HAI804I – Analyse et Traitement d'Images

Fabien Caballero

March 24, 2023

Contents

1 Lire le fichier **2**

2 MIP **4**

3 AIP **5**

1 Lire le fichier

J'utilise les fonctions fopen et fread avec fopen en mode lecture binaire.

```
    FILE *f = fopen((char *)argv[1], "rb");
     unsigned short *contenu;

     int dimX = atoi(argv[2]);
     int dimY = atoi(argv[3]);
     int dimZ = atoi(argv[4]);
     allocation_tableau(contenu, unsigned short, dimX *dimY *dimZ);

     size_t taille=fread(contenu,sizeof(unsigned short),dimX*dimY*dimZ,f);
```

Il faut penser à inverser le sens de lecture des 2 octets (MSB et LSB) En récupérant les valeurs dans un unsigned short le LSB et MSB sont inversés. Pour ne pas inverser il faudrait stocker chaque élément dans 2 unsigned char. J'ai choisi les unsigned short et d'inverser.

```
  unsigned short inverse(unsigned short val)
{
    float o1 = floor(((double)val / 256.0));
    float o2 = val - o1 * 256;

    return o2 * 256 + o1;
}
```

Une fois tout le tableau inversé, j'ai créer une fonction getValue, une max et une min.

```

unsigned short getValue(unsigned short *img, int x, int y, int z, int dimX, int dimY, int
    dimZ)
{
    return (int)img[z * dimY * dimX + y * dimX + x];
}

unsigned short minElmt(unsigned short *img, size_t taille)
{
    unsigned short min = 0;
    for (size_t i = 0; i < taille; i++)
    {
        if (img[i] < min)
            min = img[i];
    }
    return min;
}

unsigned short maxElmt(unsigned short *img, size_t taille)
{
    unsigned short max = 0;
    for (size_t i = 0; i < taille; i++)
    {
        if (img[i] > max)
            max = img[i];
    }
    return max;
}
```

J'obtient les résultats suivants:

t1-head:
val(158,143,64)= 242 min= 0 max= 885
orange:
val(128,128,32)= 9 min= 0 max= 228
INCISIX:
val(184,343,83)= 1225 min= 0 max= 4095

2 MIP

```
1       for (size_t i = 0; i < dimX; i++)
2       {
3           for (size_t j = 0; j < dimY; j++)
4           {
5               int max = 0;
6               for (size_t k = 0; k < dimZ; k++)
7               {
8                   if (getValue(contenu, i, j, k, dimX, dimY, dimZ) > max)
9                       max = getValue(contenu, i, j, k, dimX, dimY, dimZ);
10              }
11              Out[j*dimX+i]=max;
12          }
13      }
```

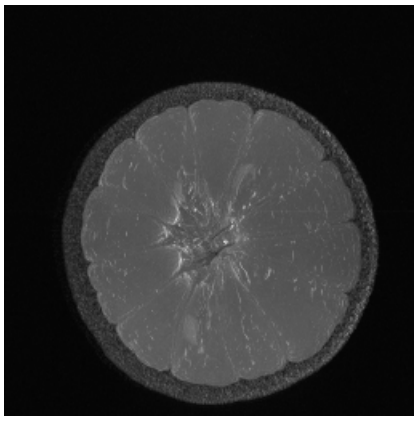

Figure 1: MIP orange

Figure 2: MIP INCISIX

Figure 3: MIP t1-head

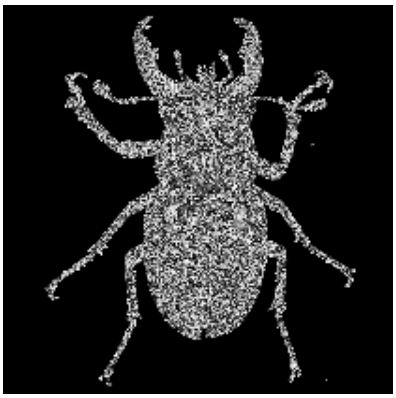

Figure 4: MIP whatisit

3 AIP

```
1       for (size_t i = 0; i < dimX; i++)
2       {
3           for (size_t j = 0; j < dimY; j++)
4           {
5               int max = 0;
6               for (size_t k = 0; k < dimZ; k++)
7               {
8                   // if (getValue(contenu, i, j, k, dimX, dimY, dimZ) < max)
9                   max += getValue(contenu, i, j, k, dimX, dimY, dimZ);
10              }
11              max=max/dimZ;
12              if(max>255) max=255;
13
14              Out[j * dimX + i] = moy ;
15          }
16      }
```

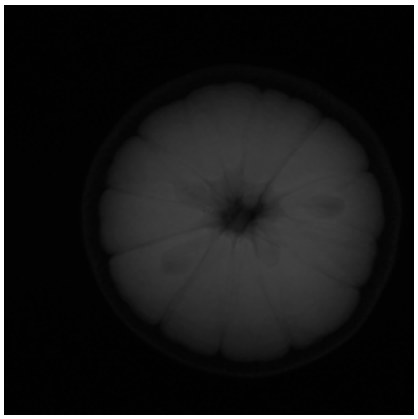

Figure 5: AIP orange

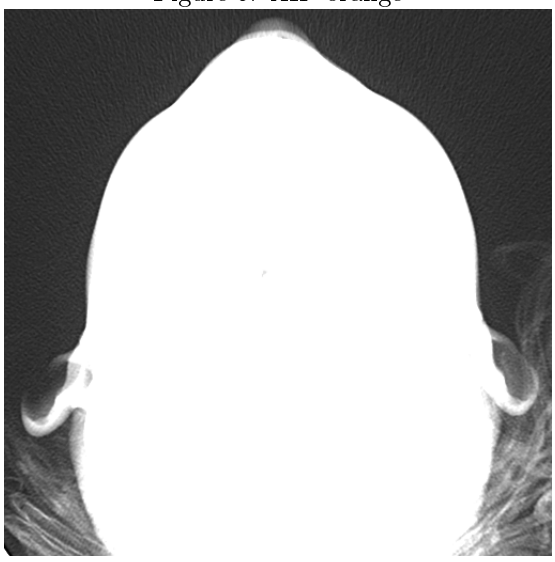

Figure 6: AIP INCISIX

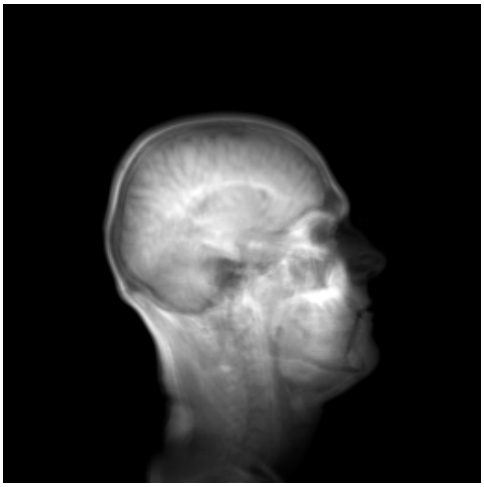

Figure 7: AIP t1-head

Figure 8: AIP whatisit