



Le 25 Février 2018
Yihong XU
Junshuai ZHU

Descripteurs locaux pour la recherche d'images similaires

TP Lot1

Ingénieur Généraliste
F4B 516
Encadrant : John Puentes



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

1. PROBLEMATIQUE

Le but de ce projet est de rechercher de manière automatique les objets similaires dans les images dont l'angle de visualisation et la condition lumineuse changent. L'entrée et une image ayant un objet principal et la sortie souhaitée est des images des objets les plus similaires. Les descripteurs utilisés sont ORB, Brief et SIFT.

2. INTRODUCTION DES ALGORITHMES

2.1 SIFT

Premièrement, l'axe de coordonnées doit être tourné vers la même orientation du point caractéristique (il s'agit de choisir une direction de référence pour chaque descripteur local) pour assurer l'invariance en rotation. L'orientation est calculée depuis l'histogramme d'orientation généré par la magnitude de gradient et l'orientation. Puis calculer le gradient des pixels adjacents dans une fenêtre de 16×16 autour du point caractéristique. Le gradient de ces pixels peut être pondéré par une fenêtre gauss et les pixels les plus proches ont un poids plus important. Une sous-région est composée de 4×4 pixels. Un histogramme d'orientation pour la sous-région peut être formé en accumulant l'orientation du pixel 4×4 qu'il contient. Il y a 4×4 sous-régions et chaque sous-région a 8 dimensions. Par conséquent, le SIFT donne un vecteur descripteur de **128** dimensions pour chaque point caractéristique.

+ : Rotation invariance ; Echelle invariance ; Luminance invariance ; Taux de correspondance haut.

- : Temps de calcul important.

2.2 Brief

Choisir la région pour calculer le descripteur ; Calculer la Convolution Gaussienne dans cette région pour éliminer le bruit ; Choisir de manière aléatoire des paires de point, ex. $\langle x_1, y_1 \rangle$, $\langle x_2, y_2 \rangle \dots$, autour du point caractéristique ; Calculer l'élément du descripteur en comparant la valeur de la paire de point choisie (ex, si $x < y$, retourne 1 ; Sinon, retourne 0) ; Générer le descripteur binaire en répétant plusieurs fois(ex.256) l'étape précédente.

+ : Rapide pour le calcul des descripteurs ; Rapide pour le pairage.

- : Pas d'invariance rotation ; Pas d'invariance en échelle ; sensible au bruit ; Taux de correspondance bas.

2.3 ORB (Oriented FAST and Rotated BRIEF)

rBRIEF utilise l'algorithme glouton pour filtrer les meilleures paires de points en choisissant la méthode : Construire un ensemble de 300k points caractéristiques. Pour chaque point caractéristique, considérez ses 31×31 points adjacents comme son patch. Pour la capacité anti-bruit, utilisez les 5×5 quartiers du point pour représenter la valeur de celui-ci. Donc, il y a $(31-5+1) * (31-5+1) = 729$ sous-fenêtres de taille 5×5 et il y a $M = (728+1) * 728/2 = 26536$ façons de choisir des paires de points. Dans ces 26535 façons de choisir, 256 meilleures méthodes de choix doivent être filtrées.

Pour chaque point clé, utilisez M façons de choisir des paires de points voisins et de générer une matrice Q égale à $300k * M$. Pour chaque colonne de Q, cela signifie que les descripteurs de ces 300k points clés sont générés en utilisant l'une des méthodes de choix de M.

Calculer la moyenne de chaque colonne dans Q ; Réorganiser le vecteur colonne de Q en utilisant la distance entre 0,5 et la moyenne de la colonne. Après que Q est réorganisé, il donne la matrice T. Mettez la première colonne de T dans R.

Calculez la corrélation entre la colonne suivante dans T et toutes les colonnes dans R. Si la corrélation est plus petite que le seuil correspondant, déplacez cette colonne vers R. Répétez l'étape précédant (le calcul de la corrélation), jusqu'à ce qu'il y ait 256 colonnes dans R. S'il n'y a pas assez de colonnes dans R, augmentez le seuil et réessayez.

