

TP LOVMI

Objectifs

Ajouter un filtre de traitement d'image au logiciel lovmi.

Pre-Requis

Notion de traitement d'image. Bonne connaissance de la programmation objet avec Java.
Connaissance de l'IDE eclipse.

Déroulement du TP

Nous allons prendre comme exemple le filtre logarithme pour illustrer l'ajout d'un filtre, et détailler chaque étapes dans l'ordre.

Description du filtre

Le filtre logarithme est utilisé pour les images radar. Dans le cas des images radar, le bruit est dit multiplicatif. On applique donc le logarithme sur l'image, afin de rendre le bruit additif.

Préliminaires

Pour vous aider, une documentation est fourni, décrivant l'interface Filter et le rôle de chaque méthode de cette interface. Pensez donc à vous poser les bonnes questions.

Sur quelle type d'images peut on appliquer le filtre?

Que doit on faire avant que le filtre soit lancé sur toutes l'image?

Est-ce que le filtre produit toujours un résultat?

Préliminaires

On va appliquer la formule suivante à chaque pixel séparément. D'abord calculer pour chaque pixel le logarithme en base 10. L'ensemble des valeurs est restreint sur un petit intervalle. Si on garde les pixels à ces valeurs, on obtiendra une image presque entièrement noir. Il faut donc strecher les valeurs sur l'intervalle de valeurs du type de l'image.

Pour chaque pixel, on va donc appliquer la formule suivante:

$\text{max} = \text{Valeur maximum de pixel du type de l'image}$

$\text{Rresultat}(i, j) = \log(\text{Rorigine}(i, j)) * \text{max} / \log(\text{max})$

Implémentation du filtre

En premier nous devons déclarer une classe implémentant l'interface Filter. Cette classe doit être mise dans

le package `fr.uml.v.lovmi.filters`.

Le constructeur de la classe doit être vide.

Si nécessaire, la méthode `init` servira à initialiser les paramètres du filtre. Dans notre cas cela ne sera pas nécessaire car le logarithme ne nécessite pas de paramètres. La méthode `init` devra tout simplement retourner `true`.