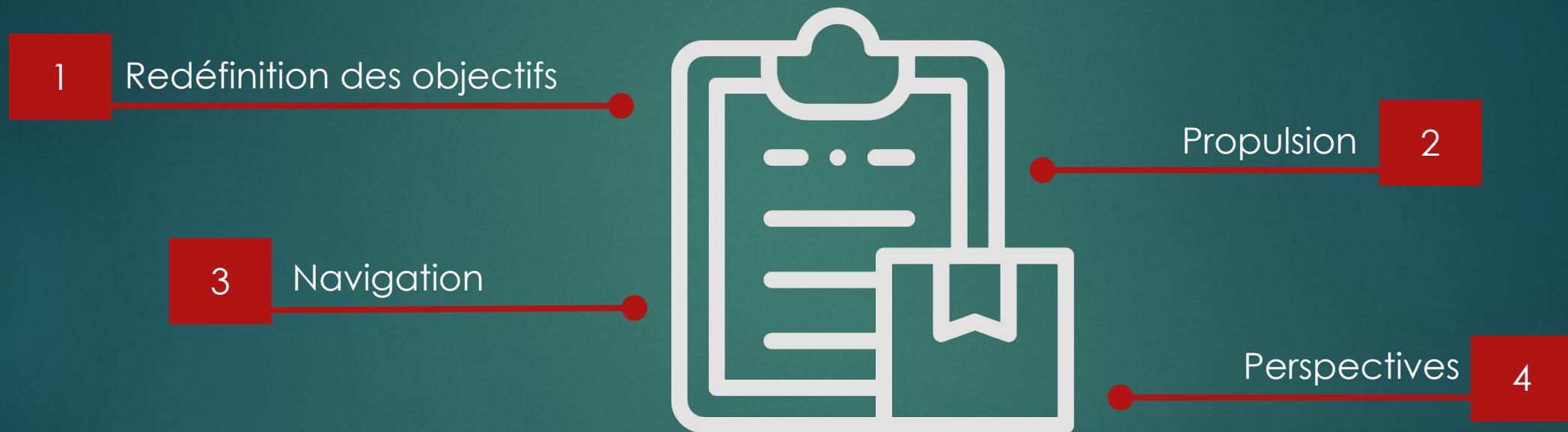


P13 Robot à système de propulsion déformable

SOUTENANCE DE PROJET | SEMESTRE 8 | DAMIEN TILLAUX & VINCENT DUBOIS

Sommaire

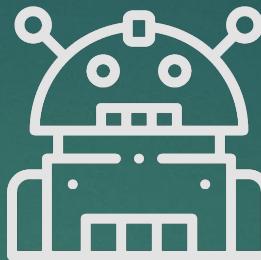
2



Objectifs

Propulsion

- Electronique embarquée (Shield)
- Dimensionnement des nageoires:
 - Matériaux
 - Epaisseur
 - Forme
 - etc ...
- Algorithme d'ondulation



Navigation

- Acquisition des données
 - Caméras
 - LIDAR
- Reconnaissance du milieu:
 - Type d'environnement
 - Position des obstacles
 - Position de la cible
- Calcul de trajectoire

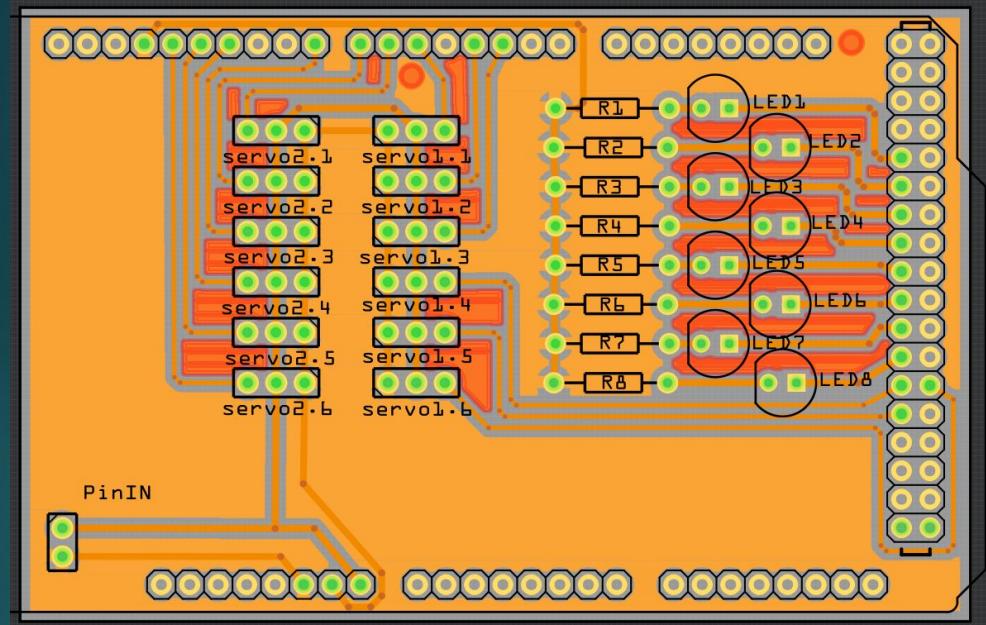
Communication
(Liaison série)

