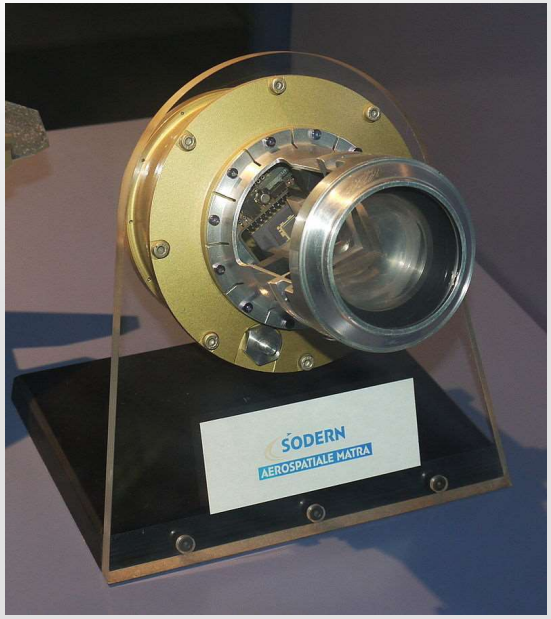
La reconnaissance de formes appliquée à l'identification de constellations

La reconnaissance de constellations

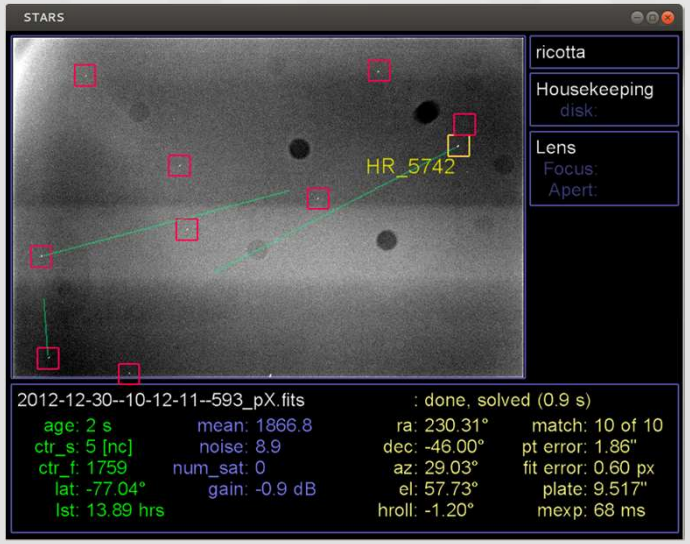
- Sert au repérage des bateaux jusqu'au XXème siècle

- Est désormais utilisé pour orienter des engins spatiaux

Tête optique d'un viseur d'étoiles



Le logiciel viseur d'étoiles STARS



Qu’est-ce qu’une constellation ?

- Groupe d’étoiles apparent
- Suffisamment proches

2 objectifs :

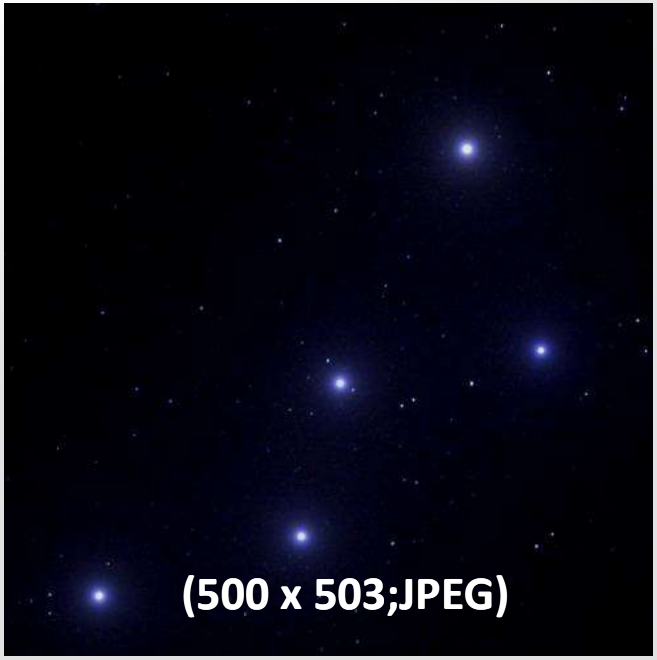
- Définir une constellation avec des données géométriques

- Identifier une constellation

Outil de travail : Python

- Image jpeg : Tableau de triplets (R,V,B) dont l'intensité est codée sur 8 bits (0 à 255)

Outil d'étude : Cassiopée

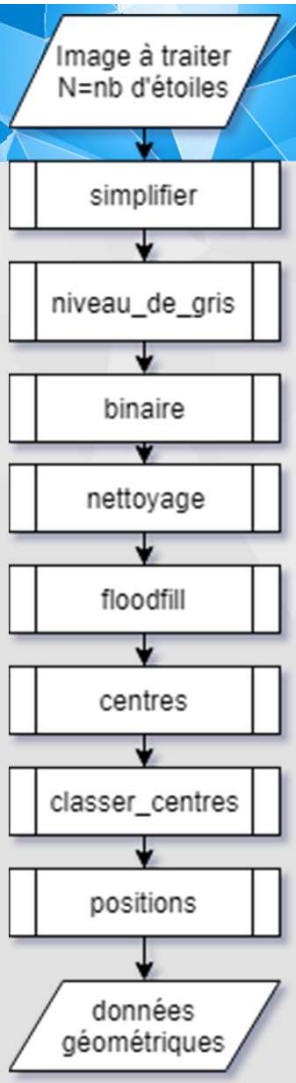


Objectif 1 :

