

Travaux Pratiques - TERI

Initiation

1 Préliminaires: utilisation d'octave/matlab

1.1 Avant de commencer le TP

Pour se connecter :

- **login:** XXXXXXXX (le numéro i à mettre dépend de la position de votre poste informatique dans la salle de TP)
- **password:** XXXXXXXX (attention, respectez les lettres capitales)
- pour démarrer l'environnement Unix tapez `startx10`;
- puis `exec zsh` dans chaque fenêtre de commandes

Pour démarrer le TP, il faut executer les commandes

```
source /tsi/zone/tp/TP/Init2006/debut_tp_octave.sh
emacs programmation/scratch.m &
/usr/pkg/bin/octave
sous octave tapez : init_octave
```

qui vont vous créer un répertoire TPANIMINIT dans lequel se trouvera un lien symbolique vers le répertoire images. Vous serez automatiquement mis dans ce répertoire. Les images se trouvent dans le (lien symbolique vers le) répertoire images (dans lequel vous ne pouvez pas créer ou sauvegarder des images, créez-les dans votre répertoire de travail).

Vous trouverez sous le répertoire programmation un certain nombre de programmes déjà écrits. Si vous devez écrire de nouvelles fonctions faites-le dans ce répertoire.

Vous avez à votre disposition:

Un terminal dans lequel le programme octave est lancé. Tapez `init_octave`.

Un éditeur de texte (emacs) dans lequel vous êtes invités à taper les commandes importantes avant de faire un copier-coller (copy-paste) dans octave. Ainsi, vous garderez une trace de votre travail. Vous pouvez créer d'autres fonctions ou fichiers de commandes octave depuis cet éditeur. Il faut s'assurer que vos fichiers de commande portent l'extension `.m` et se trouvent dans le répertoire programmation.

Si dans une autre fenêtre de terminal, vous souhaitez avoir accès à l'environnement adéquat, aller (commande `cd` du shell) dans le répertoire TPANIMINIT et taper

```
source /tsi/zone/tp/TP/Init2006/debut_tp_octave.sh
/usr/pkg/bin/octave
init_octave
```

1.2 L'environnement octave

octave est un logiciel libre dont les commandes ressemblent beaucoup à celles du logiciel matlab. Cependant, la manipulation des images dans octave diffère de celle de matlab. Le fichier `scratch.m` (qui est ouvert dans votre éditeur emacs) contient un petit tutoriel pour maîtriser octave et la manipulation d'image. Lisez-le et exécutez les commandes afin de vous familiariser avec octave. N'hésitez surtout pas à demander de l'aide si vous ne comprenez pas une commande ou si vous rencontrez un problème avec octave. Si vous avez besoin d'informations sur une commande, utilisez `help commande`.

2 Visualisation et utilisation de xv

Le but de cette partie est de se familiariser avec le logiciel de visualisation xv.

xv est un logiciel simple de visualisation et de traitement d'images. Chaque instance du logiciel ne visualise qu'une image à la fois. Pour zoomer sur une partie de l'image il suffit d'appuyer sur Control et cliquer-gauche. Pour dézoomer on utilise le clic droit (et Control). Le bouton du milieu permet d'avoir un affichage de la position du curseur dans l'image ainsi que la couleur du pixel correspondant. Le bouton droit permet d'ouvrir une boîte de dialogue (sauvegarde, changement de taille de l'image...).

2.1 Zooms

Ouvrez une image de votre choix dans xv (`viewimage`). Zoomez-la d'un facteur 2 (par l'intermédiaire de la boîte de dialogue) puis sauvegardez-la (en format tif et faites attention au répertoire où elle a été sauvee) avant de l'ouvrir à nouveau avec xv. Comment xv a-t-il procédé pour effectuer le zoom? Même question avec un zoom de 10%.

Appliquez la même démarche à l'image **lena.tif** mais cette fois-ci en dézoomant d'un facteur 2.

Ouvrez l'image **lena_petit.tif** (dans le répertoire images) et zoomez-la d'un facteur 2. Comparez le résultat par rapport au zoom de la petite image que vous avez produite avec xv. Quelle hypothèse pouvez-vous faire sur la génération de **lena_petit.tif**? (elle a été produite à l'aide du logiciel gimp)

3 Niveaux de gris, histogrammes

Le but de cette partie est de manipuler les niveaux de gris des images et d'en explorer (de manière simplifiée) les statistiques, la principale étant l'histogramme de l'image.