- + : Rapide, Rotation invariance, invariance en échelle, moins sensible au bruit.
- : Taux de correspondance bas.

2.4 Résultat de l'extraction des points caractéristiques :

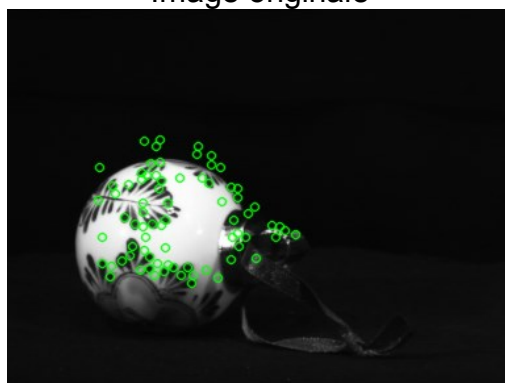
Dans l'exemple, le nombre de points détectés est varié. En plus, nous constatons aussi que les points détectés par Brief (*Star detector*) sont plus dispersés de l'objet par rapport aux autres détecteurs (DoG pour SIFT ; détecteur rapide pour ORB). SIFT est capable de détecter les points qui couvrent chaque partie de l'objet. Une remarque est que la taille de l'image a une influence importante au nombre de points caractéristiques détectés. Une taille de pixels assez grande est nécessaire pour la détection des *keypoints* dans l'objet. Dans ce projet, nous avons choisi une taille de 384x288.



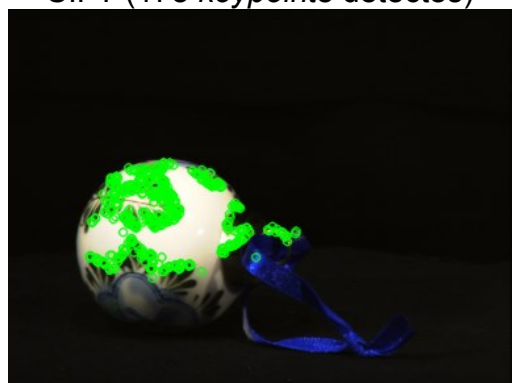
Image originale



SIFT (178 *keypoints* détectés)



Brief (75 *keypoints* détectés)



Résultat ORB (478 *keypoints* détectés)

Figure 1 Extraction des points caractéristiques.

3. INTRODUCTION DE LA BASE DE DONNEES

Nous utilisons une partie de la base d'images ALOI (Amsterdam Library of Object Images) et 2 groupes d'images ont été choisis :

3.1 Changement de température de couleur

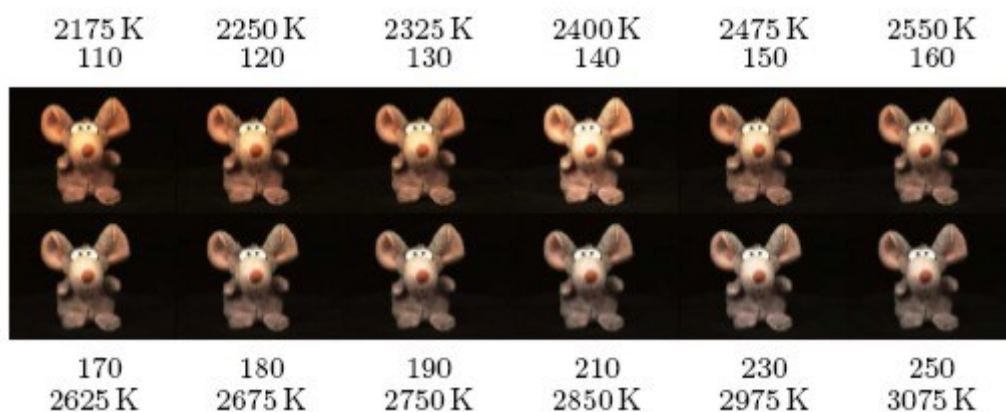


Figure 2 : Changement de température de couleur. (Source : ALOI)