DÉFINIR UNE CONSTELLATION GÉOMÉTRIQUEMENT

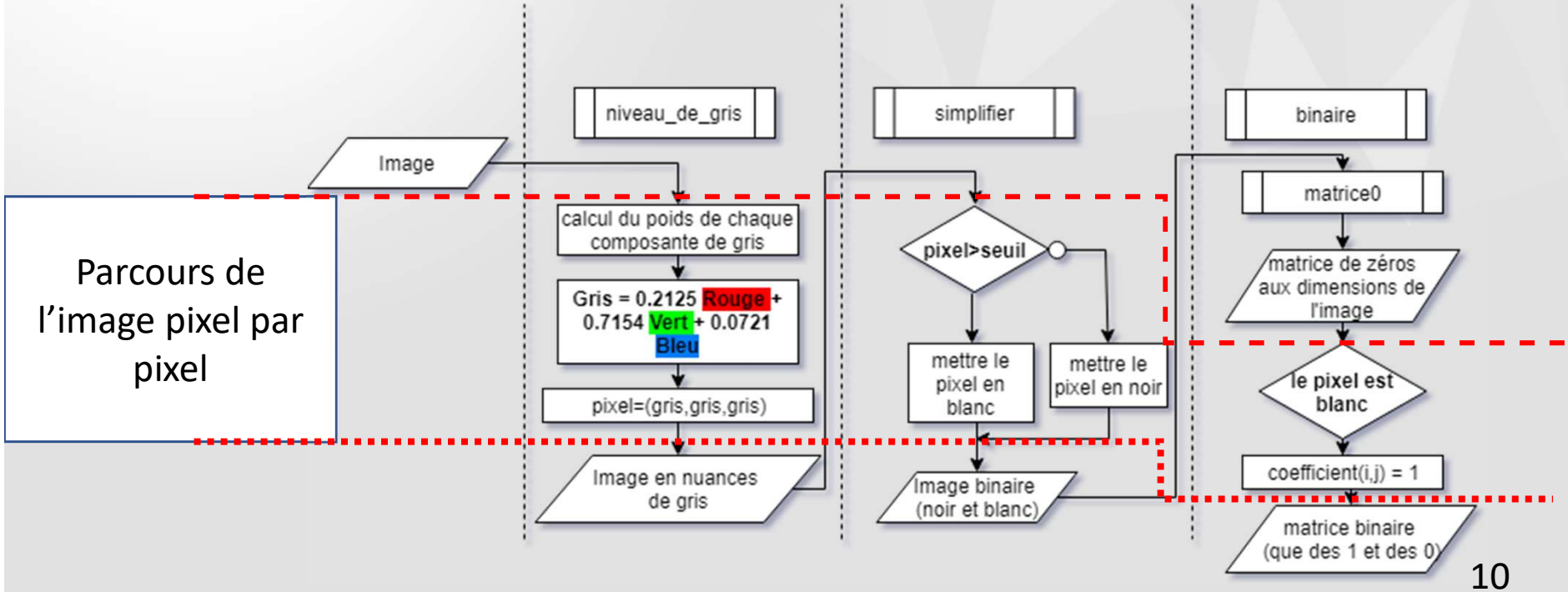
La fonction définir()

- 7 sous-fonctions
- Renvoie des données géométriques propres



Passage d'une image à une matrice binaire

- Assuré par les fonctions *simplifier*, *niveau_de_gris* et *binaire*



Passage d'une image à une matrice binaire

Image en niveau de gris

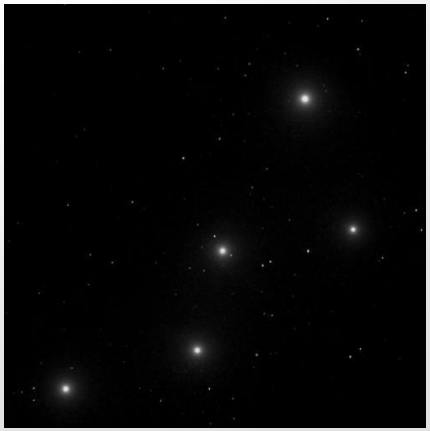
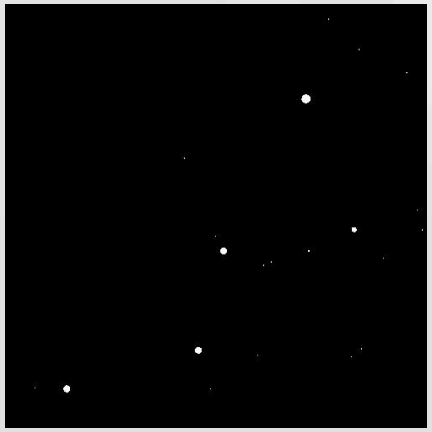


Image binaire

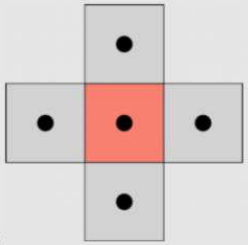


Matrice binaire

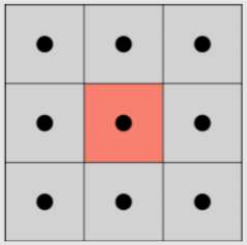
0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	1
0	1	1	0	0	1	0	1	1
0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Analyses des composantes connexes

- **Connexité**
  - **Pixels « connectés » ?**



4-connexité



8-connexité

***Rem.*: pour les régions, on choisit la 4-connexité**

  - **Exemples de composantes connexes**

0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	1
0	1	1	0	0	1	0	1	1
0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Image binaire

0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	1
0	1	1	0	0	1	0	1	1
0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Composantes 4-connexes

0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	1
0	1	1	0	0	1	0	1	1
0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Composantes 8-connexes

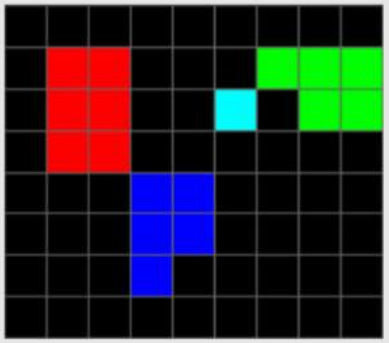
- **Étiquetage**
  - **Chaque composante connexe est identifiée de manière unique par une étiquette (*label*)**

0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	1
0	1	1	0	0	1	0	1	1
0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0

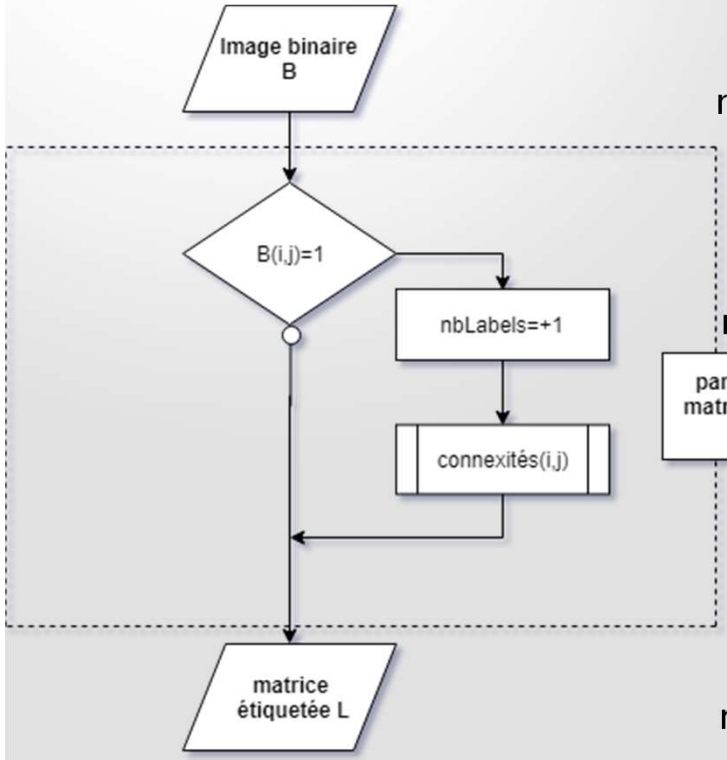
- Étiquettes numériques (entiers)

