



# Programmation Séquentielle

Thomas Dagier

11 Décembre 2019

# Introduction



Prise en main du langage C et manipulation de bibliothèques

Mathématiques et algorithmique pratique

Application concrète de manipulation d'image



## I / Présentation du code

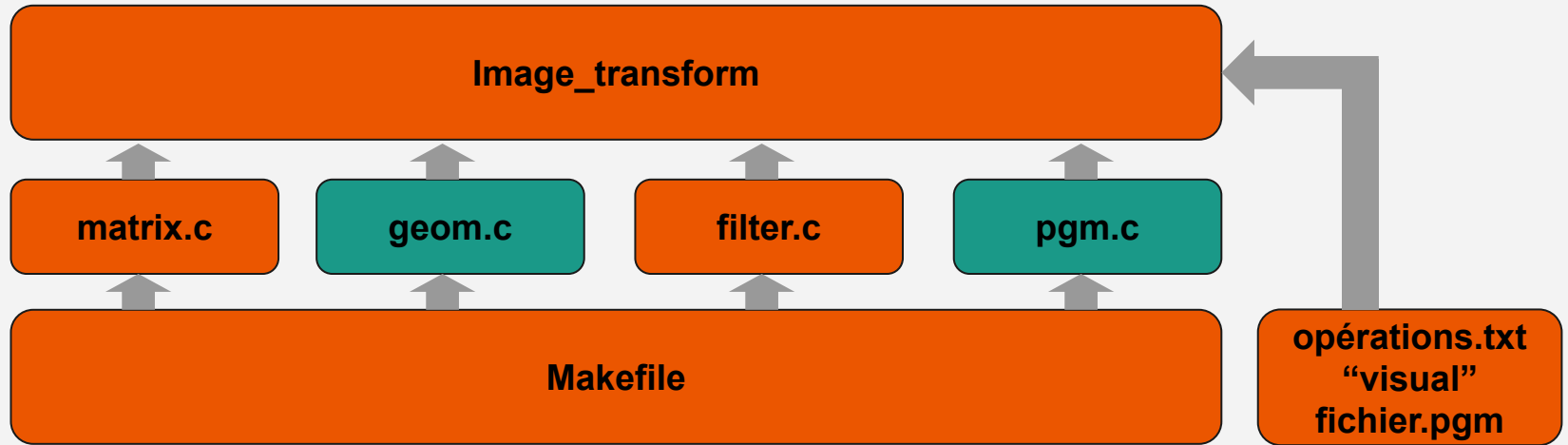
- appréhension du problème
- difficultés rencontrées
- solution proposée

## II / Démonstration

- étude d'un fichier d'opérations
- application des fonctionnalités du projet

## III / Conclusion

# Présentation du code : structure du projet



## Présentation du code : lire un fichier pgm

ouvrir le fichier pgm et créer un élément de type pgm

**pour chaque ligne :**

si on trouve **m** et **n** → ajouter m et n à la structure

si on trouve la valeur **max** → ajouter max à la structure

si on tombe sur des valeurs →

**tant qu'on trouve un nombre séparé d'un espace :**

**on ajoute la valeur normalisée à la matrice**

**fin tant que**

**fin pour chaque**

libérer la mémoire

## Présentation du code : appliquer une rotation

on a une matrice, les coordonnées du centre de rotation et un angle :