3.2 Changement de l'angle de visualisation



Figure 3 : Rotation de l'image. (Source : ALOI)

A la vue des figures ci-dessus, pour chaque classe (type) d'objet, il y a, au total, 15 images. Nous avons choisi 100 classes pour ce projet, cela fait 1500 (15x100) images. Ces images ont été divisées en deux parties. 2/3 sont attribuées au groupe de l'entraînement (*clustering* par K-means) et servent comme la base de recherche. Le reste 1/3 sont la base de test. En résumé, pour chaque classe, 10 images (2 images de rotation + 8 images de changement de température) forment la base de recherche et 5 images sont pour le test (1 image de rotation + 4 images de température).

4. CHAINE DE TRAITEMENT

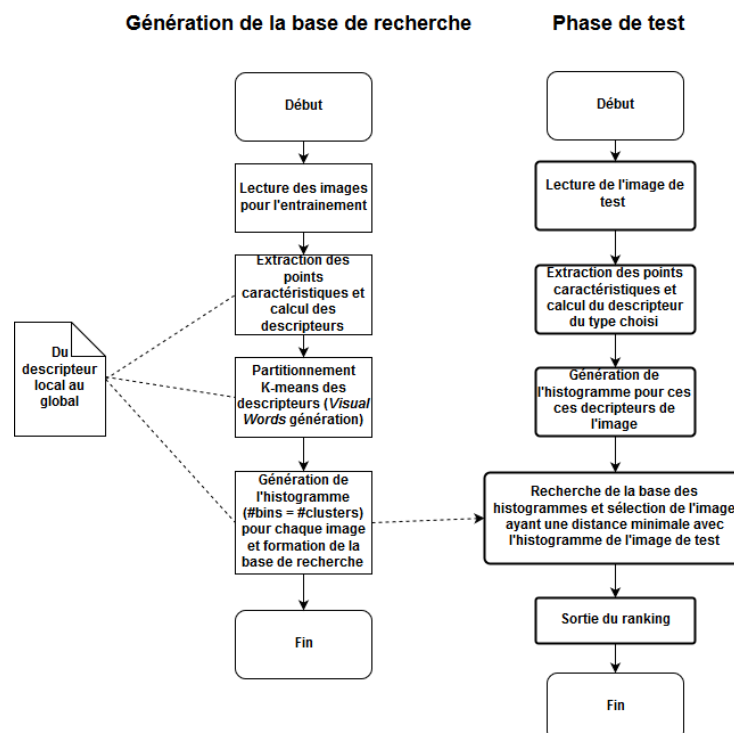


Figure 4 : Schéma de traitement.

5. PHASE DE TESTS ET COMPARAISONS

Premièrement, nous faisons la comparaison du graphe P-R qui est un métrique faisant la courbe ($Recall = \frac{TP}{TP+FN}$, $Precision = \frac{TP}{TP+FP}$) pour les classeurs binaires. Pour chaque classe, toutes les images (5 exemples positifs) de test de cette classe sont sélectionnées et à cause de temps de calcul, seulement 10 autres images venant d'autres classes sont choisies, de manière aléatoire (exemples négatifs). A partir de ces 15 images, un graphe P-R est construit pour chaque classe et nous évaluons l'AUC (Area Under Curve) de chaque graphe.

Le graphe P-R s'adapte au classifieur binaire, ce qui n'est le cas dans notre problème. En plus, P-R n'évalue que sur la probabilité ou le score du niveau de ressemblance entre l'image en entrée et la classe évaluée.

Autrement dit, le niveau de ressemblance avec d'autres classes ne sont pas évalués. Il n'est pas donc approprié naturellement dans notre cas (multi-classe et sortie étant la distance). Pour cela, nous avons adopté une approche *one v.s all* et le score se fait par l'équation ci-dessous :

$$score = \sum_i \frac{1}{distance \text{ à l'image } i \text{ de cette classe dans la base} + 0.001}$$

De cette manière, l'image ayant le plus grand score a une distance minimale à cette classe.