0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	2	2	2
0	1	1	0	0	3	0	2	2
0	1	1	0	0	0	0	0	0
0	0	0	4	4	0	0	0	0
0	0	0	4	4	0	0	0	0
0	0	0	4	0	0	0	0	0
0	0	0	0	0	0	0	0	0

- Étiquettes de couleurs



Méthode du remplissage (*floodfill*)



parcours de la matrice pixel par pixel

nbLabels=0

nbLabels=1

nbLabels=1

nbLabels=1

0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	1	1	1	0
0	1	0	0	1	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	1	1	1	0
0	1	0	0	1	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	1	1	1	0
0	1	0	0	1	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	1	1	0
0	1	0	0	1	0	0
0	0	0	0	0	0	0

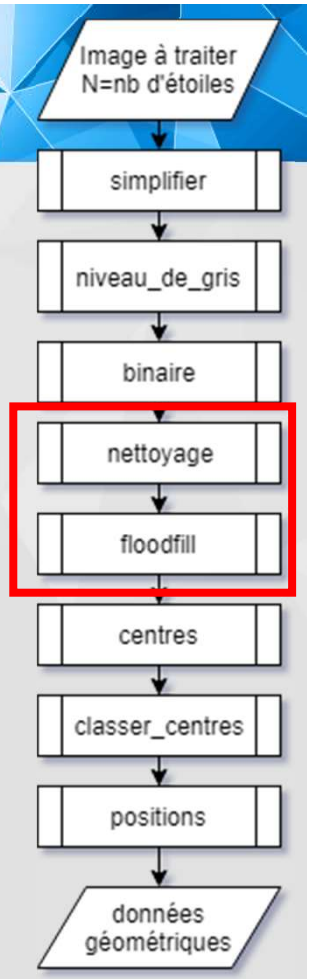
L

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

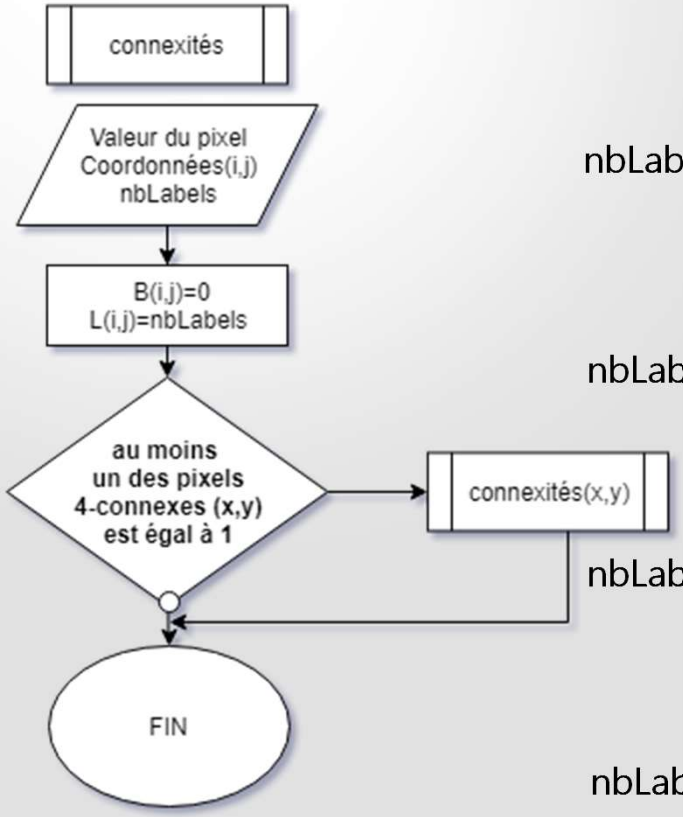
0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0



La visée d'étoiles

Objectif 1 : Définir

Analyse des composantes connexes



nbLabels=1

nbLabels=2

nbLabels=2

nbLabels=2

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	1	1	0
0	0	0	0	1	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	1	1	0
0	0	0	0	1	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	0
0	0	0	0	1	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	0	0	0	0	0

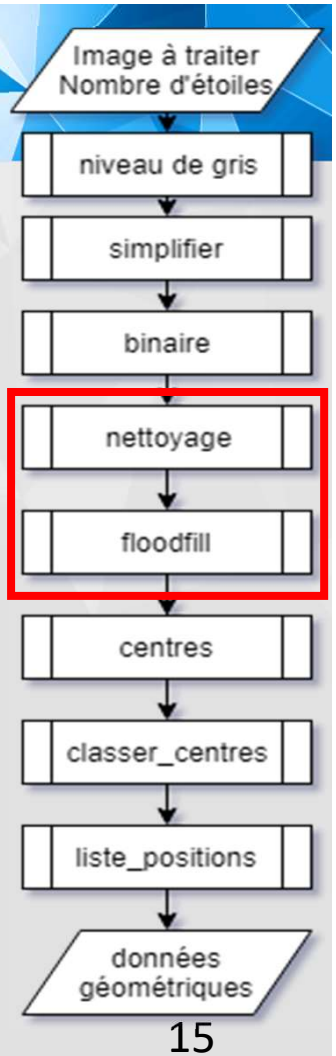
L

0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	2	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	2	2	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	2	2	2	0
0	1	0	0	2	0	0
0	0	0	0	0	0	0