**pour chaque ligne :**

**pour chaque colonne :**

trouver  $P(x ; y)$  en fonction du centre

trouver  $P'(x' ; y')$  si on fait une rotation

**appliquer un round()**

**fin pour chaque**

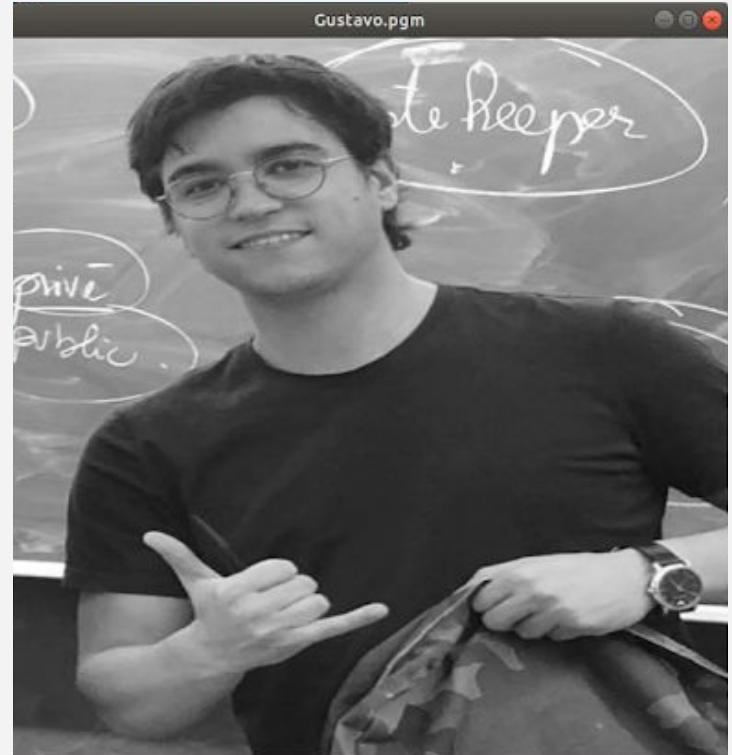
**fin pour chaque**

ajouter les valeurs du clone à la matrice

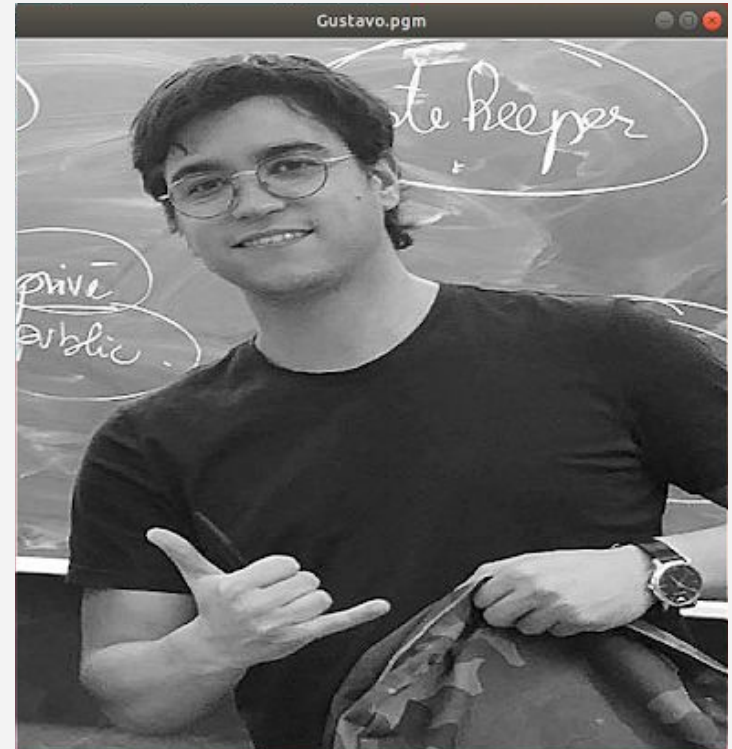
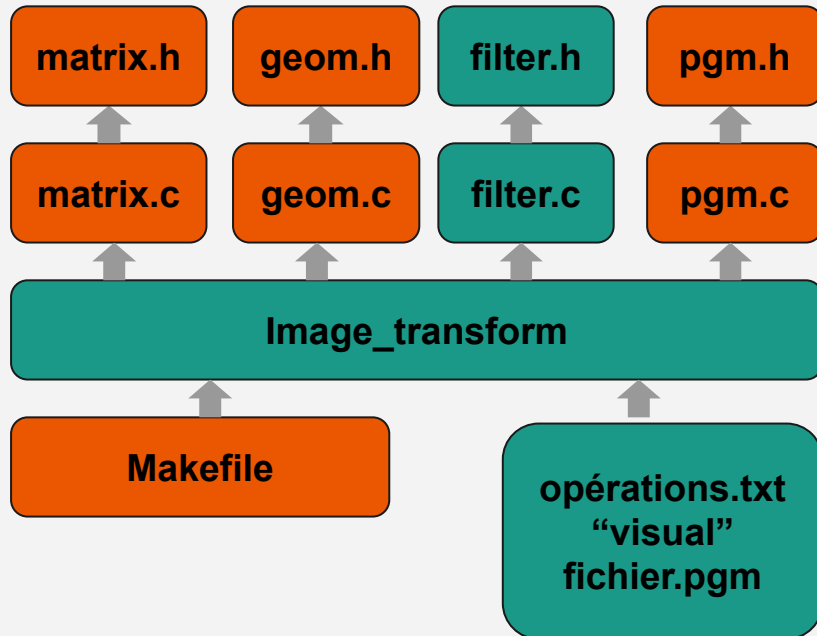
libérer le clone

# Démonstration : modifications d'un fichier pgm

```
zoom 1,5 1,5
#le zoom n'a pas marché
read gustavo.pgm
translate 100 100
rotate 300 300 un angle
# 0.78539816339 correspond à  $\pi/4$ 
rotate 300 300 0.78539816339
fonction_au_hasard
sharpen
symmetry vertical
photomaton
sobel
write result.pgm
```



## Démonstration : ajout d'un filtre

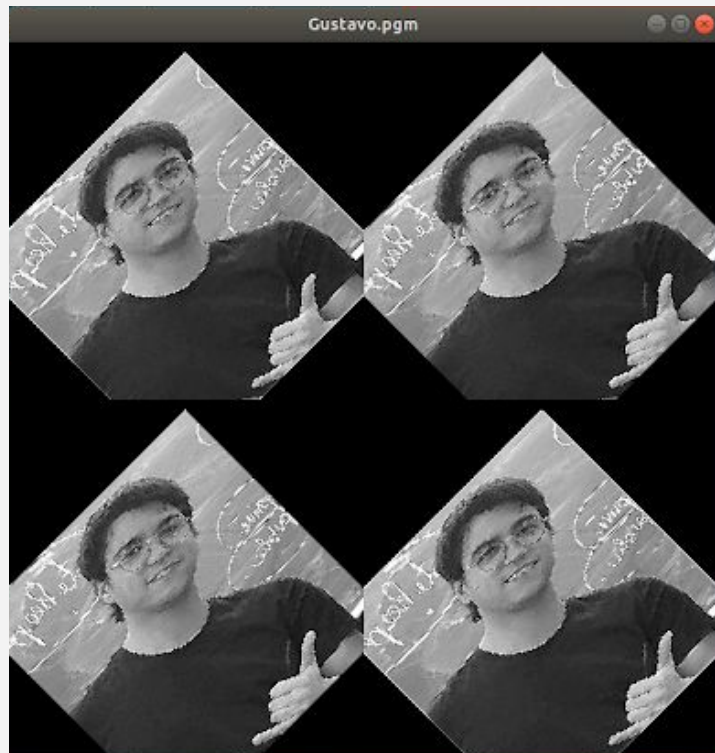




# Conclusion

apport sur le plan technique

un point plus personnel



# Questions

