

TP n°3

Filtre inverse vidéo et floutages d'images

L'objectif de ce TP est d'appliquer différents filtres sur des images comme un filtre inverse vidéo ou un floutage d'images à différents niveaux de résolution.

1) Création d'une image couleur (ppm) et d'une image en niveau de gris (pgm), histogramme de l'image pgm et profil de ligne

- Télécharger une image issue d'un APN (c-a-d pas une image de synthèse). Faire valider l'image choisie par l'enseignant. Transformer cette image au format ppm, puis pgm de petite taille (256x256 pixels par exemple).
- Avec l'aide de GNUPLOT, afficher l'histogramme de l'image en niveau de gris.
- Avec l'aide de GNUPLOT, tracer le profil d'une ligne de l'image en niveau de gris.

Dans votre compte rendu :

- Insérer l'image originale en précisant l'adresse web de référence.
- Insérer l'image au format pgm de taille réduite.
- Insérer l'histogramme.
- Insérer le profil de ligne en indiquant le numéro de la ligne choisie.

2) Inverse vidéo

Créer un nouveau programme `inverse.cpp` qui va inverser les niveaux de gris d'une image. En fait le noir (0) doit devenir blanc (255), et le blanc doit devenir noir, ainsi que tous les niveaux de gris qui vont être inversés.

- Ecrire le programme `inverse.cpp`, le compiler et l'exécuter en utilisant l'image choisie en début de TP. Pour l'exécution, 3 arguments sont nécessaires, à savoir, le nom du programme, le nom de l'image d'entrée et le nom de l'image de sortie.
- Visualiser les profils d'une ligne des 2 images (entrée et sortie). Comparer les 2 profils. Que constatez-vous ?

Dans votre compte rendu :

- Insérer l'image d'entrée et l'image de sortie inversée.
- Insérer les profils d'une ligne de l'image d'entrée et de l'image de sortie en indiquant le numéro de la ligne choisie.
- Indiquer ce que vous constatez.

3) Filtre flou1

- Créer un nouveau programme `filtre_flou1.cpp` qui va remplacer, dans l'image de sortie, la valeur d'un pixel par la valeur moyenne de ce pixel avec ses 4 voisins :

$$p_out(i, j) = (p(i, j) + p(i-1, j) + p(i+1, j) + p(i, j-1) + p(i, j+1)) / 5.$$

Attention aux bords de l'image ! En effet, la première et la dernière colonnes, ainsi que la première et la dernière lignes ne peuvent pas être traitées, il faudra conserver les valeurs initiales.

- a) Visualiser les profils d'une ligne des 2 images (entrée et sortie). Comparer les 2 profils. Que constatez-vous ?

Dans votre compte rendu :

- Insérer l'image d'entrée et l'image de sortie floutée.
- Insérer les profils d'une ligne de l'image d'entrée et de l'image de sortie en indiquant le numéro de la ligne choisie.
- Indiquer ce que vous constatez.

4) Filtre flou2

- a) Créer un nouveau programme `filtre_flou2.cpp` qui va remplacer, dans l'image de sortie, la valeur d'un pixel par la valeur moyenne de ce pixel avec ses 8 voisins.
- b) Utiliser l'image de sortie comme image d'entrée pour flouter une seconde fois l'image. Appliquer jusqu'à 5 fois le floutage.
- c) Visualiser les profils d'une ligne des 6 images (entrée, sortie1, sortie2, ..., sortie5). Comparer les 6 profils. Que constatez-vous ?
- d) Visualiser les histogrammes de l'image d'entrée, l'image floutée 1 fois, l'image floutée 2 fois et l'image finale floutée 5 fois.

Dans votre compte rendu :

- Insérer l'image d'entrée et les images de sortie floutées 1 fois, 2 fois et 5 fois.
- Insérer les profils d'une ligne de l'image d'entrée et des images de sortie floutées 1 fois, 2 fois et 5 fois en indiquant le numéro de la ligne choisie.
- Insérer les histogrammes de l'image d'entrée et des images de sortie floutées 1 fois, 2 fois et 5 fois.
- Indiquer ce que vous constatez.

5) Floutage de l'image couleur

- a) Ecrire un programme permettant de flouter une image couleur (ppm) dans sa globalité. En fait séparément sur chacune des composantes de l'image, chaque pixel de l'image va être remplacé par la valeur moyenne calculée avec ses voisins.
- b) Visualiser les histogrammes de l'image d'entrée et de l'image de sortie.

Dans votre compte rendu :

- Insérer l'image couleur originale ainsi que l'image floutée obtenue.
- Insérer les histogrammes de l'image d'entrée et de l'image de sortie.