



Cours de segmentation d'images

TP2 – Morphologie avec OpenCV

Master M2TI - Paris V

John Chaussard

LAGA – Université Paris 13

chaussard@math.univ-paris13.fr

Eléments structurants

Utiliser un module personnel

Un module est en fait un fichier contenant une ou plusieurs fonctions utiles. Récupérez, parmi les fichiers du TP, le fichier `strel.py` contenant une fonction pour générer des éléments structurants.

Pour l'utiliser dans votre code, placez le fichier dans le même répertoire que votre fichier code, où vous écrirez en tête :

```
import strel
```

Dans la fonction principale, créez un élément structurant de type diamant (Γ_4), de taille 2, en faisant

```
test = strel.build('diamant', 2, None)
print(test)
```

Comme vous pouvez le constater, cette fonction renvoie une liste de coordonnées des pixels appartenant à l'élément structurant.

Création d'élément structurant

Cette fonction permet de construire quatre types d'éléments structurants :

- Pour construire un **carré** (Γ_8), passez le mot clef **carre** en paramètre, la taille en seconde paramètre, et None en troisième. **Attention : le carré sera en réalité de taille $2 * taille + 1$**
- Pour construire un **diamant** (Γ_4), passez le mot clef **diamant** en paramètre, puis la taille en second paramètre, puis None en troisième.
- Pour construire un **disque**, passez le mot clef **disque**, puis le rayon, puis None.
- Pour construire une **droite**, passez le mot clef **ligne**, puis la demie longueur de la droite, puis l'angle avec l'axe des abscisses.

Visualiser un élément structurant

L'élément structurant généré est une liste... Il est possible de le convertir en image afin de visualiser le résultat :

```
test = strel.build('diamant', 150, None)
Imtest = strel.toImage(test)
afficher_image(Imtest)
```

A vous de jouer : créez un cercle de rayon 90, puis une ligne de taille 300 et angle 30 degrés. Visualisez le résultat afin de vous assurer que l'objet est créé convenablement.

La morphologie avec OpenCV

Morphologie avec OpenCV

OpenCV peut faire les opérations de base de morphologie très rapidement. Par exemple, pour faire une dilatation, par un élément structurant, on commence par générer l'élément structurant :

```
elem = strel.build('cercle', 100, None)
```

On convertit ensuite cet élément structurant en une image (car c'est une liste) :

```
elemi = strel.toImage(elem)
```

Puis on appelle la fonction openCV de dilatation :

```
dil= cv2.dilate(image_entree, elemi, iterations=1)
```

Comparez la vitesse d'exécution de votre dilatation (avec un grand élément structurant) et de celle d'OpenCV.