sift.detectAndCompute()** qui recherchez les points d'intérêts directement en une seule étape.

Pour facilter les choses nous avons utilises la fonction **sift.detectAndCompute()** pour déterminer les points d'intérêts et le descripteur **SIFT** à la fois.

Ainsi nous avons la fonction précédente qui dessine les petits cercles sur les emplacements des points-clés. Si nous passons un indicateur, **cv2.DRAW MATCHES FLAGS DRAW RICH KEYPOINTS** ,il va dessiner un cercle de la taille d'un point clé et même montrer son orientation. Voir l'image ci-dessous.

FIGURE 4 – les descripteurs des points d'intérêts

Nous avons donc des points clés, des descripteurs, etc. Maintenant,nous voulons voir comment faire correspondre les points clés dans différentes images. Que nous présenterons dans la partie suivante.

2.4 Mise en correspondance

La mise en correspondance des points d'intérêt peut être utilisée pour estimer les paramètres de la transformation qui relie deux images (comme par exemple une homo-graphie). C'est également, comme nous allons le voir, une étape importante de la mise en correspondance à partir de germes.

2.4.1 Matching

Considérons deux images 1 et 2 d'une scène, il s'agit ici de déterminer, pour un élément de l'image 1, l'élément qui lui correspond dans l'image 2 et éventuellement dans d'autres images. La mise en correspondance de primitives est un problème fondamental de la vision par ordinateur. C'est un processus intermédiaire entre les processus dit de haut niveaux : reconstruction, reconnaissance, etc. et ceux de bas niveaux : extraction d'indices.

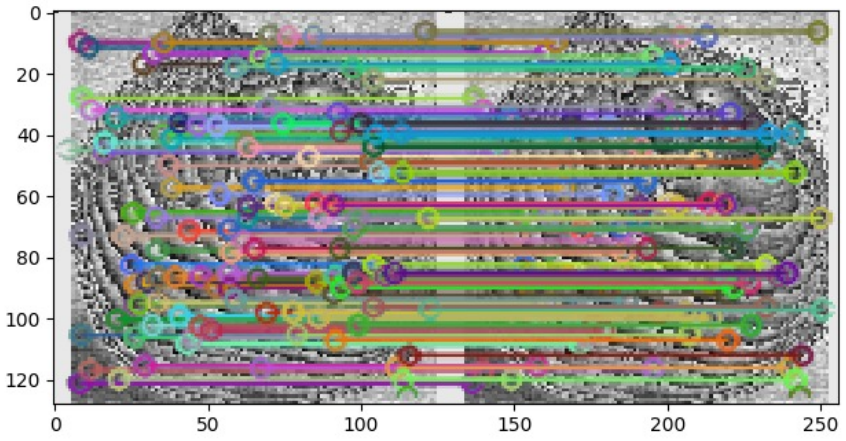


FIGURE 5 – Mise en correspondance d'une image par elle-même

Effectivement, nous remarquons que le nombre des points détectés est le même et les points sont bien mis en correspondance. Donc les points d'intérêts sont bien détectes et la correspondance est respectée.

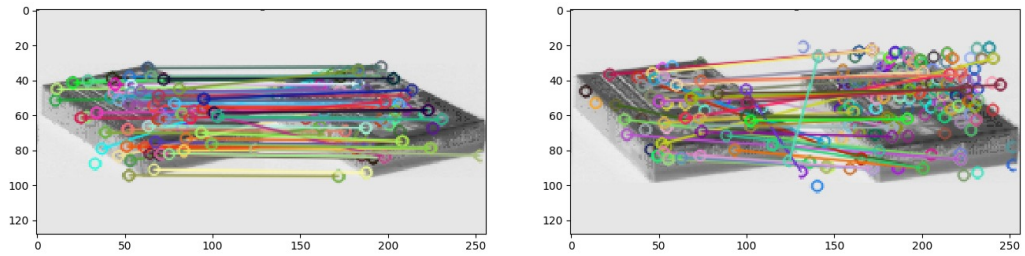


FIGURE 6 – Mise en correspondance des objets de même classe

Nous constatons que certains points d'intérêts des images de la figure 7 n'ont pas des correspondances, puisque l'objet se trouve sous un autre angle.

2.4.2 Mise en correspondance de deux images différentes

Nous remarquons que certains points d'intérêts ne correspond a aucun point dans l'image correspondant, il est tout 'a fait logique puisque l'objet se trouve sous des angles différents, en effet, il y a certains points ne correspond en aucun point (image au centre), certains correspond 'a d'autres points (image à droite),c'est dû à la considération des points jonctions.