2. Propulsion

Damien

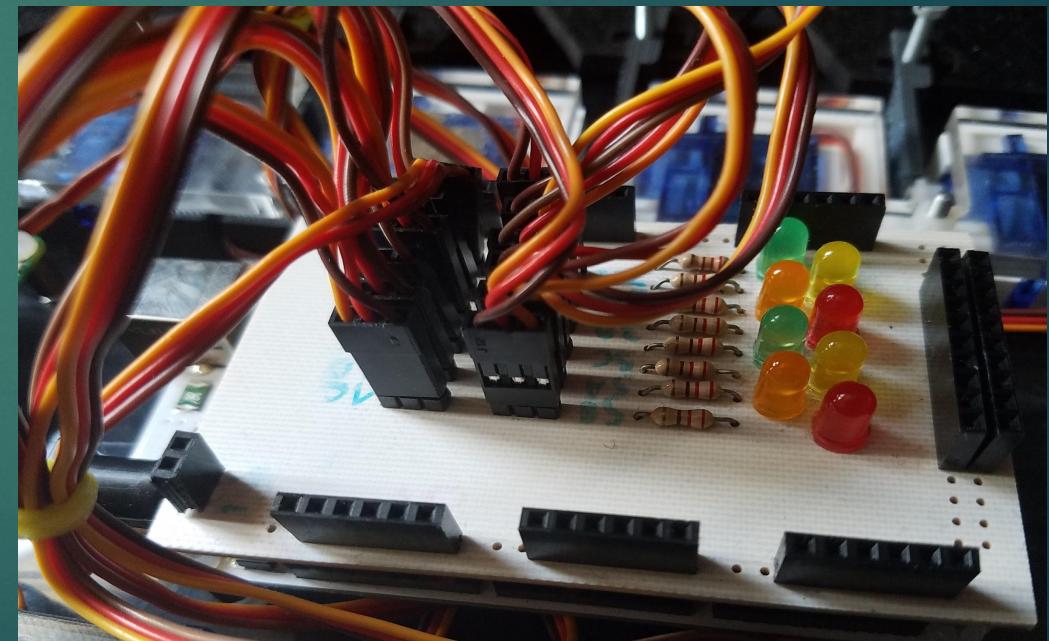


Shield Arduino



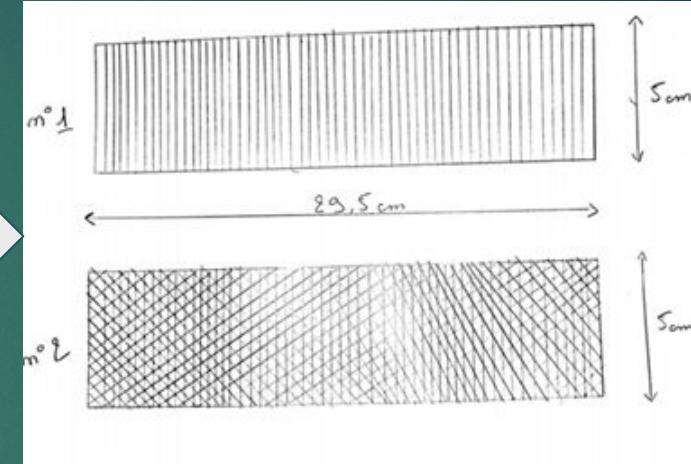
Schématique du shield

Shield branché



Conception des pales

Prototype en lino

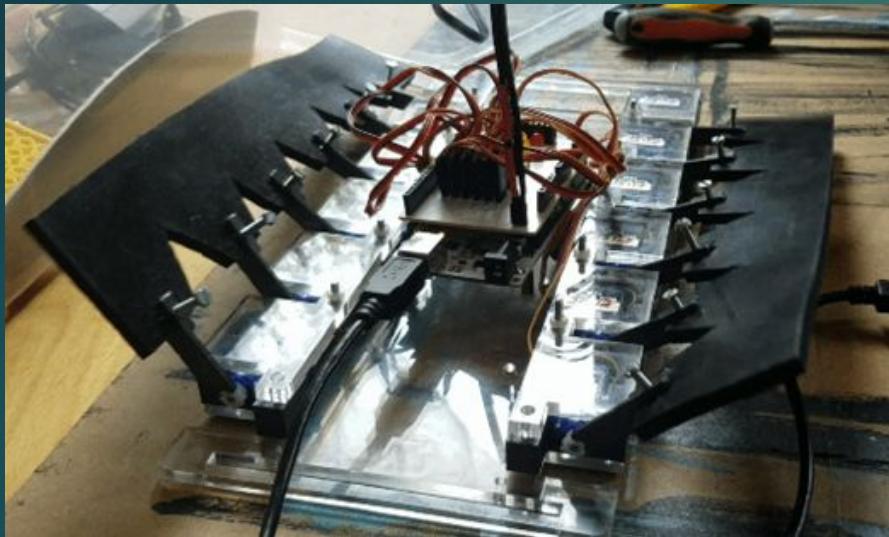


Prototype final



Prototype 1 en caoutchouc

Algorithme de propulsion



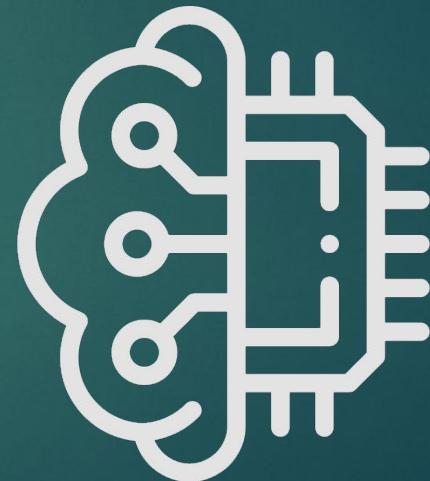
Exécution de l'algorithme

<https://github.com/vincentduboisdlc/P13.git>

```
void UneOndulation(void){  
    if(OndulationisFinish == false){  
        if(inclinaison_servo[0] >= inclinaison_init_droite + inclinaison_max || inclinaison_servo[0] <= inclinaison_init_droite - inclinaison_max)  
        {  
            sens_rotation[0] = !sens_rotation[0];  
        }  
        if(sens_rotation[0])  
        {  
            inclinaison_servo[0] = inclinaison_servo[0] + pas;  
            inclinaison_servo[6] = inclinaison_servo[6] - pas;  
        }  
        else  
        {  
            inclinaison_servo[0] = inclinaison_servo[0] - pas;  
            inclinaison_servo[6] = inclinaison_servo[6] + pas;  
        }  
        if(inclinaison_servo[0] == inclinaison_init_droite && sens_rotation[0] == true){  
            isFinish = true;  
        }  
    }  
    else{  
        if(wait != nbWait){  
            wait ++;  
        }  
        else{  
            OndulationisFinish= false;  
            wait = 0;  
        }  
        for(int i = nbAncien-1; i > 0; i--){  
            ancienne_incliDroite[i] = ancienne_incliDroite[i-1];  
            ancienne_incliGauche[i] = ancienne_incliGauche[i-1];  
        }  
        ancienne_incliDroite[0] = inclinaison_servo[0];  
        ancienne_incliGauche[0] = inclinaison_servo[6];  
        for(int i = 5; i > 0; i-){  
            inclinaison_servo[i] = ancienne_incliDroite[(i+1)*nbAncien/6-1];  
            inclinaison_servo[i+6] = ancienne_incliGauche[(i+1)*nbAncien/6-1];  
        }  
    }  
}
```

