

Introduction à l'Informatique par le Web

TD numéro 3

Algorithme et utilisation de fichiers

Objectif

L'objectif de ce TD est d'écrire des algorithmes en rapport avec de la manipulation de fichiers. On y trouve également un exercice de logique pour se détendre un peu entre deux algorithmes.

Rappels

- La sémantique d'un opérateur logique et plus généralement d'une formule de logique est donnée par une table de vérité.

1 format PPM simplifié

Les images manipulées la semaine précédente étaient au format PPM, sous format P3. Une image au format PPM P3 est stockée dans un fichier comme illustré sur la figure suivante.

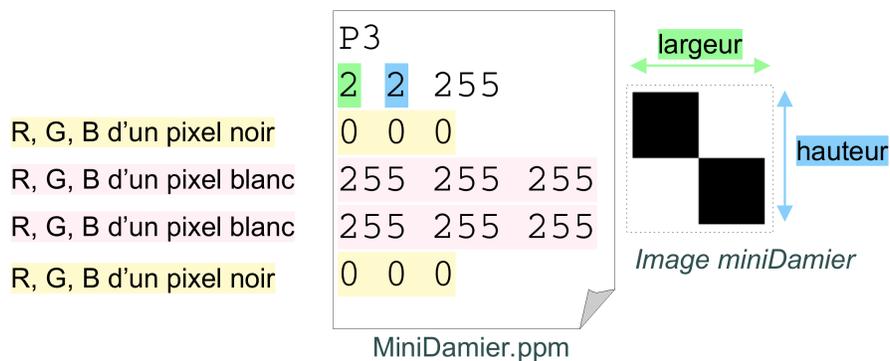


Figure 1: Un exemple de sérialisation d'une image de 2 pixels sur 2 pixels au format PPM-P3

le format PPM ne spécifiant pas si les différentes valeurs doivent être séparées par des espaces ou des retours à la ligne, vous ferez un choix.

1.1 Sérialisation d'images

Écrire un algorithme qui étant donné une variable nommée `img` de type `ImagePPM` permet de créer un fichier nommé `result.ppm` au format ppm sur le disque.

2 Désérialisation d'images

Étant donné un fichier au format PPM, sous format P3 tel que sérialisé par l'algorithme ci dessus, écrivez un algorithme permettant de créer une variable de type `ImagePPM` à partir de ce fichier.

3 format IIW

On décide d'inventer un nouveau format de sérialisation qui, on l'espère, devrait permettre de prendre moins de place sur le disque dur pour chaque image. Pour ce faire on ne veut plus stocker les couleurs de chacun des pixels dans le fichiers mais uniquement les changements de couleur entre deux pixels successifs. Pour ce faire nous sommes obligé de stocker le nombre de pixel égaux successifs comme illustré dans la figure suivante.

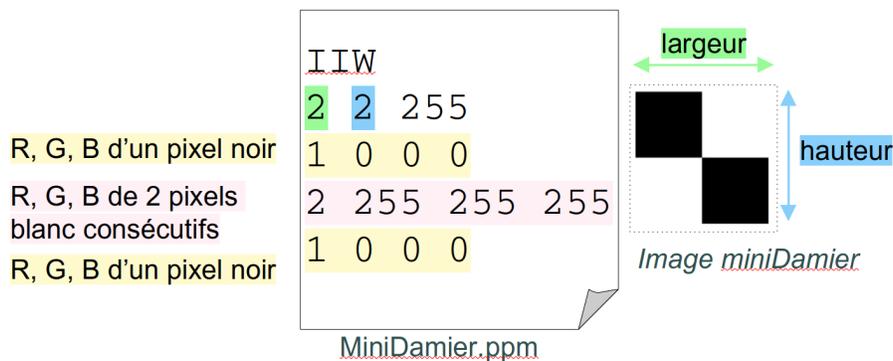


Figure 2: Un exemple de sérialisation d'une image de 2 pixels sur 2 pixels au format IIW

Selon vous, pour quel genre d'image ce format est-il intéressant ?

Comment faire pour rendre ce format avantageux dans plus de cas ? Quelle est la conséquence ?

Écrivez l'algorithme de sérialisation et désérialisation associé à ce format.

4 Compression d'image avec perte

Afin de tirer partie du format IIW on décide de compresser les images. Cette compression engendrera une perte de qualité. En effet dans l'image compressée on désire remplacer une suite de n pixels par n pixels ayant pour valeur r, g, b la moyenne des valeurs r, g, b des n pixels d'origine. Une fois la compression réalisée, l'utilisation du format IIW induira une réduction significative de la taille sur le disque dur. La figure suivante montre l'image du parc de valrose avant compression (à gauche) et avec une compression (trop forte ?, à droite) ou $n == 8$. La taille du fichier résultant sur le disque est alors de 15% de celui d'origine ! À noter que pour $n == 2$ la compression est visuellement très acceptable et l'image résultante occupe 60% de la place de l'image d'origine sur le disque.