5.1 Performance des différents descripteurs

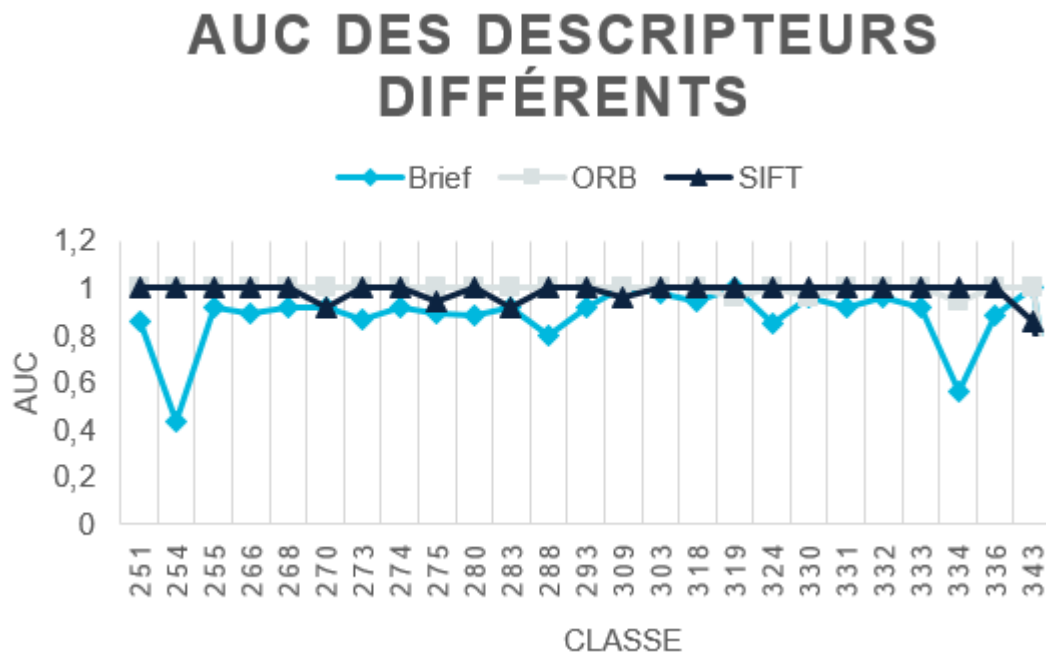


Figure 5 : AUC des descripteurs differents.

Nous ne montrons que 27 classes ayant un AUC moins de 1 dans certains types de descripteurs. Nous fixons le nombre de clusters à 50 et le nombre maximum de points caractéristiques détectable pour une image est fixé à 100.

Parmi ces descripteurs, l'ORB a une performance meilleure avec la valeur moyenne égale à 0.9944 ; SIFT a aussi une meilleure performance avec la valeur moyenne .égale à 0.984 ; BRIEF a une valeur moyenne égale à 0.8844.

	moyenne	Variance
Brief	0.8844	0.125669
ORB	0.9944	0.15513
SIFT	0.9840	0.035327

Tableau 1 : Moyenne et variance de l'AUC.

5.2 Performance pour quantité des points caractéristiques différente

Pour voir l'influence du nombre de points caractéristiques sur la performance de la recherche des images similaire. Comme l'exemple, nous prenons le descripteur ORB et fixons le nombre de clusters du K-means à 50.