3. Navigation

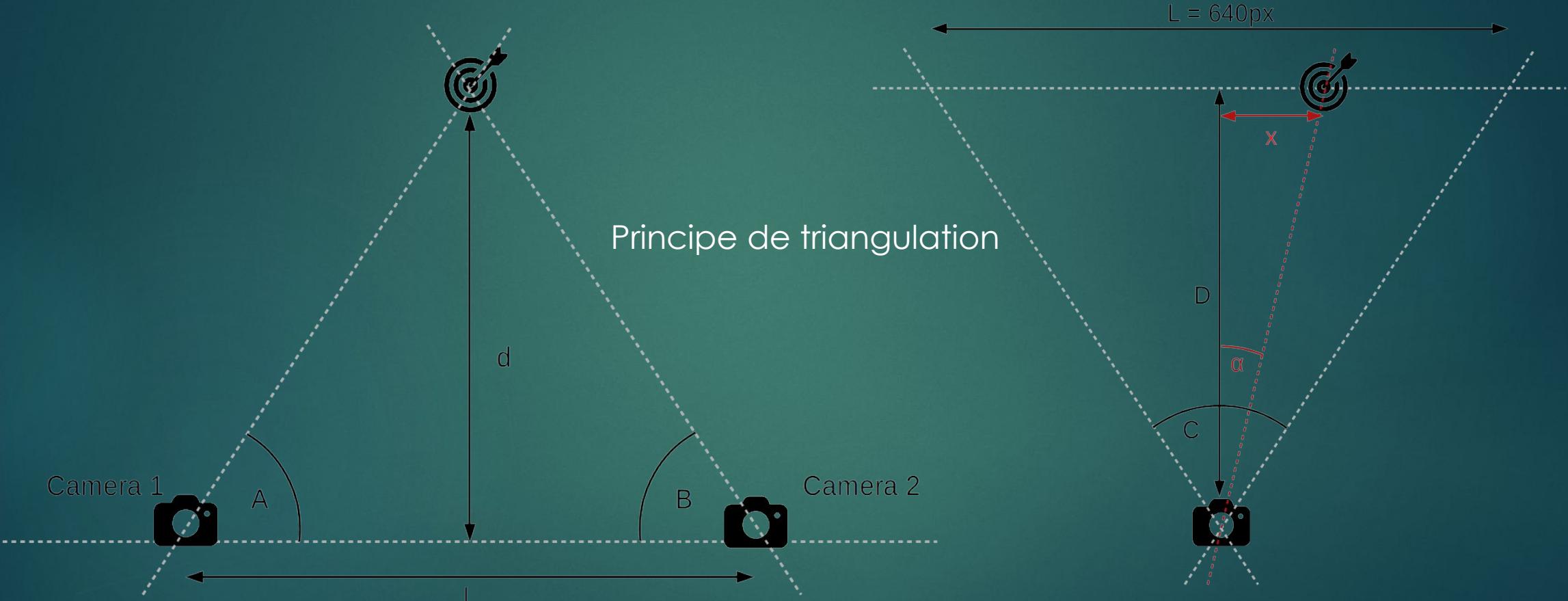
Vincent



Stéréovision – idée générale



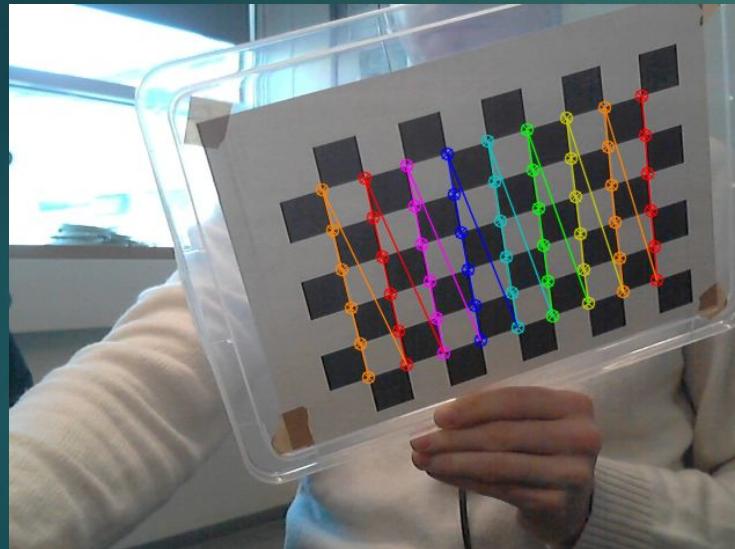
9



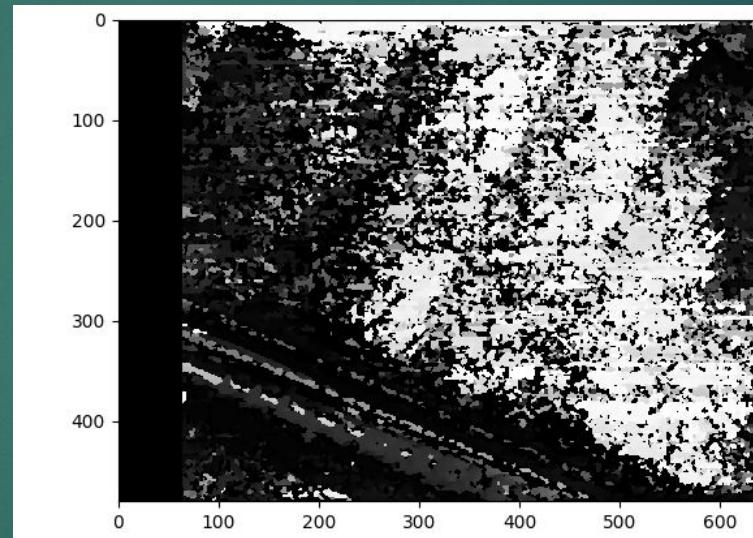
Trouver un point similaire sur chaque image : SURF (Speeded Up Robust Features)