Analyse des composantes connexes

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

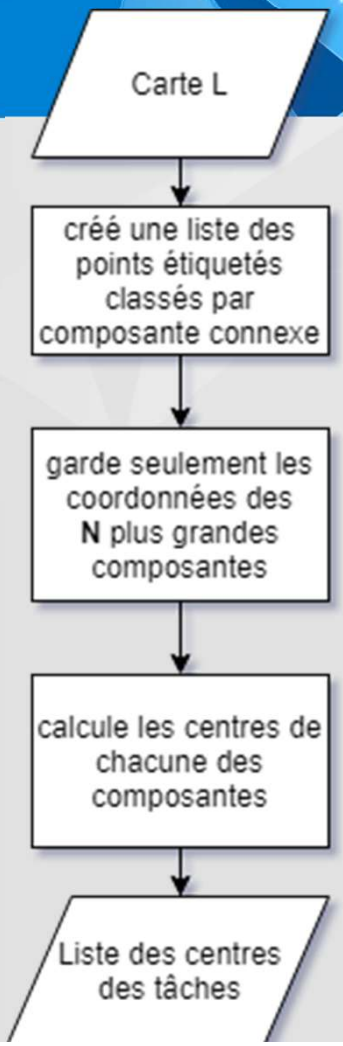
L

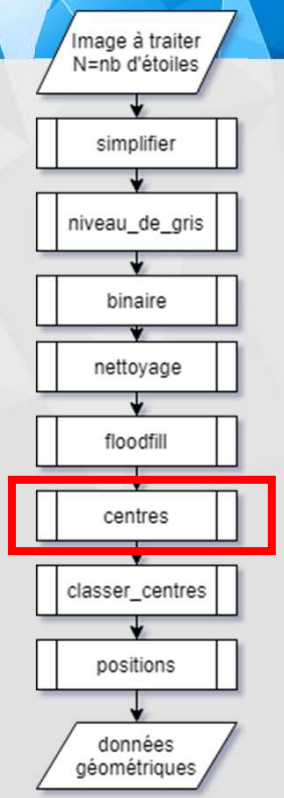
0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	2	2	2	0
0	1	0	0	2	0	0
0	0	0	0	0	0	0

Carte d'étiquettes L

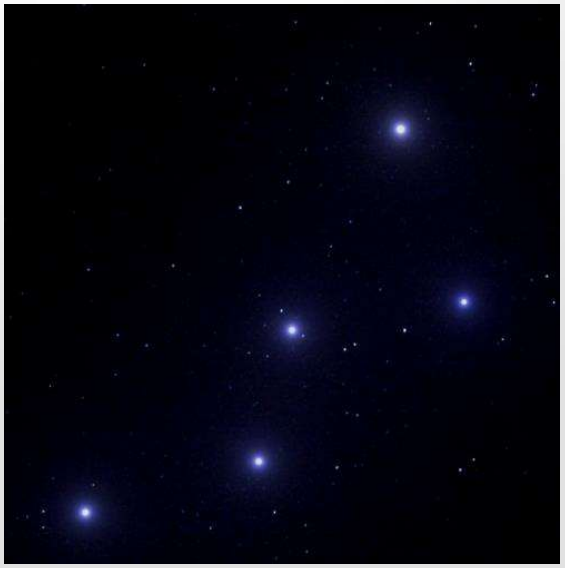
La fonction *centre()*

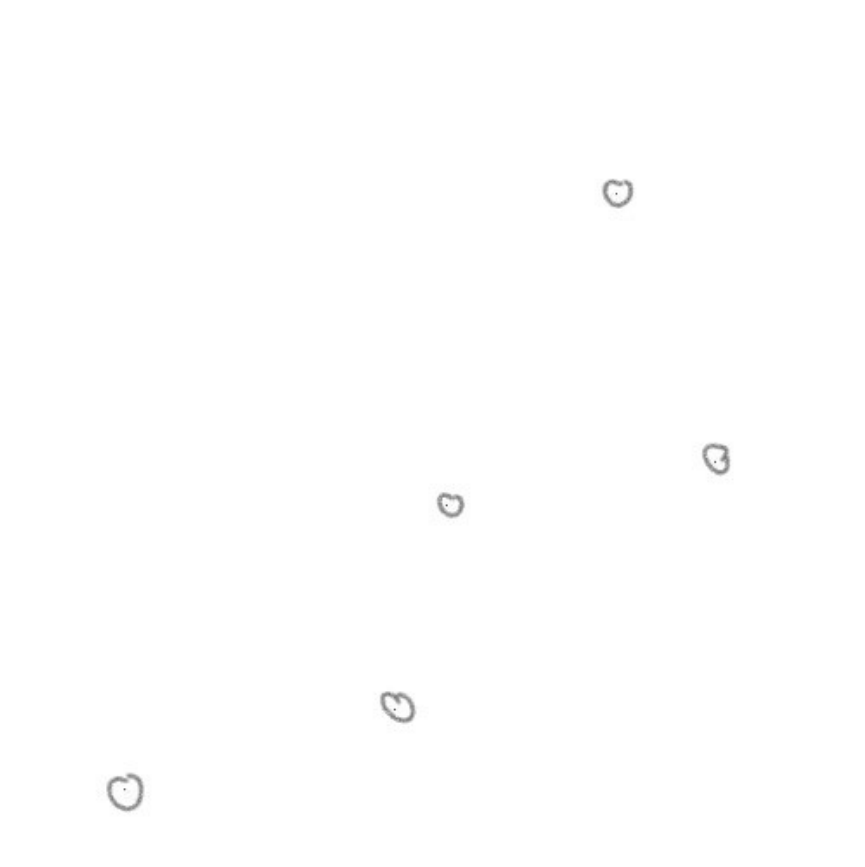
- Prend une **carte étiquetée L** en argument
- **N** le nombre d'étoiles
- Renvoie la **liste des coordonnées des centres** des étoiles de la constellation