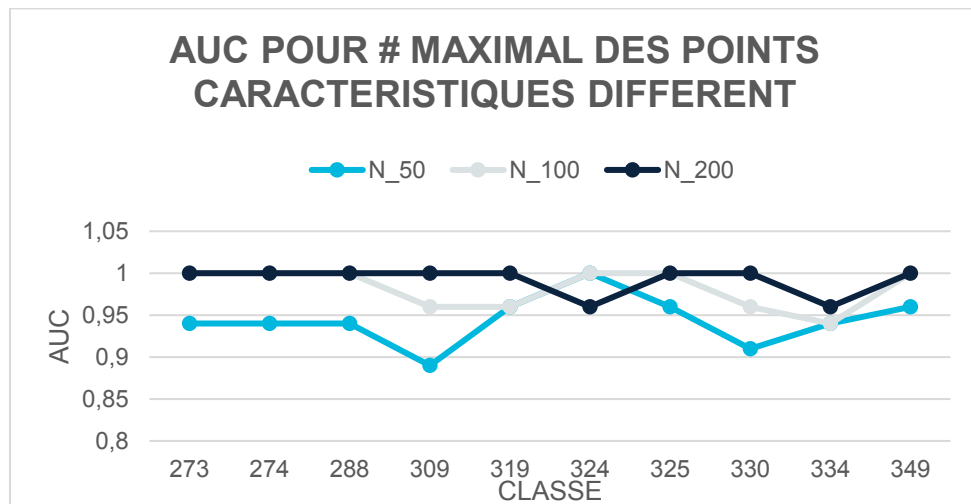


Figure 6 : AUC pour nombre des points caractéristiques différent.

Nous trouvons que la performance de la recherche pour des images similaires se dégrade en limitant le nombre maximum autorisé des points caractéristiques détectés. Cette dégradation de performance est raisonnable parce que le nombre très limité ne peut pas caractériser l'image en entière. Par contre, N=100 et N= 200 n'ont pas de différence importante.

	Moyenne	Variance
#N = 50	0.9440	0.0284
#N = 100	0.9820	0.0227
#N = 200	0.9920	0.0160

Tableau 2 : Moyenne et variance de l'AUC.

5.3 Performance pour quantité du cluster différente

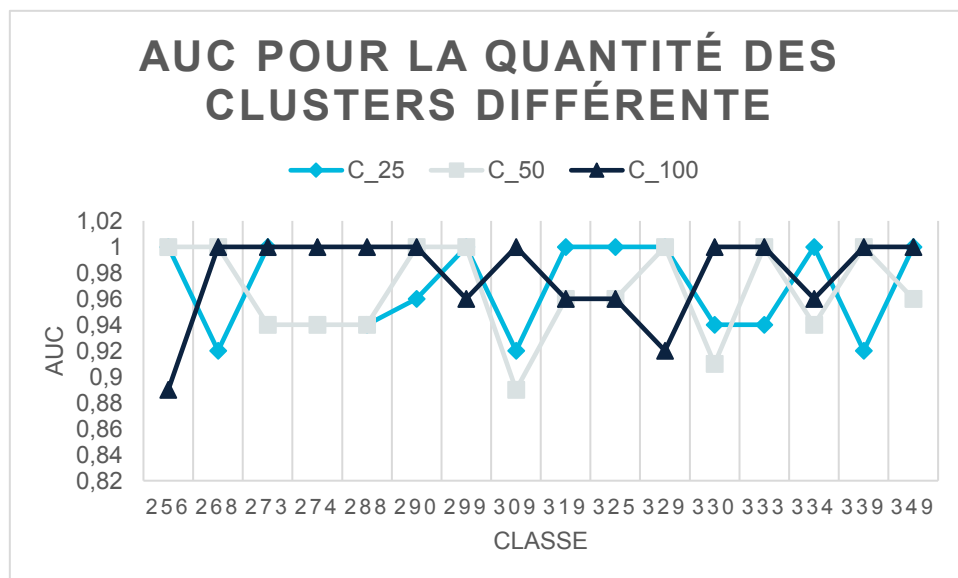


Figure 7 : AUC pour la quantité du cluster différente.

Dans le cas où la quantité du cluster est différente (nous testons sur l'ORB et nombre maximum de points caractéristiques est fixé à 50), nous pouvons trouver que la quantité du cluster n'influence pas de manière conséquente sur la performance. La valeur moyenne de l'AUC pour la quantité du cluster égale à 25, 50 et 100 sont respectivement, 0.969, 0.965, et 0.978.

	Moyenne	Variance
#C = 25	0.9693	0.0342
#C = 50	0.9650	0.0352
#C = 100	0.9781	0.0328

Tableau 3 : Moyenne et variance de l'AUC.