Stéréovision – calibration

10



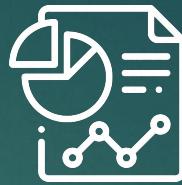
Calibration des caméras pour compenser la distorsion



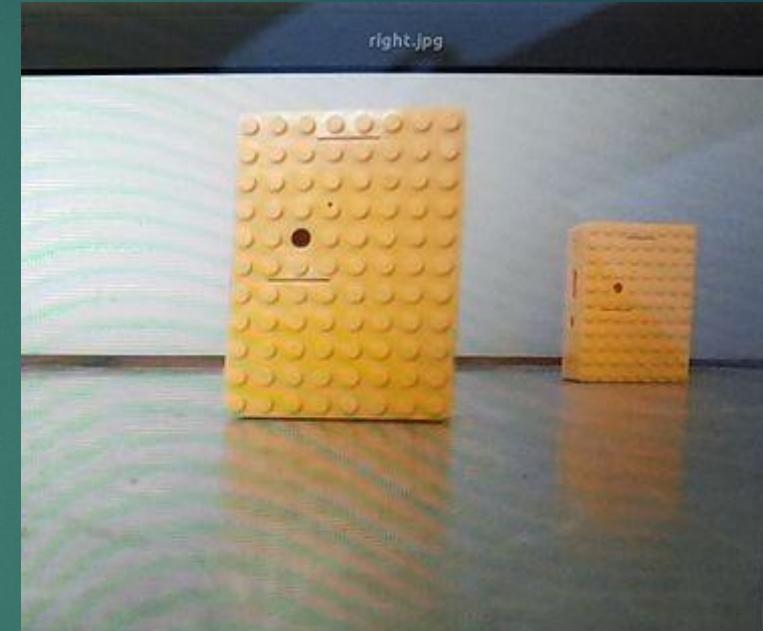
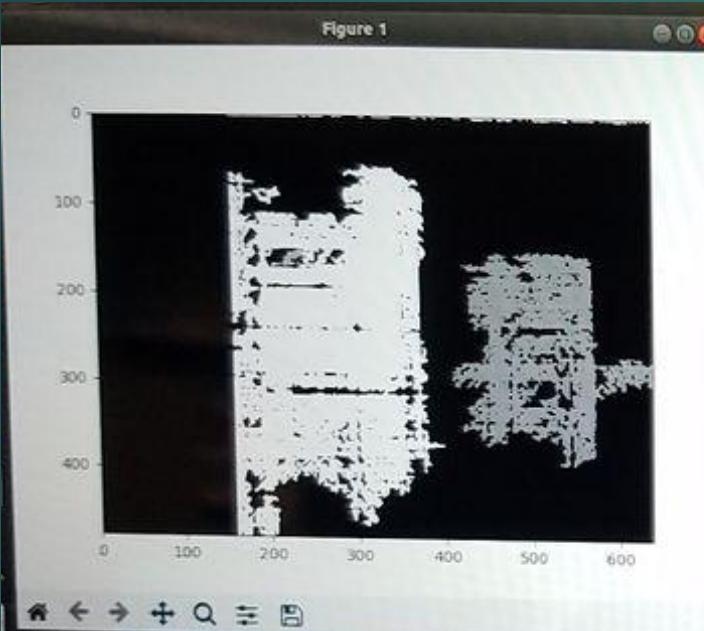
Etalonnage du « Matcher » pour obtenir de meilleurs résultats