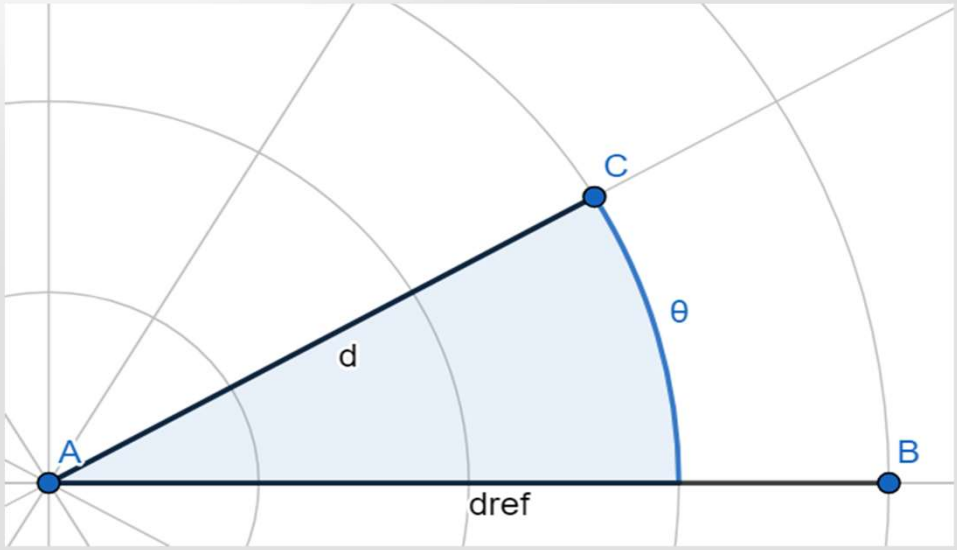
La fonction *centre()*





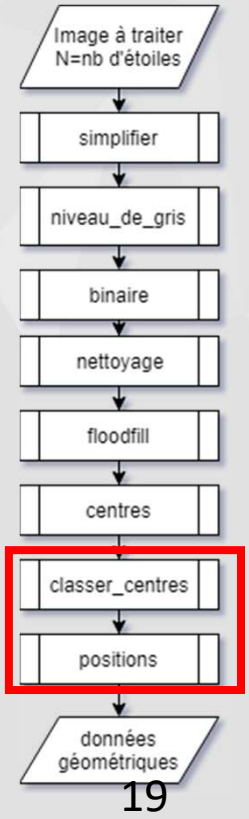
Repérer les points dans une base propre à la constellation

- Assuré par les fonctions *classer_centres()* et *positions()*



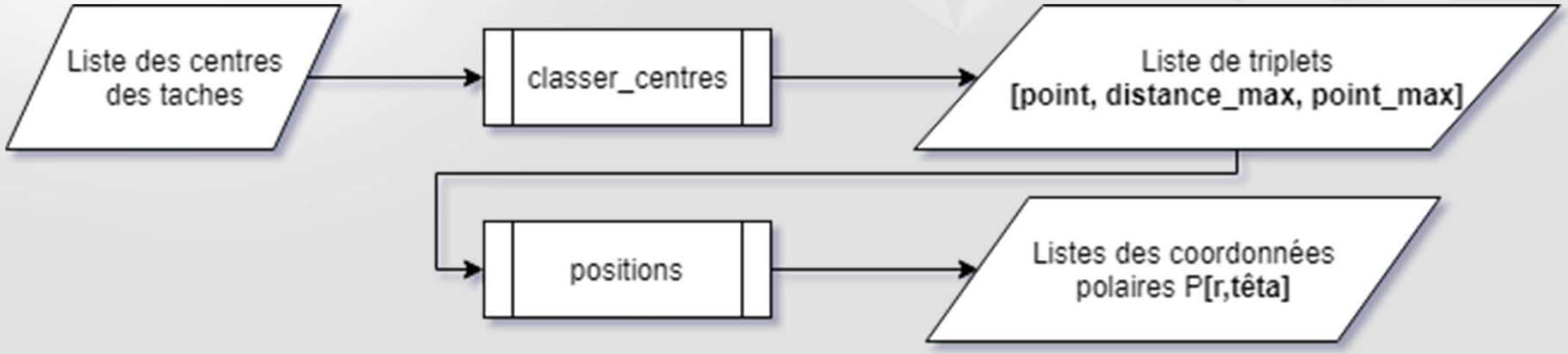
$$r=\frac{d}{d_{ref}}$$

$$C(r;\theta)$$



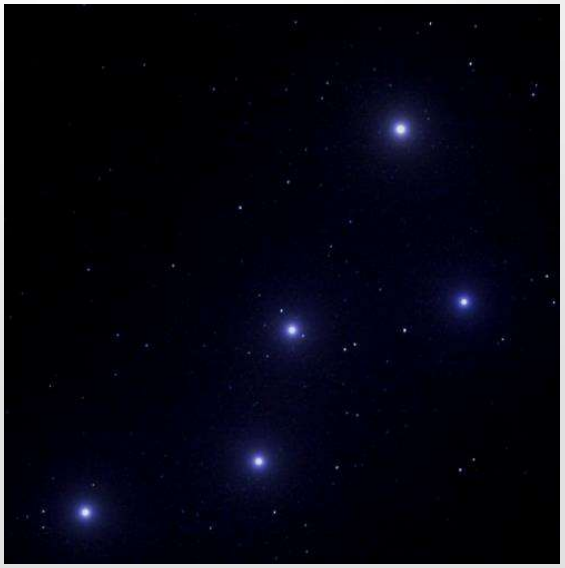
Objectif 1 : Définir

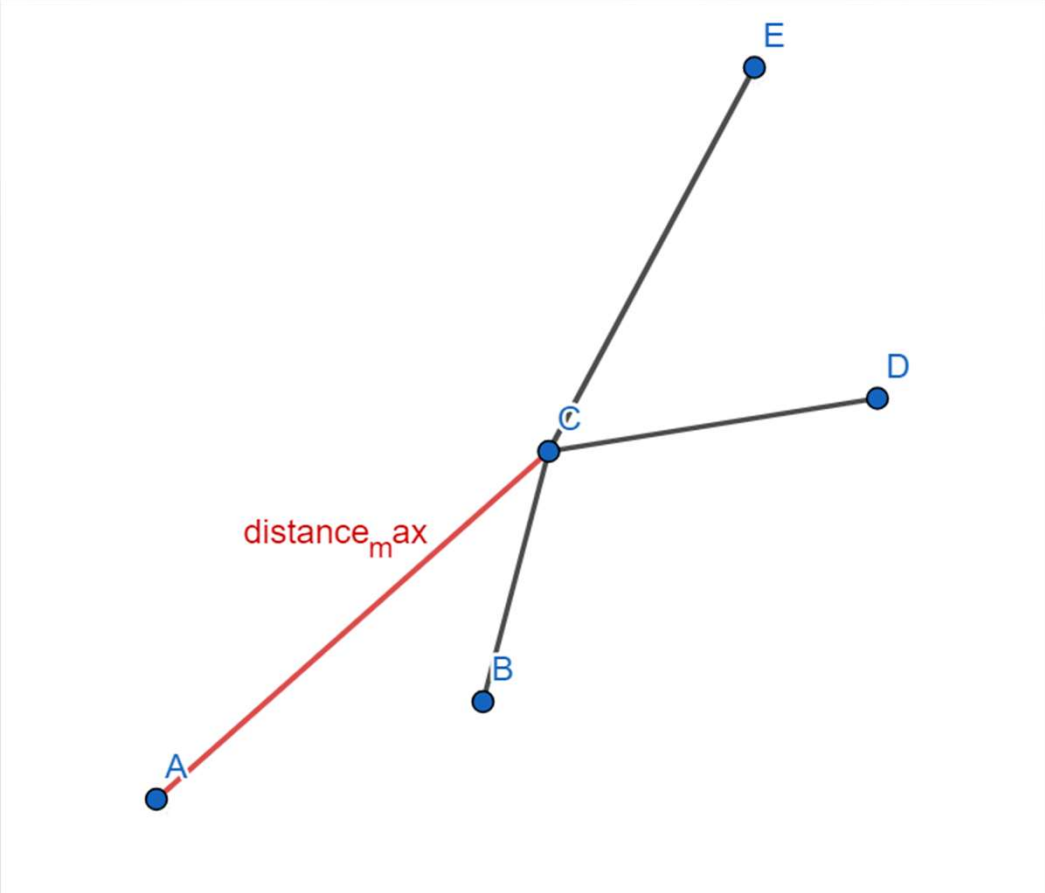
Repérer les points dans une base propre à la constellation