4.1 version 1

Dans une première version, considérez que n est un multiple de la largeur de l'image. Écrire l'algorithme qui étant donné une variable `img` de type `ImagePPM`, crée une nouvelle variable `res` de type `ImagePPM`

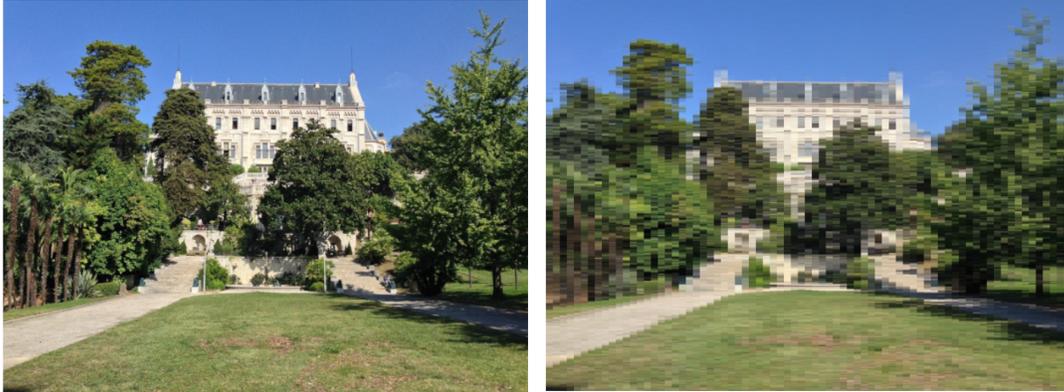


Figure 3: à gauche l'image d'origine, à droite l'image (trop) compressée

compressée. Réfléchissez bien à combien d'objet de type pixels vous voulez créer dans la nouvelle image.

4.1.1 version 2

Dans la deuxième version, considérez le cas général, c'est à dire qu'il peut y avoir en fin de ligne m pixels restant tel que $m < n$. Dans ce cas vous ne moyennerez que ces m pixels pour éviter de moyennner des pixels à cheval sur plusieurs lignes. Modifiez l'algorithme de la question précédente pour prendre ce cas en considération.

5 Une princesse ou un tigre ?

Inspiré par la deuxième épreuve du livre qui rend fou¹:

Suite au cours #3 de cette UE, un nouvel étudiant doit résoudre une énigme afin d'éviter d'ouvrir une cellule contenant un tigre. Les affiches des portes de cellule sont les suivantes:

1. Une au moins des deux cellules contient une princesse
2. Il y a une tigre dans l'autre cellule

Le roi ajoute: "Les affiches sont sincères toutes les deux ou bien elles sont fausses toutes les deux." Quelle porte l'étudiant doit-il ouvrir pour ne pas se faire dévorer ?

6 format PPM complet

Dans le format PPM, rien ne dit si on sépare différentes valeurs par un espace ou un retour à la ligne ou un mélange des deux. De plus, on peut à tout moment avoir une ligne de commentaire commençant par le caractère #. On peut également avoir une ligne vide à tout moment.

Écrire l'algorithme de sérialisation du format PPM non simplifié.

7 format IIW++

On décide d'inventer encore une nouvelle manière de sérialiser les images. Cette fois le fichier contiendra 2 parties. La première partie contiendra uniquement les Pixel de couleurs différentes utilisés dans l'image (c'est à dire la liste des couleurs utilisées). La deuxième partie contiendra *largeur * hauteur*

¹<https://www.dunod.com/sciences-techniques/livre-qui-rend-fou-casse-tetes-logiques-en-perdre-raison>

nombres où chaque nombre donnera la position dans la liste de la première partie de la couleur à utiliser pour le Pixel. La figure suivante illustre ce concept

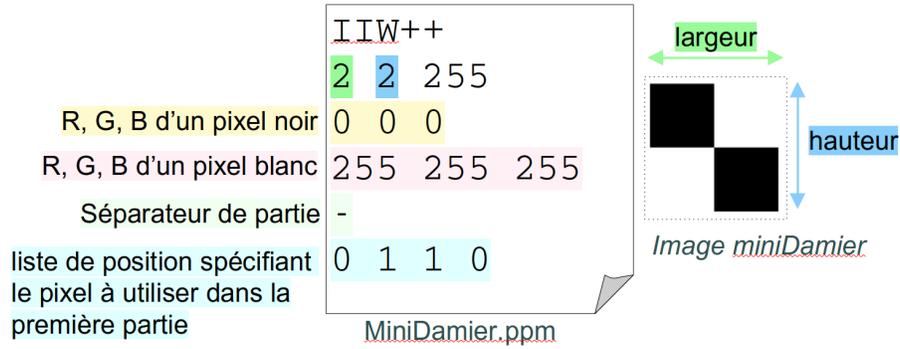


Figure 4: Un exemple de sérialisation d'une image de 2 pixels sur 2 pixels au format I IW++

Écrivez l'algorithme de sérialisation associé à ce format.

Modifier votre algorithme pour afficher à l'écran le nombre de couleurs différentes utilisées. Affichez également le ratio entre le nombre de pixels et le nombre de couleurs utilisées. Que représente ce ratio