Stéréovision – résultats



11

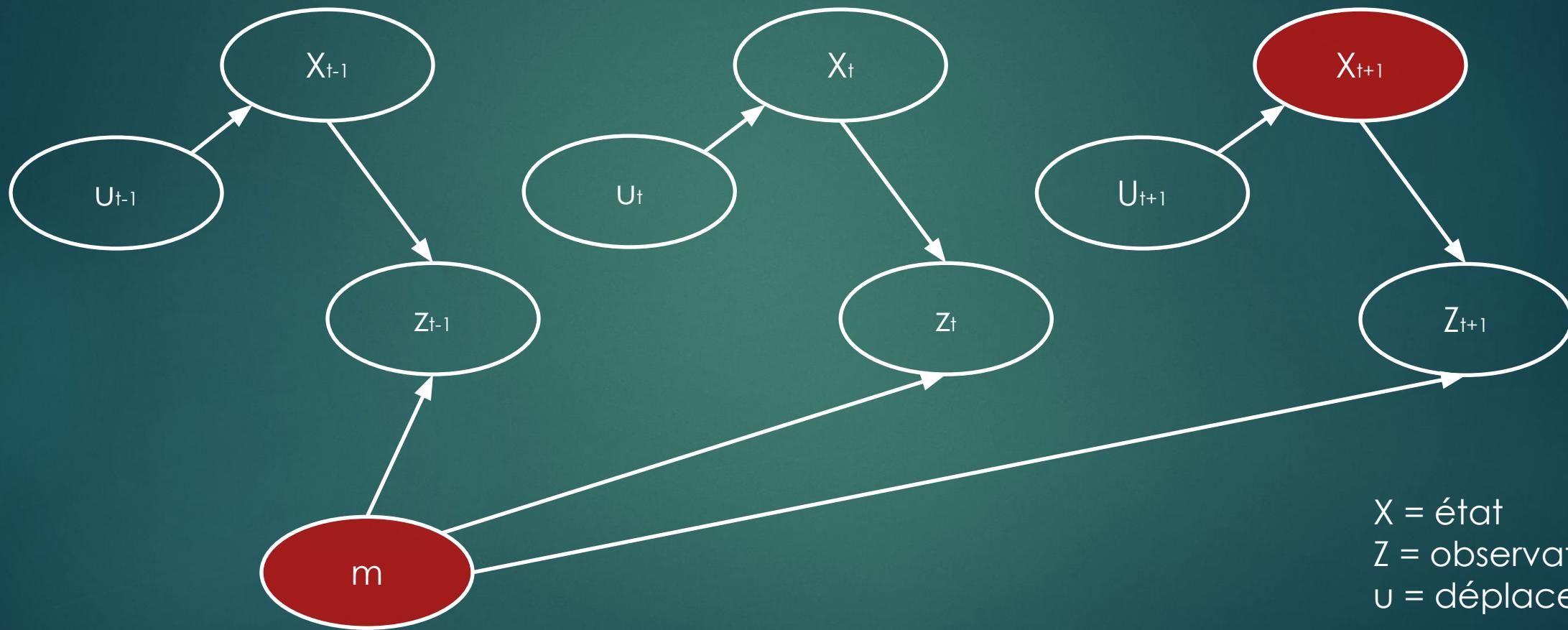


Problème? Inclinaison des caméra

Mapping - idée générale



12



X = état
Z = observation
u = déplacement
m = map

SLAM = Simultaneous Localisation And Mapping

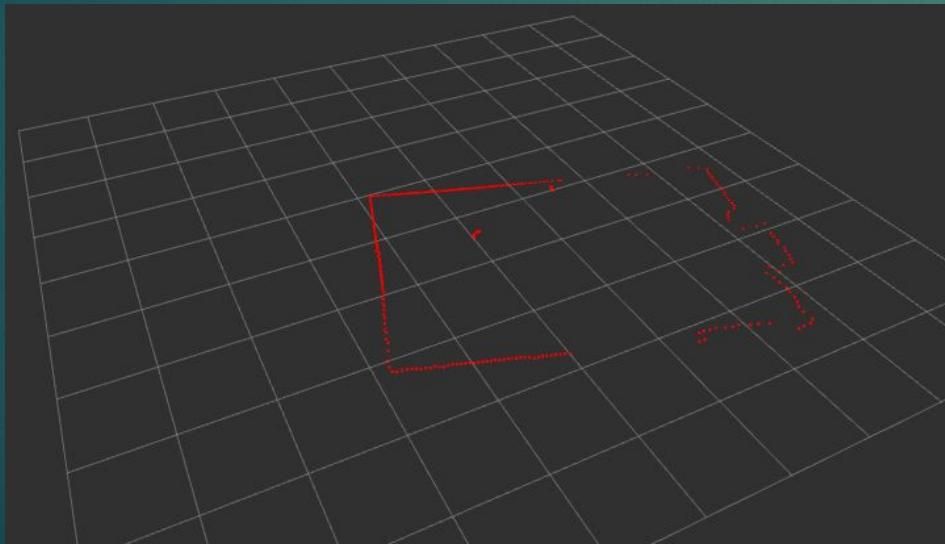
Mapping – résultats

13



Contrainte:
Pas d'odométrie

- ROS (Google cartographer)
- PythonRobotics
- BreezSLAM (Simon D. Levy)



