TP7 : Encore des images

Dans ce TP, vous continuerez à manipuler des images en C.

1 Transformer une partie d'image en noir et blanc

On souhaite faire un effet de noir et blanc local dans une image. Dans l'image *tokyo.png*, on souhaite colorier les deux personnes en kimono au premier plan en noir et blanc. Cela permettra de faire croire qu'une ancienne photo a été incrustée sur la scène en couleur...

Pour ce faire, vous possédez une image *calque.png*, de la même dimension que *tokyo.png*, et où les pixels blancs représentent les pixels appartenant aux deux personnes que l'on souhaite afficher en noir et blanc. Proposez un programme permettant de générer, en sortie, une image où seules les deux personnes apparaissent en noir et blanc. Afin de rendre l'effet plus convaincant, nous ajouterons du "bruit" sur les personnes (et les personnes seulement), en rajoutant des valeurs aléatoires (pour simuler une certaine vieillesse du grain de l'image). Pour ce faire, vous rajouterez, aux pixels des personnes dans l'image de sortie, une valeur tirée aléatoirement dans l'intervalle [-25 ; 25] (attention de bien rester entre 0 et 255).

L'image ci-dessous représente le résultat que vous devriez obtenir à la fin.

Questions

Proposez un programme permettant de réaliser l'effet demandé.

2 Rendu de votre programme

Vous devez rendre le travail réalisé pour le TP7 sur Moodle.
Ce rendu doit suivre un format spécifique. Votre programme devra s'exécuter à l'aide de la commande

```
./incrustation entree.png calque.png sortie.png
```

- **entree.png** doit être l'image d'entrée en couleur.
- **calque.png** doit être l'image de calque, où les parties à transformer en noir et blanc apparaissent en blanc.
- **sortie.png** sera le fichier PNG de sortie de votre programme.
- Si votre programme ne parvient pas à se terminer correctement (fichier d'entrée introuvable, mauvais nombre de paramètres en entrée du programme, etc), il devra renvoyer **EXIT_FAILURE**. Sinon, il devra se terminer avec le code **EXIT_SUCCESS**.

Votre programme devra être compilé à l'aide de la commande

```
gcc incrustation.c lodepng.c -o incrustation
```

Si votre ligne de compilation est plus complexe, vous devrez alors la spécifier dans un *Makefile*. Tous vos fichiers (y compris les fichiers lodepng.c et lodepng.h) devront être directement placés dans un fichier zip (et ne pas être dans un sous dossier du fichier zip), dont le nom sera

```
VotreNom_VotrePrenom_VotreNumeroEtudiant.zip
```

sans aucun espace (si votre nom ou votre prénom contiennent un espace, ne les faîtes pas figurer). Les formats de compression acceptés sont .zip, .tgz, .tar.gz, .7z, .rar.

Si vous avez un binôme, il vous faudra écrire son nom, prénom et numéro d'étudiant séparés par des espaces, en plus des vôtres, dans un fichier *binome.txt*, dont le format sera

```
Nom1 Prenom1 Numero_etudiant1
Nom2 Prenom2 Numero_etudiant2
```

Vous ne devez rendre le fichier qu'une seule fois par binôme.

3 Le détecteur de contour

On souhaite détecter les contours des objets sur une image noir et blanc, comment montré sur l'exemple suivant :

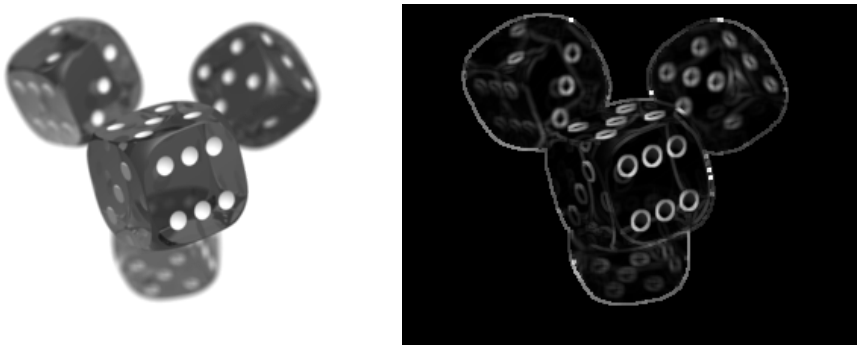

FIGURE 1 – A gauche l'image originale, à droite le résultat du détecteur de contour

