

Projet M1 Informatique

Représentation de Données Visuelles

Sujet : Utilisation d'un descripteur de Fourier pour la reconnaissance de formes

January 23, 2019

Ce sujet vise à implémenter une version simplifiée de l'article présenté dans [1]. L'objectif est de retrouver parmi un ensemble d'images noir et blanc, représentant des classes d'objets (*watch*, *turtle*, *truck*, etc.), les images montrant le même objet que celui représenté dans une image donnée (appelée *image requête*).

Plus exactement, on souhaite pouvoir calculer un score de distance entre l'image requête et toutes les images de la base de référence, tel que si l'on trie ces scores par ordre croissant, les images représentant la classe d'objet de l'image requête apparaissent en premier dans la liste triée. L'algorithme idéal obtiendrait ainsi les K images montrant l'objet recherché aux K premières places de la liste triée. Bien entendu, cet algorithme n'existe pas, mais on espère tout du moins que ce soit le cas pour des objets facile à reconnaître, et que les résultats soient le moins mauvais possible pour des objets difficiles à reconnaître. Par exemple en figure 1, un cheval a été pris pour un cerf, mais parmi les cinq meilleurs scores, on trouve tout de même quatre images montrant un cerf, y compris un cerf plus petit que dans l'image requête et ayant subi une rotation dans l'image.

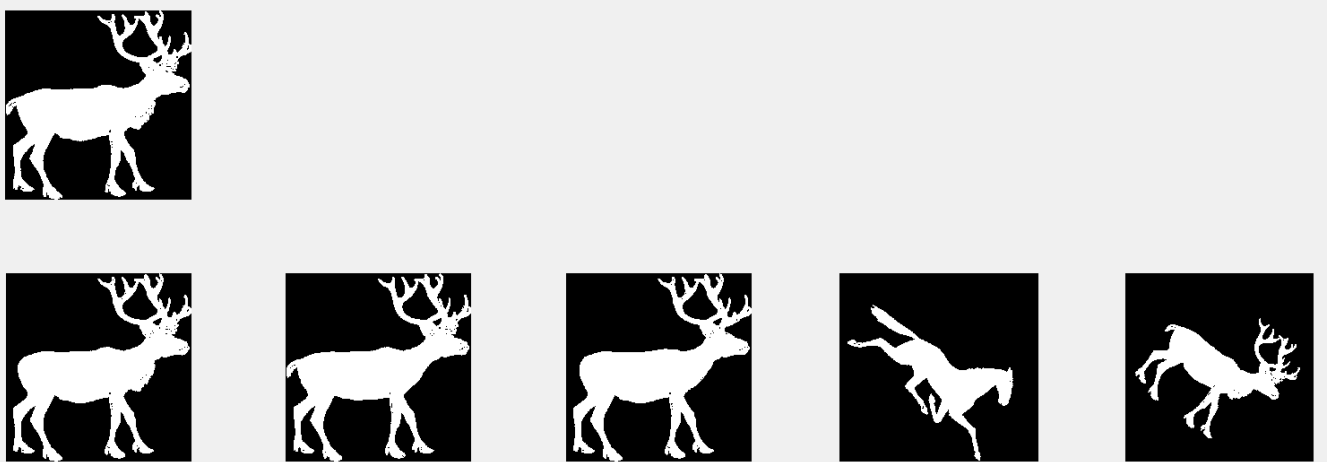


Figure 1: Exemple d'image requête (en haut) et des 5 premiers résultats (en bas) triés selon la distance euclidienne entre les descripteurs de Fourier.

L'algorithme que nous proposons est le suivant (nous utilisons une Transformée de Fourier – TF mono-dimensionnelle, tandis que les auteurs de [1] utilisent une TF bi-dimensionnelle) :

1. Calculer le barycentre m des pixels blancs.
2. Pour N valeurs d'un angle t variant de 0 à 2π (le choix de N est laissé à votre appréciation), calculer l'intersection $p(t)$ entre le contour de l'objet et le rayon partant de m formant un angle t avec l'axe horizontal de l'image. Soit $r(t)$ la distance euclidienne entre m et $p(t)$. Nous appellerons *profil* de la forme la courbe $r(t)$ (voir la figure 2).
3. Calculer la TF $R(f)$ de $r(t)$. Le descripteur de Fourier que nous utiliserons pour calculer les scores est le vecteur fd formé par les M premiers coefficients du vecteur $|R(f)|/|R(0)|$. Le choix de M est également laissé à votre appréciation.