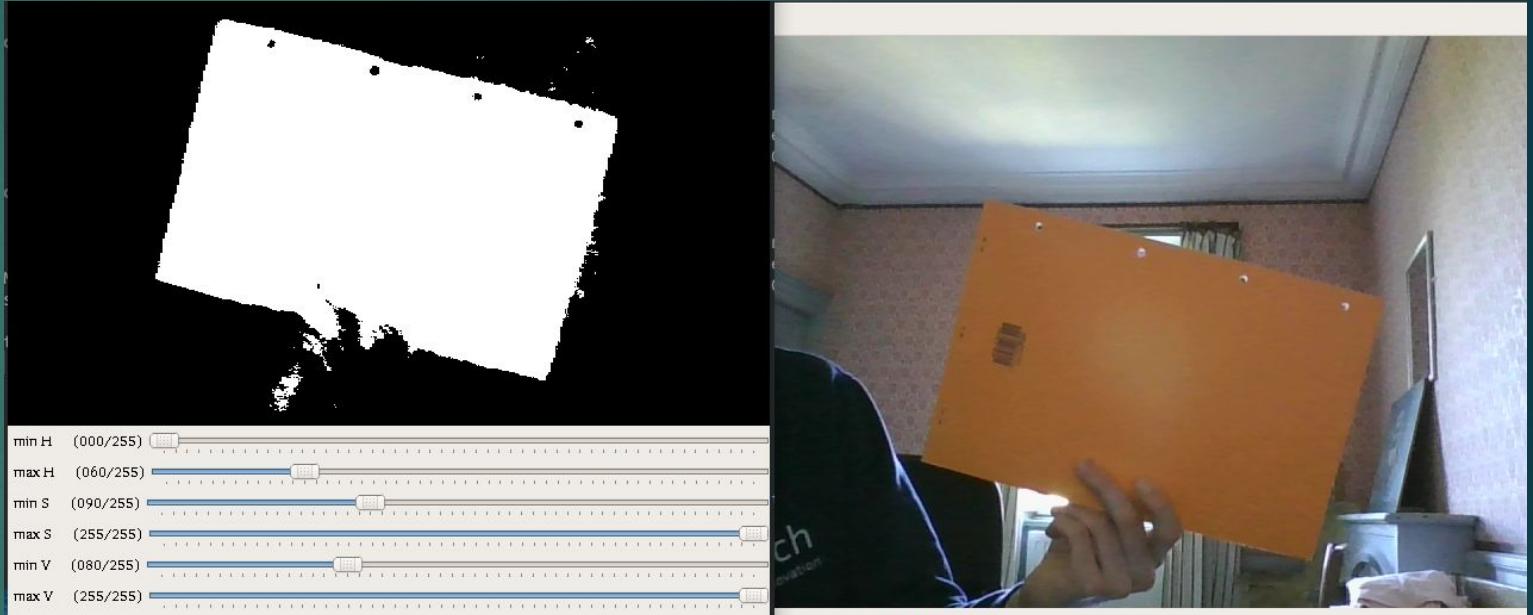
Détection cible



14

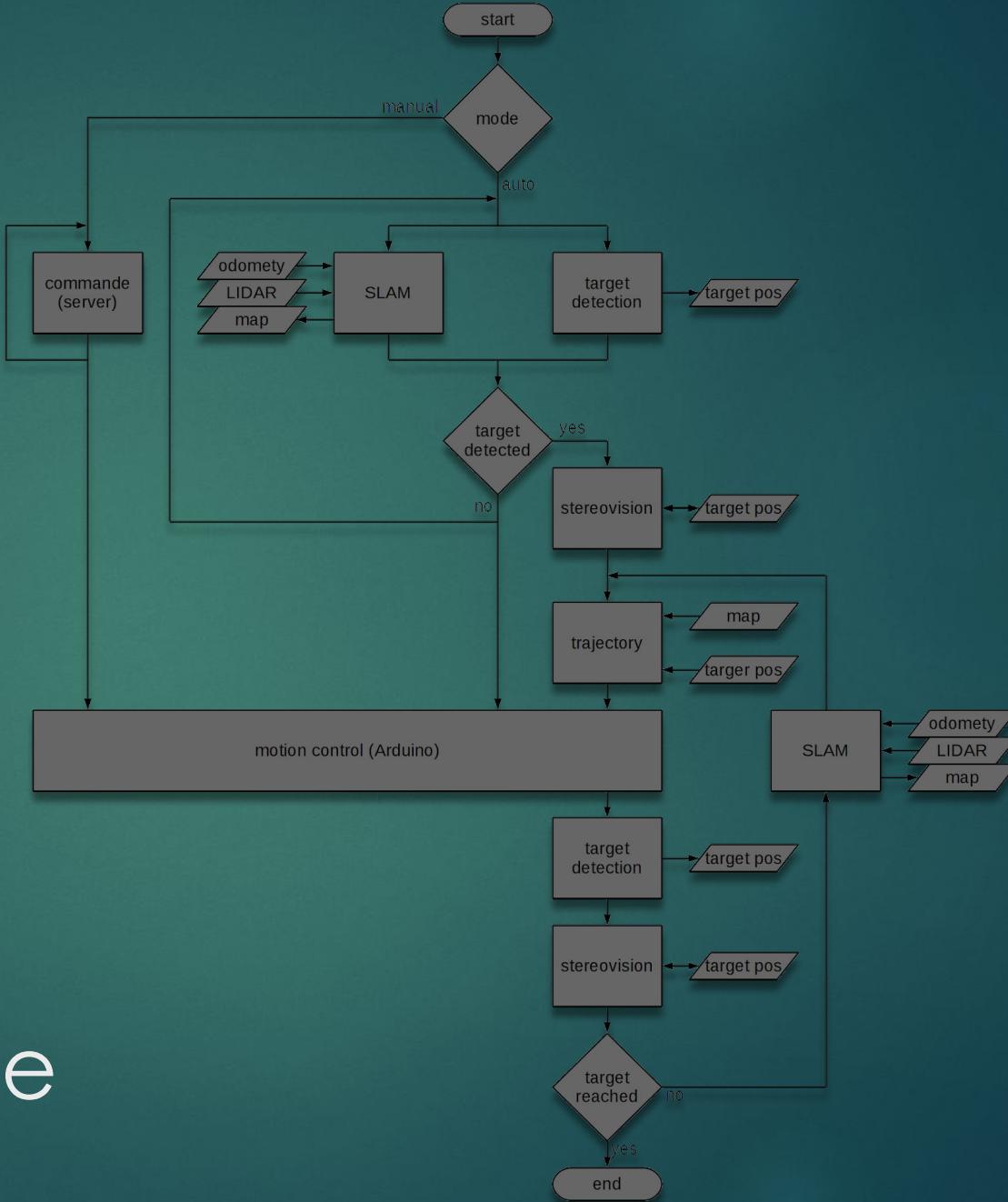
✗ Filtre de couleur

◻ Algo. Machine Learning



Map + Position cible = Trajectoire (a^*)

Architecture globale



Perspectives

16



Limites

Ressource

Projet multidisciplinaire

Servomoteurs sous dimensionnés

Pas d'odométrie sur le robot



Améliorations

Utilisation de servomoteurs plus puissance ou des moteurs

Mise en place total de l'architecture global

Détection de cible par Machine learning

Matériels pour la stéréovision