Pour ce faire, on réalise l'opération suivante : pour chaque pixel p de l'image, on mesure la valeur maximale M ainsi que la valeur minimale m de p et de ses huit voisins. Dans une image de sortie I', on attribue au pixel p la valeur $M - m$.

> **Questions**
>
> Proposez un programme permettant de calculer les contours d'une image (en noir et blanc) donnée en entrée. Le nom de l'image à ouvrir et de l'image à écrire doit être spécifié en paramètre de votre programme (et non pas codé directement dans le programme). On ne s'occupera pas des pixels situés sur les bords de l'image.

4 Rendu de votre programme

Vous devez rendre le travail réalisé pour le TP7 sur Moodle.
Ce rendu doit suivre un format spécifique. Votre programme devra s'exécuter à l'aide de la commande

```
./gradient entree.png sortie.png
```

— **entree.png** doit être l'image d'entrée.
— **sortie.png** sera le fichier PNG de sortie de votre programme.
— Si votre programme ne parvient pas à se terminer correctement (fichier d'entrée introuvable, mauvais nombre de paramètres en entrée du programme, etc), il devra renvoyer **EXIT_FAILURE**. Sinon, il devra se terminer avec le code **EXIT_SUCCESS**.

Votre programme devra être compilé à l'aide de la commande

```
gcc gradient.c lodepng.c -o gradient
```

Si votre ligne de compilation est plus complexe, vous devrez alors la spécifier dans un *Makefile*. Tous vos fichiers (y compris les fichiers lodepng.c et lodepng.h) devront être directement placés dans un fichier zip (et ne pas être dans un sous dossier du fichier zip), dont le nom sera

```
VotreNom_VotrePrenom_VotreNumeroEtudiant.zip
```

sans aucun espace (si votre nom ou votre prénom contiennent un espace, ne les faîtes pas figurer). Les formats de compression acceptés sont .zip, .tgz, .tar.gz, .7z, .rar.

Si vous avez un binôme, il vous faudra écrire son nom, prénom et numéro d'étudiant séparés par des espaces, en plus des vôtres, dans un fichier *binome.txt*, dont le format sera

```
Nom1 Prenom1 Numero_etudiant1
Nom2 Prenom2 Numero_etudiant2
```

Vous ne devez rendre le fichier qu'une seule fois par binôme.

5 Retirer du bruit à une image

Dans une image, il existe parfois des parasites qui empêchent de bien voir ce qui est représenté dessus. Par exemple, si vous regardez l'image *bruit.png* représentée ci-dessous, vous verrez de petits points blancs qui semblent ne pas appartenir à l'image d'origine.

Pour retirer ces points blancs (appelé donc le bruit), différentes techniques, plus ou moins efficaces, existent. La méthode du filtre médian est le suivante : pour chaque pixel p de l'image, on regarde les valeurs des 8 pixels alentours (cela fait en tout 9 valeurs à examiner : le pixel en lui-même et ses 8 voisins) et on calcule la valeur médiane m de ces valeurs. Dans une nouvelle image I', on attribue au pixel p la valeur m.

> **Questions**
>
> Proposez un programme permettant de retirer, à l'aide d'un filtre médian, le bruit d'une image (en noir et blanc) donnée en entrée. Le nom de l'image à ouvrir et de l'image à écrire doivent être spécifié en paramètre de votre programme (et non pas codé directement dans le programme). On ne s'occupera pas des pixels situés sur les bords de l'image.