

Programmation C

Traitement d'image au format pgm

—

Thibault Chatillon - Décembre 2020

Au programme

- But du projet
- Réalisations
- Structure de fichiers
- Exemple concret
- Petite démo
- Conclusion

But du projet :

- Lire, manipuler, sauvegarder et afficher des images sous format pgm



.PGM

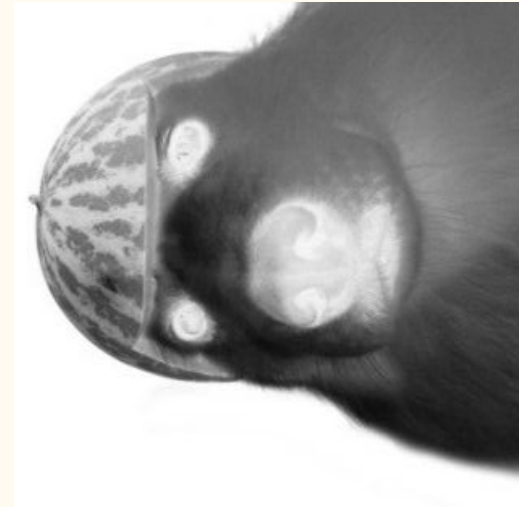
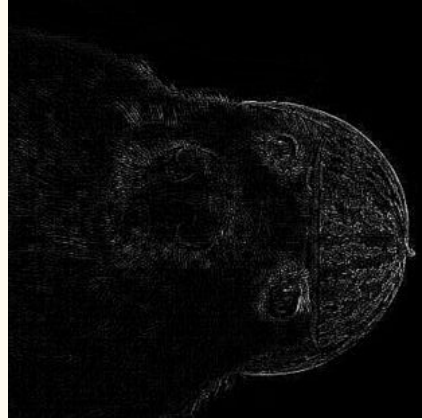


.PGM

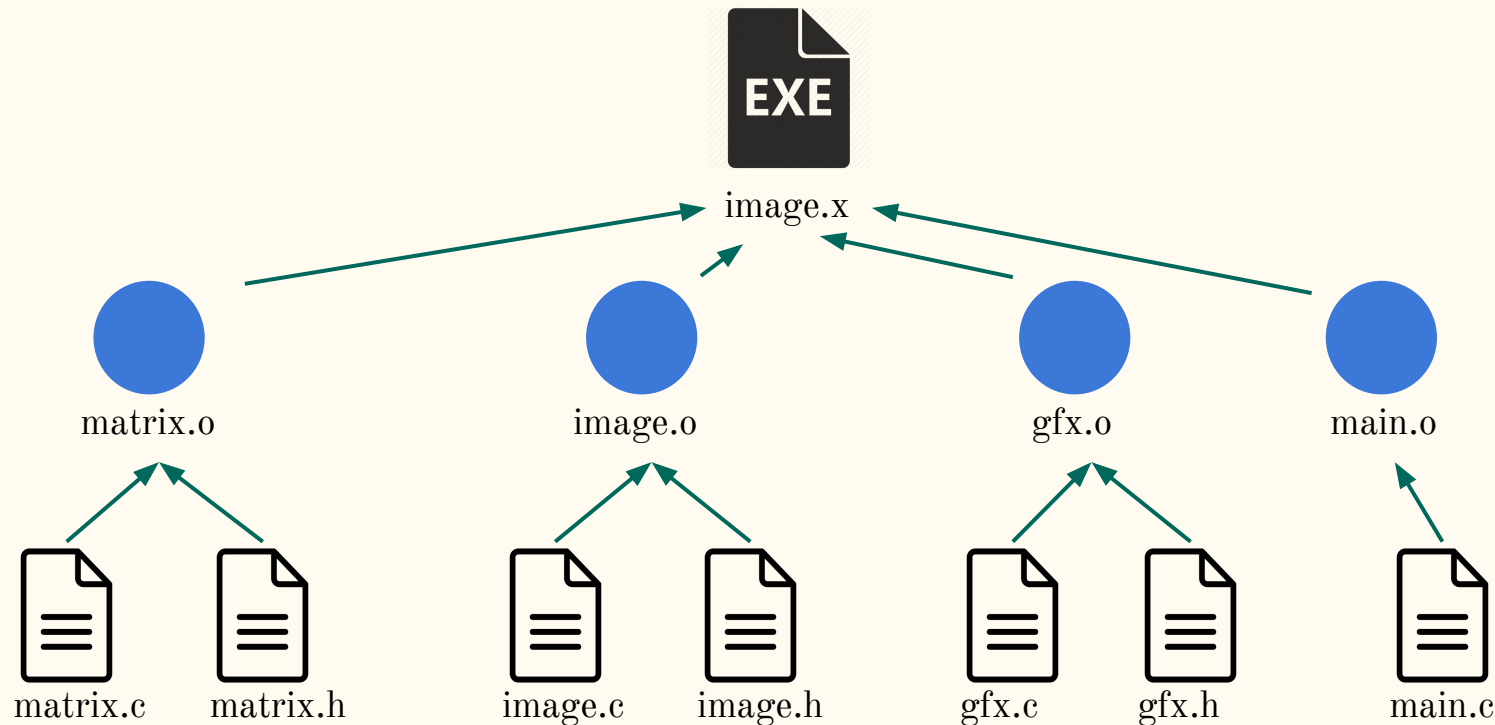
Fonctions :