Objectif 1 : Définir

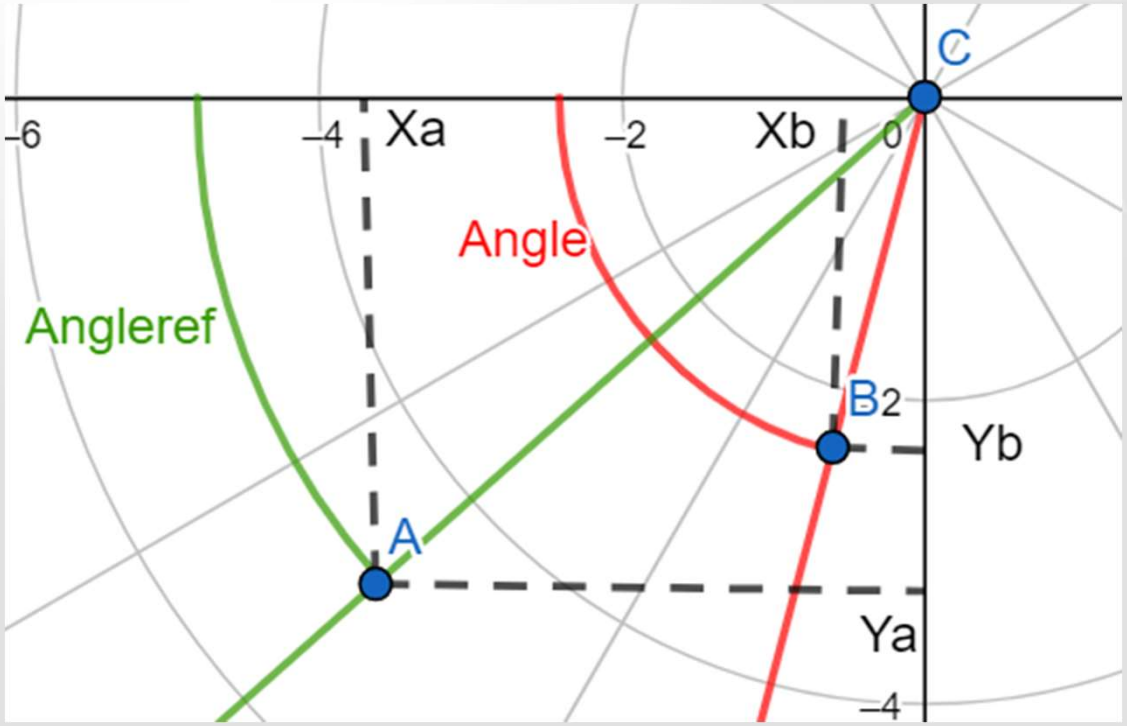
Repérer les points dans une base propre à la constellation





Objectif 1 : Définir

Repérer les points dans une base propre à la constellation



$$d_{ref} = \sqrt{(Xa - Xc)^2 + (Ya - Yc)^2}$$

$$d = \sqrt{(Xb - Xc)^2 + (Ya - Yc)^2}$$

Angle=atan2(Yb,Xb)

Angleref=atan2(Ya,Yb)

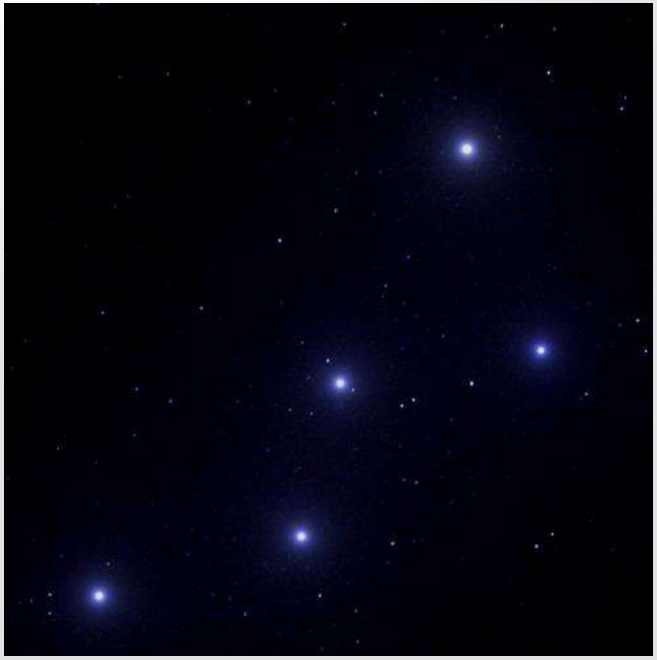
- Si Angle≥ *Angleref :*

  θ= Angle-Angleref

- Si Angle≤ *Angleref*:

  θ= 2π+Angle-Angleref

Résultat



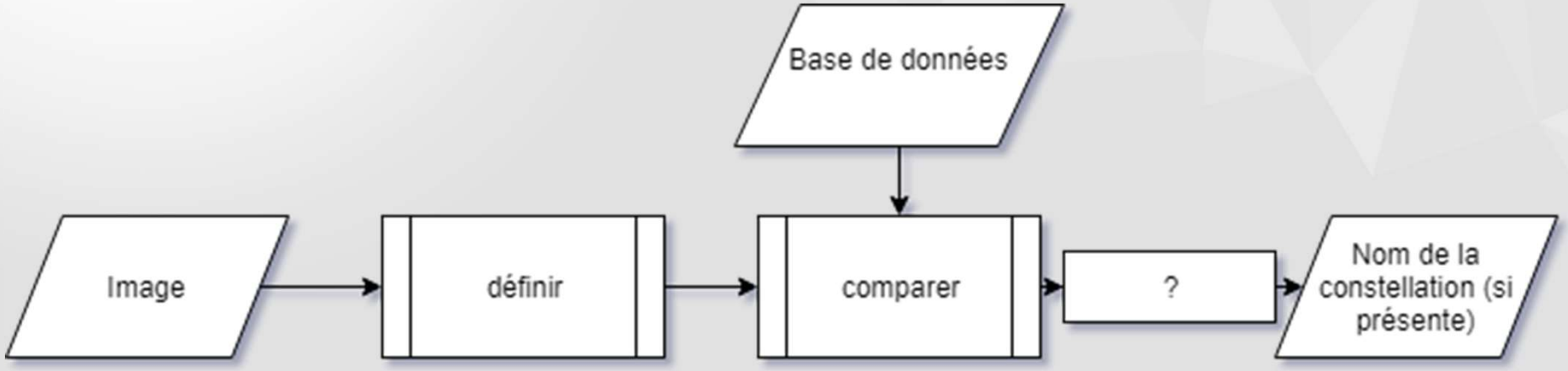
4 doublets (r,Θ)

```
[[ 0.82890784  3.49434109]
 [ 0.63329236  2.58135119]
 [ 0.49123224  0.60032042]
 [ 1.          0.        ]]
```