3.1 Changement de contraste

Modifiez les niveaux de gris d'une image de votre choix en utilisant la boîte "Intensity". L'aspect global de l'image s'en trouve-t-il modifié? Que se passe-t-il si l'on applique une transformation non-croissante des niveaux de gris?

3.2 Histogramme

Visualisez une image de votre choix (la commande `ls images` liste le contenu du répertoire images) et affichez son histogramme (pour obtenir l'histogramme d'une image on utilise la fonction `hist(im(:),N)`, N étant le nombre de cases, voir l'exemple donné dans **scratch.m**). Donnez l'interprétation de l'histogramme pour l'image choisie.

Nous étudions maintenant l'effet de dégradations simples des images sur leur histogramme. À l'aide de la fonction `noise`, ajoutez un bruit gaussien à une image et étudiez l'effet produit sur son histogramme (pour afficher un autre histogramme dans une autre figure, utilisez la commande `figure(2)`).

En considérant les niveaux de gris d'une image comme la réalisation d'une variable aléatoire dont la loi est l'histogramme de l'image, interprétez le résultat. Vers quoi tend l'histogramme lorsque le bruit devient très important ?

3.3 Quantification

Le dithering (ou détramage) est une technique utilisée à l'origine pour obtenir un rendu en niveaux de gris lorsque l'appareil d'impression ne disposait que de deux couleurs (noir et blanc). À l'aide de la fonction `quantize`, observez l'effet d'une quantification de plus en plus brutale sur une image. Visualisez l'histogramme.

4 Spectre des images et transformation de Fourier

4.1 Visualisation de spectres

Grâce à la fonction `view_spectre` on peut visualiser le spectre d'une image. Visualisez les spectres de différentes images en utilisant les options 1 et 2 (`help view_spectre`).

Que constatez-vous? Qu'en déduisez-vous par rapport au spectre d'une image? (Pour la suite on visualisera toujours avec l'option 2)

Comment influe l'option hamming sur le spectre de l'image? (multiplication de l'image originale par une fonction très lisse qui s'annule aux bords de l'image)

Visualisez le spectre de l'image de bâtiments **montreuil.tif**. Expliquez les différentes informations qui sont visibles dans le spectre de l'image.

Visualisez le spectre de l'image synthétique **rayures.tif**. Que constatez-vous? Peut-on retrouver les caractéristiques des rayures de l'image à partir de son spectre? Essayer de comprendre le spectre en décomposant l'image et en visualisant les spectres des sous-parties. Expliquez la différence entre la visualisation avec et sans l'option hamming?

Quel effet a le sous-échantillonnage sur le spectre (on peut utiliser une image synthétique et une image naturelle (Par exemple **carte_nb.tif**))? Expliquer la nouvelle apparence des rayures sur **carte_nb.tif**.

4.2 Effet de Gibbs

A l'aide de la fonction `filterlow`, appliquez un filtre passe bas parfait à une image. Visualisez l'image résultante, ainsi que son spectre. Que constatez-vous?

Mêmes questions en utilisant la commande `filtergauss`.

Visualisez les deux masques (sous le répertoire images) **masque_bas_centre.tif** (pour le filtrage passe-bas parfait) et **masque_gauss_centre.tif** (pour le filtrage gaussien) et évaluez leur moyenne par `mean`. Si l'on admet que le but du filtrage est d'éliminer le bruit blanc, est-ce que les deux filtres sont équivalents?

5 Gradient et statistiques

5.1 Différences de niveaux de gris voisins

On veut maintenant étudier la statistique de la différence de niveau de gris entre deux pixels adjacents. Grâce à la commande `gradx` on peut obtenir une image de la différence entre pixels adjacents. Visualisez l'histogramme d'une telle image. Visualisez le logarithme de l'histogramme.

La distribution des différences vous semble-t-elle obéir à une loi gaussienne? Pourquoi? Quelle aurait été la forme de l'histogramme si l'on avait considéré la différences entre pixels plus éloignés? (On ne demande pas de faire l'expérience).