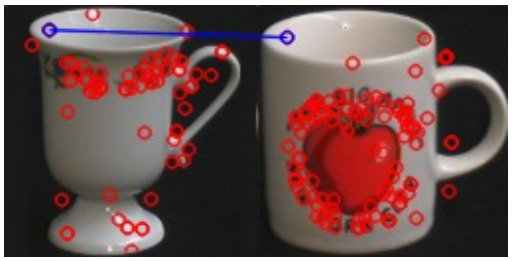


FIGURE 7 – mise en correspondance deux images différentes

Nous remarquons qu'il y a un seul point de correspondance entre ces deux images,tout 'a fait logique, puisque ce sont des images différentes et qui n'appartient mˆeme pas 'a la même catégorie. le seul point correspond est une coıncidence. Par ailleurs la première image n'a que deux images par ce que c'est une image homogène donc il n'y a presque pas des points d'intérêts.

2.4.3 les plus proches images

Implémentons une méthode qui nous permettra de sélectionner les images les plus proches voisins en terme de distance.

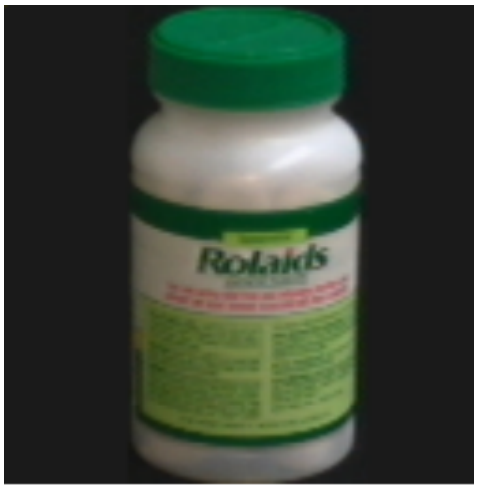


FIGURE 8 – Image de la requête (Obj5__200.png)

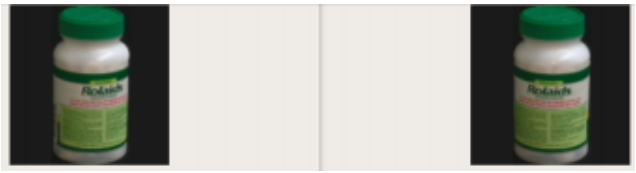


FIGURE 9 – Image de la requête (Obj5__200.png)

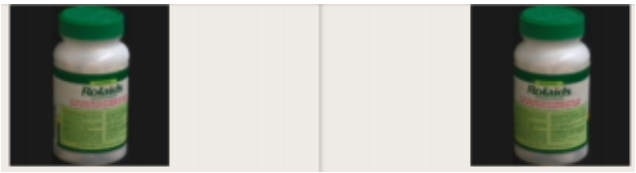


FIGURE 10 – Image de la requête (Obj5__200.png)

Voilà nous avons retrouvé les quatre (4) objets les plus proches à l'objet **xxx.png**

2.4.4 Détermination de fausses correspondance

Pour les fausses correspondances, nous avons utilisé la technique donné qui est une métrique possible est de calculer le ratio entre la distance du descripteur de l'image in-

connue et le descripteur de l'image modèle. Si ce ratio est inférieur a un seuil =0.75, alors la correspondance peut être considérée comme robuste, sinon elle est rejetée :

$$ratio = \frac{d_{descripteurImageInconnu}}{d_{descripteurImageModele}} < 0.75$$

2.4.5 Calcul de correspondance entre images

Maintenant, nous pouvons compter, non seulement le score mais aussi la correspondance réussies pour déterminer la classe l'image.

$$score = \frac{\#correspondancesréussies}{\#descripteursdelímagemodèle}$$

2.5 Contraintes de réalisation

Intuitivement, pour résoudre un problème il existe généralement plusieurs méthodes, techniques, façon de le faire. Cependant, pour la réalisation de ce TP, il nous a été assigné des contraintes de réalisation qui nous permettrons de suivre les exigences du sujet pour mener les travaux à bien.

Pour faire la reconnaissance d'objets il existe plusieurs méthodes, mais c'est avec la méthode du descripteur SIFT que nous sommes appelés à utiliser.

3 CONCLUSION

Après assimilation de ce TP, nous avons pu atteindre la quasi totalité de nos objectifs, dont on s'est assignés au départ qui consistait à faire la reconnaissance des objets à l'aide du descripteur SIFT. Retenons que nos résultats sont beaucoup plus encourageant visa à visa du cahier de charges du TP à travers les voyants sauf les matrices que confutions que nous avons pas pu visualisées. En effet, il est difficile de détecter les images disposant de très peu (nombre réduit) de descripteurs comme ce sont le cas des images lices. Cependant, comme toute oeuvre humaine, ce programme n'est pas parfait. En perspective, Nous comptons travailler beaucoup plus sur la performance de nos différents algorithmes. Ainsi, notre futur travail consistera à améliorer nos résultats et à tester d'autre jeux de données plus grands.

Références

[1] https ://docs.opencv.org/master/da/df5/tutorial_py_sift_intro.html

[2] http ://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php

[3] https ://www.cs.ubc.ca/ lowe/keypoints/