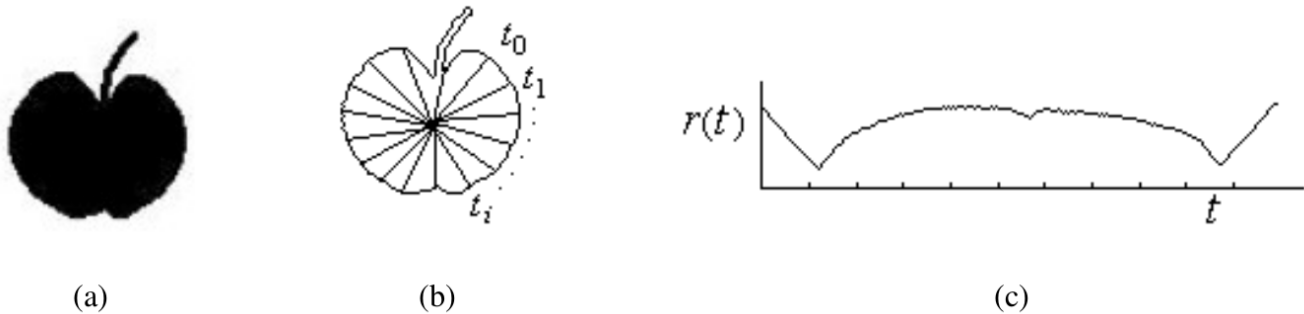


Figure 2: Calcul du profil de forme. (a) Image originale. (b) Rayons partant du barycentre des pixels noirs et allant jusqu'au contour de la forme. (c) Profil de forme obtenu.

Une méthode simple pour calculer les points d'intersection $p(t)$ sans avoir à détecter le contour de l'objet au préalable est la suivante :

- partir du barycentre m ,
- avancer le long du rayon d'angle t tant que le pixel est blanc.

Le dernier pixel blanc rencontré le long du rayon correspond à $p(t)$ (n'hésitez pas à proposer une autre méthode si celle-ci vous semble trop basique et imprécise).

Le score utilisé pour trier les images est alors simplement la distance euclidienne entre le descripteur de Fourier fd de l'image requête et celui de l'image de la base. Vous pouvez bien-sûr visualiser les réponses obtenues par votre algorithme, par exemple sous la forme de la figure 1, mais il est surtout important de respecter le profil et le nom de la fonction `compute_fd` indiqué ci-dessous, car c'est cette fonction que nous utiliserons pour évaluer votre travail :

```
function [fd,r,m,poly] = compute_fd(img)
```

avec :

- `img` : image en noir et blanc (obtenue à l'aide de la fonction `logical()`) dont la fonction calcule le descripteur de Fourier,
- `fd` : descripteur de Fourier (vecteur ligne de taille M),
- `r` : profil de la forme (vecteur ligne de taille N),

- `m` : barycentre des pixels blancs (vecteur ligne de taille 2),
- `poly` : polygone 2D décrivant le contour de l'objet (matrice de taille N lignes, 2 colonnes).

Le code permettant de charger les images de la base dans une liste, charger les images de requête, appeler la fonction `compute_fd()` et afficher les meilleurs résultats vous est donné. Le code qui vous est remis est exécutable, mais le contour de l'objet est arbitraire (un cercle) et le descripteur aléatoire. A vous de modifier la fonction `compute_fd()` pour produire de meilleurs résultats.

Référence

- [1] Dengsheng Zhang and Guojun Lu. Shape-based image retrieval using generic Fourier descriptor. In *Signal Processing: Image Communication, Volume 17, Issue 10, 2002*.