Objectif 2 :

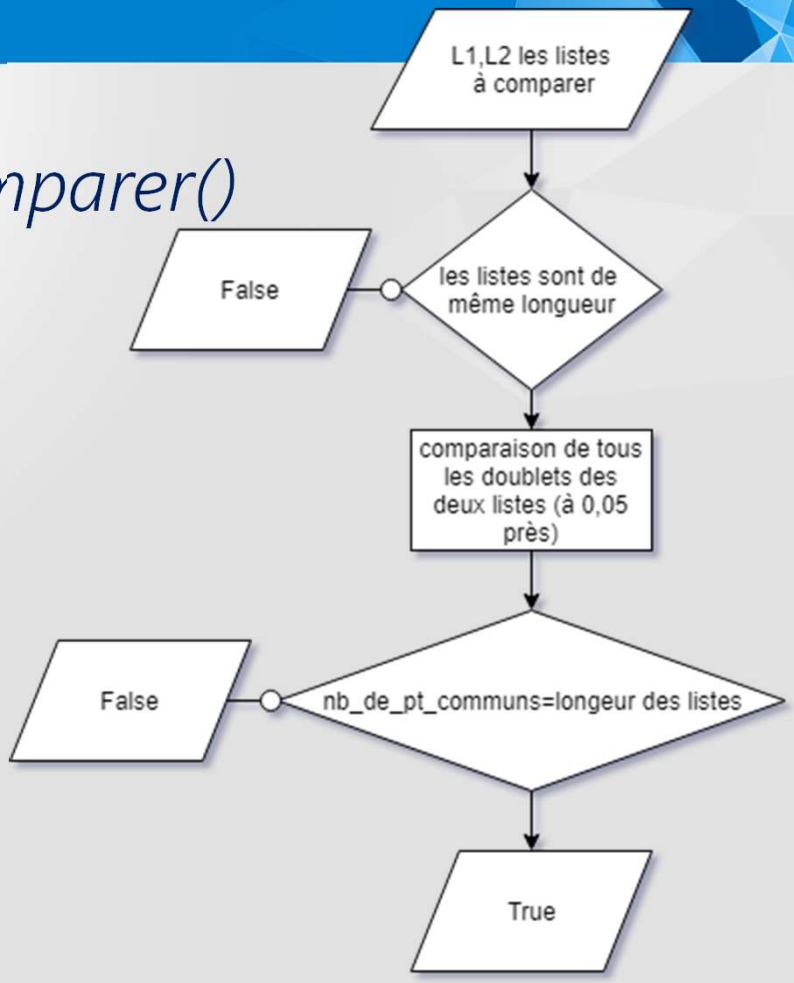
Identifier une constellation

Objectif 2 : Identifier

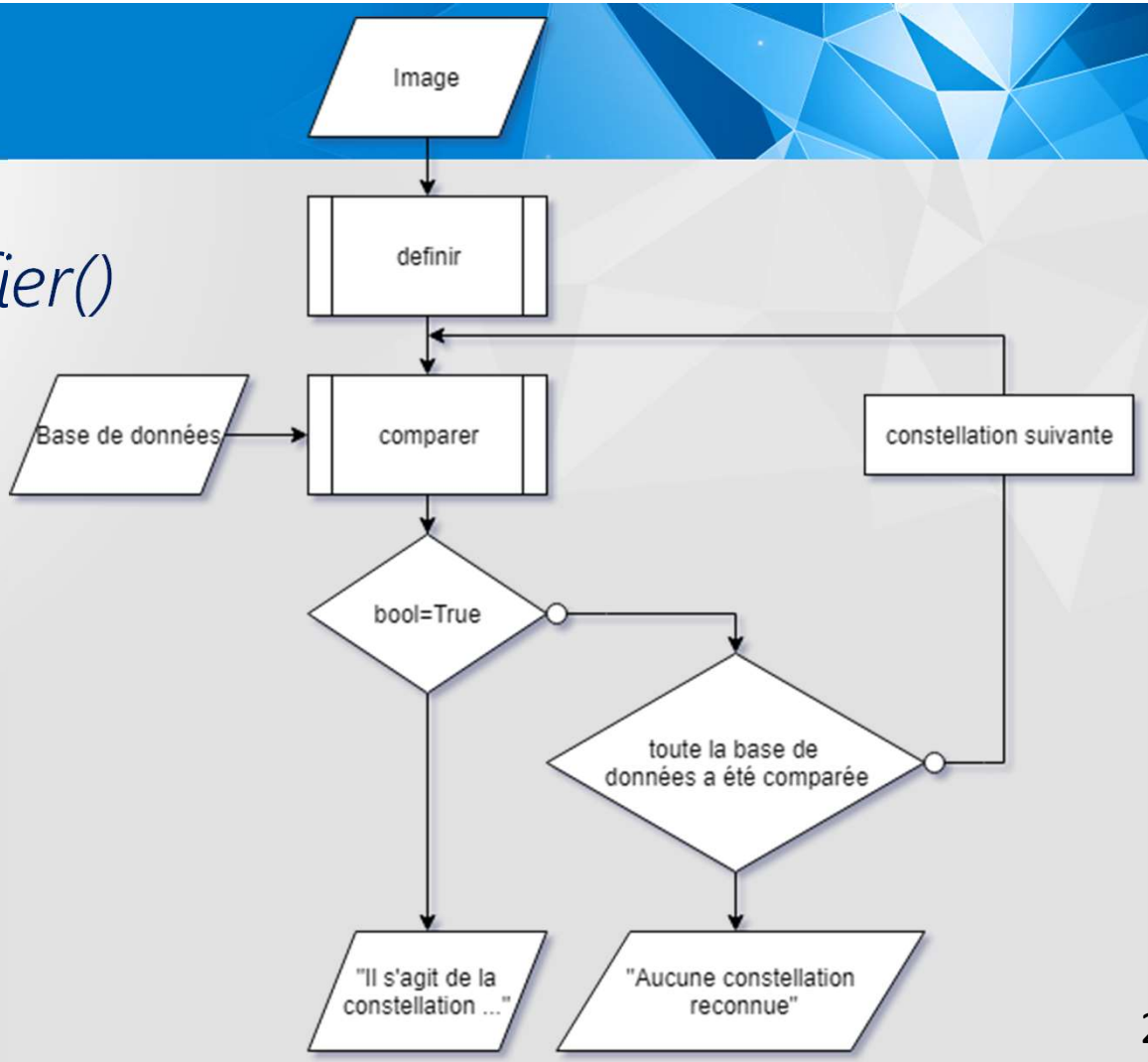


La fonction *comparer()*

- Renvoie un **booléen**



La fonction *identifier()*



3 expériences

Avec Cassiopée et la Grande Ourse dans la base

- Deux images de Cassiopée et de la Grande Ourse
- Une photographie

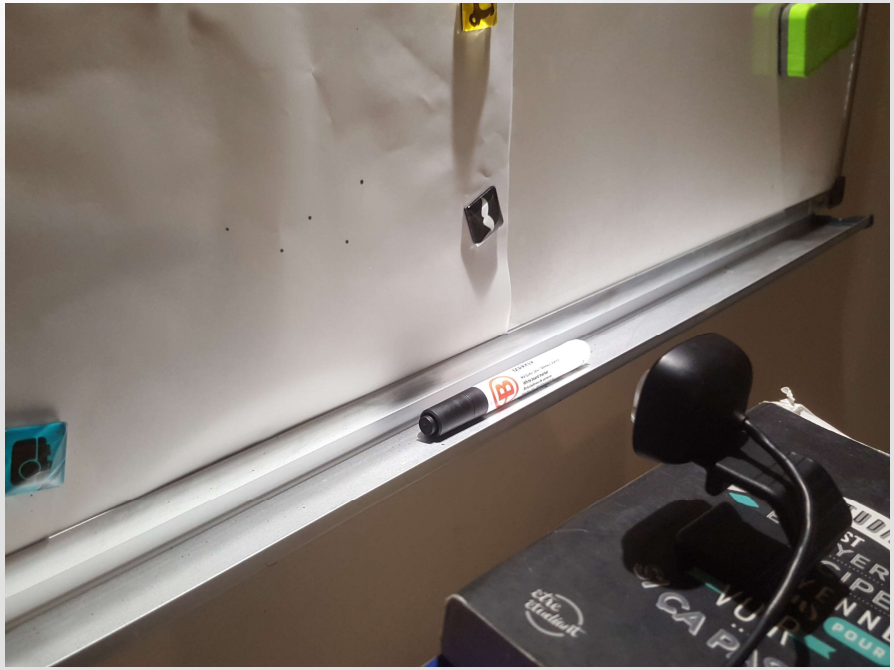