En revanche, il faut noter qu'un bon résultat en termes de valeur de l'AUC pour une classe ne veut pas dire que les images de même classe seront forcément sélectionnées lors du test. C'est parce qu'il existe parfois que même une image de classe i en entrée ayant une distance minimale à toutes images de classe i par rapport aux exemples négatifs, il est probable qu'elle une distance plus petite à d'autres images d'une classe différente dans la base.

Pour cette raison, nous ensuite montrons ci-dessous le TOP10 Ranking en termes de distance avec les images dans la base, ceci reflète quelles images vont être sélectionnées lors que la recherche. Deux images de classe 251 ont été choisies (251_c.png et 251_i120.png) et #cluster = 50, #max keypoints = 100.

Ranking \ type	ORB		SIFT		Brief	
	251_c (rotation)	251_i120 (color temperature)	251_c (rotation)	251_i120 (color temperature)	251_c (rotation)	251_i120 (color temperature)
1	251_i190	251_i130	251_i210	251_i130	260_l	251_i210
2	251_i180	251_i110	251_i190	251_i110	304_i180	319_i210
3	251_i110	251_i180	251_i230	251_i230	304_i170	251_i250
4	251_i230	251_i140	251_i180	251_i180	309_c	319_i190
5	251_i250	251_i210	336_i170	251_i190	285_i110	319_c
6	251_i210	251_i250	251_i250	251_i140	285_i170	251_i180
7	251_i130	251_i230	278_c	265_i120	285_i180	319_i120
8	251_i140	251_i190	336_i110	313_i250	340_i130	251_i190
9	346_i110	260_l	336_i120	277_c	300_l	251_i110
10	346_i230	251_l	336_i190	300_l	285_i190	251_i130

Tableau 4 : Ranking de la recherche.

D'une part, l'objet tourné est difficile à trouver ses images similaires parce que la forme est déformée et une partie de l'objet est cachée. En revanche, le simple changement de la luminosité n'est pas gênant pour les détecteurs. D'autre part, nous pouvons voir que l'ORB a un meilleur résultat pour cette image (voir ci-dessous) et souvent l'ORB est capable de trouver plus de points caractéristiques que d'autres méthodes. La performance de Brief s'améliore quand il existe plus de patterns dans l'objet (le nombre de points détectés augmente).

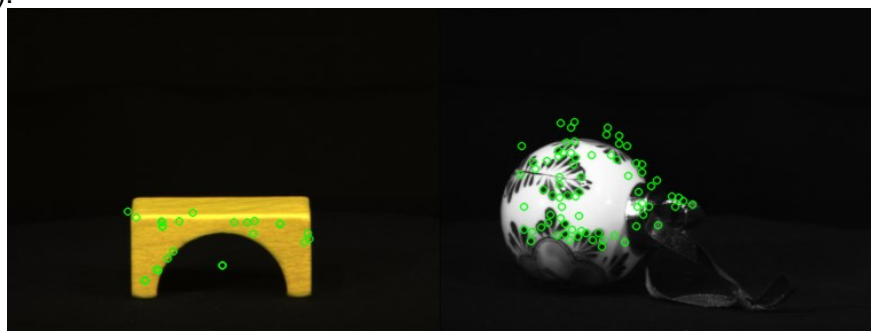


Figure 8 : 251_c (25 keypoints) et 353_i120 (75 keypoints).

6. CONCLUSIONS

- L'ORB a une performance meilleure que d'autres méthodes dans le cas général parce qu'il est capable de détecter plus de keypoints.
- La rotation de l'objet déforme et cache la forme de l'objet. Il est donc difficile à trouver ses images similaires.
- L'ORB est la méthode la plus rapide en termes de temps de calcul parmi les trois.
- Le nombre de points caractéristiques a une influence importante sur la performance, nous constatons qu'il a besoin au moins 100 pour caractériser l'objet dans une image.
- La performance n'est pas très sensible au nombre de clusters (c'est-à-dire, nombre de *visual words* pour calculer les histogrammes). Un nombre de 25, 50, 100 n'a pas de différence conséquente.
- les points caractéristiques se concentrent sur les textures et la forme de l'objet. Un objet ayant une couleur uniforme est plus difficile à extraire des points caractéristiques.