- Lire un fichier pgm et le garder en mémoire
- Ecrire un fichier pgm a partir de la mémoire
- Faire le négatif d'un image
- Faire la symétrie horizontale, verticale et centrale
- Faire un “photomaton” d'une image
- Rogner une image
- Appliquer diverses filtres : flou, netteté améliorée, détection des contours ...
- Afficher l'image

Quelques réalisations



Structure de fichiers



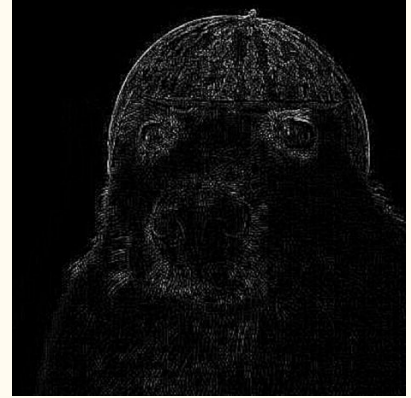
Filtres de convolution



Edge detection



$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



35	50	45	46	67
36	47	22	25	29
38	52	51	50	54
42	35	47	49	51
53	62	63	66	70

\times

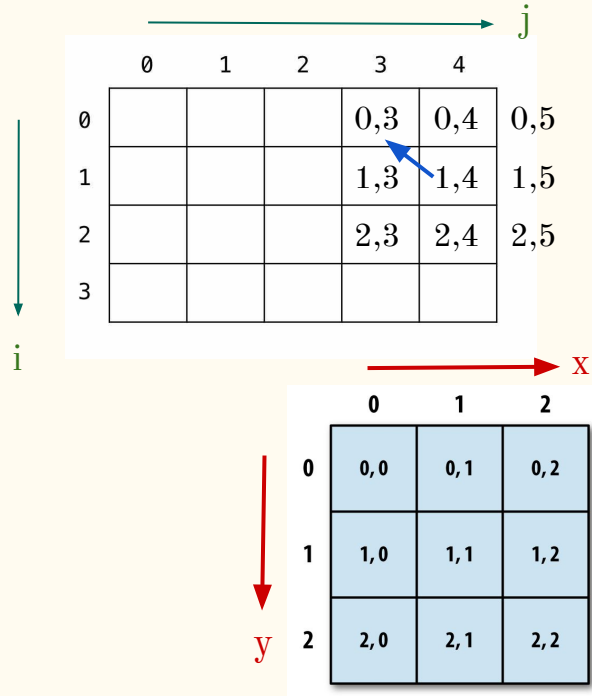
0	1	0
1	4	1
0	1	0

$=$

		47		

Divisor = $0 + 1 + 0 + 1 + 4 + 1 + 0 + 1 + 0 = 8$

Concrètement :



```
int divider = get_divider(kernel);

for (int i = 0; i < orig->pixels.m; i++) {
    for (int j = 0; j < orig->pixels.n; j++) {
        double pixel = 0.0;

        for (int y = 0; y < kernel->m; y++) {
            for (int x = 0; x < kernel->n; x++) {

                int yi = i - kernel->m / 2 + y;
                int xj = j - kernel->n / 2 + x;
                //if pixel is inside matrix
                if (xj >= 0 && yi >= 0 && xj < orig->pixels.m && yi < orig->pixels.n) {
                    pixel += orig->pixels.data[yi][xj] * kernel->data[y][x];
                }
            }
        }
        pixel /= divider;
        //check pixel's val
        if (pixel < 0) {
            pixel = 0;
        } else if (pixel > conv->max) {
            pixel = conv->max;
        }
        //set pixel
        conv->pixels.data[i][j] = (int)pixel;
    }
}
```


Petite démo

Conclusion

- Projet 99.9% fonctionnel !
- Aucun mandrill maltraité (promis)
- Convolution possible avec des kernels de tailles différentes
- A améliorer : recadrage d'interval des pixels

Questions ?