Cassiopée

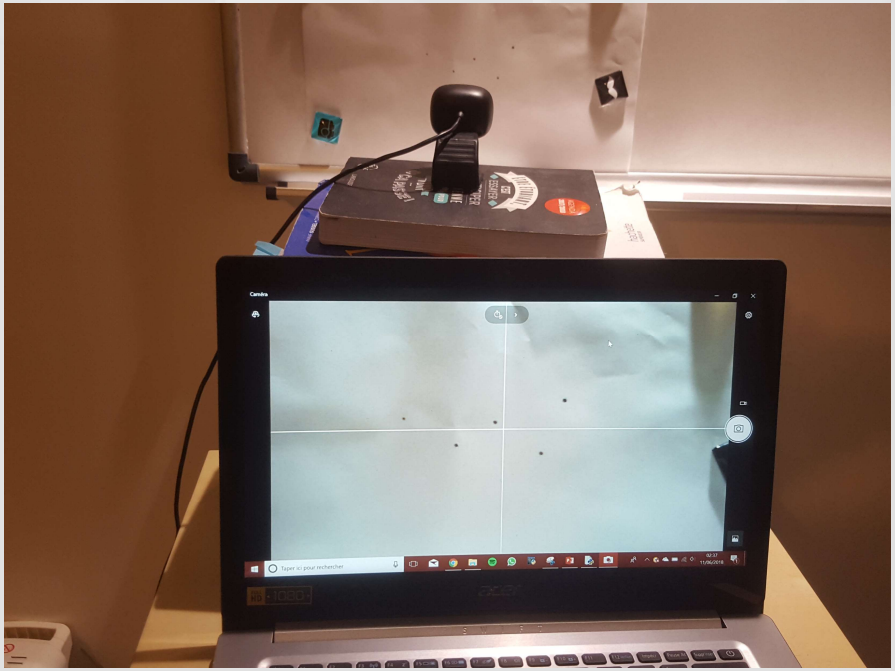
```
>>> identifie(im)
"Il s'agit de la constellation Cassiopée"
```

La grande Ourse

```
>>> identifie(im)
"Il s'agit de la constellation de la Grande Ourse"
```

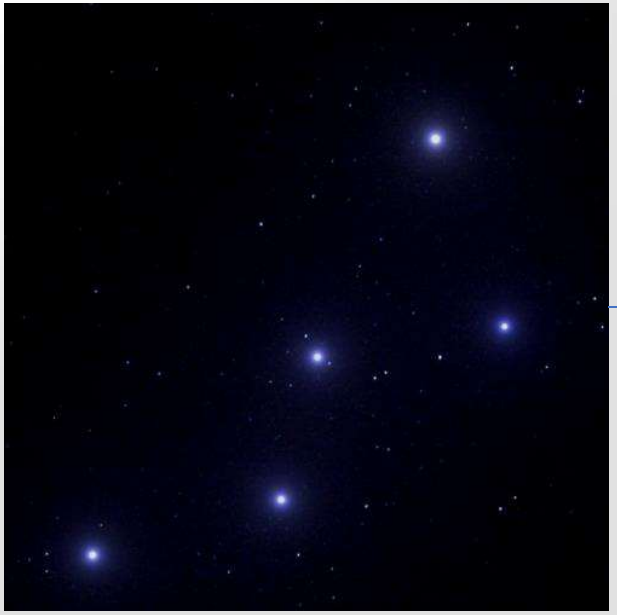
Photographie





Objectif 2 : Identifier

Photographie



```
>>> identifie(im)
"Il s'agit de la constellation Cassiopée"
```

POUR ALLER PLUS LOIN …