

TP - Opérations morphologiques et classification

Traitement d'images - M1 STIC -

Vous devrez rendre un compte rendu détaillé à la fin de chaque séance à l'adresse : cyrille.migniot@u-bourgogne.fr Pensez à bien analyser et à commenter intelligemment chacun de vos résultats.

1 TP1 : opérations morphologiques

Objectifs : Manipuler des opérateurs morphologiques élémentaires / Analyser les effets de ces opérations sur des images binaires. / Vérifier certaines propriétés.
Supports : MATLAB

1. Coder les fonctions qui permettent de réaliser l'érosion, la dilatation, l'ouverture et la fermeture sur une image binaire. Sur un exemple vérifier les propriétés ((anti-)extensive / croissante / idempotente) pour l'une de ces fonctions.
2. Réaliser la fonction qui détermine le squelette d'un objet (sur image binaire) à partir soit de la méthode de Lantuejoul soit de l'amincissement homotopique.

2 TP2 : classification

Objectifs : Manipuler plusieurs algorithmes de classification. / Comparer les deux modes de classification. / Utiliser un descripteur de texture. / Analyser des images de texture.
Supports : MATLAB

2.1 Algorithme

Télécharger le dossier `classification.zip` sur Plubel.

Ce dossier contient des fichiers matlab qui réaliseront le traitement demandé. Le dossier `imgTexture` contient un ensemble d'images possédant des textures bien descriptives et le dossier `imgCouleur` des images étant mieux définie par la couleur.

Pour chacune des images, on calcule la matrice des cooccurrences spatiales et à partir de cette matrice on obtient l'énergie, le contraste, la corrélation (lié à la variance) et l'homogénéité (lié au moment inverse).

2.2 Manipulation

1. La fonction `descripteurTexture` calcule pour une image un descripteur (vecteur) qui contient plusieurs paramètres de texture. Le fichier `calculDescripteur` permet d'afficher le descripteur pour chaque image dans le dossier `img`. Afficher les 4 paramètres de la matrice des cooccurrences spatiales pour l'ensemble des images de texture. Analyser les résultats. (Par exemple à quelles images correspondent les plus hautes et les plus basses valeurs.) Essayer avec plusieurs orientations et plusieurs distances. Quelles sont les paramètres les plus discriminants.

2. La fonction `classificationNueesDynamiques` permet de classer les images contenues dans le dossier `img` par la méthode des nuées dynamiques. Le nombre de classe est à désigner par l'utilisateur.
Réaliser un descripteur qui contienne les quatre paramètres de textures. Utiliser la méthode des nuées dynamiques pour séparer l'ensemble en 2, 3, 4 puis 5 ensembles. Commenter la répartition.
3. Répartir les images dans les trois dossiers : `classe1`, `classe2` et `tests`. Les deux premiers ensembles forment les deux classes d'apprentissage. Le dernier contient les images qui seront testées pour évaluer la classification. La fonction `classificationSVM` permet d'utiliser cet apprentissage pour classer par la méthode des SVM les images du dossier `tests`. Pour une représentation 2D utiliser la fonction `classificationSVM2D`. Le descripteur ne devra alors prendre en compte que 2 paramètres.
Réaliser un descripteur qui contienne le contraste et l'énergie. Essayer la répartition suivante et analyser les résultats.
 - classe 1 : brique, laine, motif3 et motif4.
 - classe 2 : lignes, points, sable et tissu.Utiliser une fonction kernel linéaire puis polynomiale. Essayer d'autres répartition.
4. La fonction `descripteurCouleur` réalise un descripteur prenant en compte la couleur. En fait il isole simplement les n couleurs principales de l'image (n à choisir par l'utilisateur). Refaire les classifications précédentes avec ce descripteur et les images du dossier `imgCouleur`. Tester avec de nouvelles images.