

ECOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

[polytech.univ-tours.fr](http://polytech.univ-tours.fr)

## Projet Recherche & Développement

2017-2018

# Projet Recherche et Développement polytech

Implémentation du algorithme de suivi

**POLYTECH<sup>®</sup>**  
TOURS

**Tuteurs académiques**

Quoccuong LE

Donatello CONTE

Pierre GAUCHER

Moncef HIDANE

**Étudiant**

Rujia WANG (DI5)



# Liste des intervenants

| Nom             | Email  | Qualité  |
|-----------------|--|--|
| Rujia WANG      | <a href="mailto:rujia.wang@etu.univ-tours.fr">rujia.wang@etu.univ-tours.fr</a>     | Étudiant DI5                                   |
| Quoccuong LE    | <a href="mailto:quoccuong.le@etu.univ-tours.fr">quoccuong.le@etu.univ-tours.fr</a> | Tuteur académique,<br>Département Informatique |
| Donatello CONTE | <a href="mailto:donatello.conte@univ-tours.fr">donatello.conte@univ-tours.fr</a>   | Tuteur académique,<br>Département Informatique |
| Pierre GAUCHER  | <a href="mailto:pierre.gaucher@univ-tours.fr">pierre.gaucher@univ-tours.fr</a>     | Tuteur académique,<br>Département Informatique |
| Moncef HIDANE   | <a href="mailto:moncef.hidane@univ-tours.fr">moncef.hidane@univ-tours.fr</a>       | Tuteur académique,<br>Département Informatique |



# Avertissement

Ce document a été rédigé par Rujia Wang susnommé l'auteur.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par Quoccuong Le, Donatello Conte, Pierre Gaucher et Moncef Hidane susnommés les tuteurs académiques.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable des tuteurs académiques et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



## Pour citer ce document

Rujia Wang, *Projet Recherche et Développement polytech: Implémentation du algorithme de suivi*,  
Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais  
de Tours, Tours, France, 2017-2018.

```
@mastersthesis{
  author={Wang, Rujia},
  title={Projet Recherche et Développement polytech: Implémentation du algorithme de
    suivi},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2017-2018}
}
```



# Table des matières

|                                  |      |
|----------------------------------|------|
| Liste des intervenants           | a    |
| Avertissement                    | b    |
| Pour citer ce document           | c    |
| Table des matières               | i    |
| Table des figures                | v    |
| Liste des tableaux               | vii  |
| Liste des Algorithmes            | viii |
| Introduction                     | 1    |
| 1 L'acteur du projet est : ..... | 1    |
| 2 Objectifs .....                | 1    |
| 3 Bases méthodologiques .....    | 1    |
| Description générale             | 3    |
| 1 Environnement du projet .....  | 3    |
| 1.1 Raspberry Pi.....            | 3    |
| 1.2 RASPBIAN .....               | 3    |
| 1.3 OpenCV .....                 | 3    |
| 1.4 Python .....                 | 4    |
| 1.5 NumPy .....                  | 4    |
| 1.6 SciPy.....                   | 4    |
| 1.7 Matplotlib .....             | 4    |

|  |   |           |
|--|---|-----------|
| 2  | Caractéristiques des utilisateurs .....   | 4         |
| 3  | Fonctionnalités du système .....  | 5         |
| 4  | Structure générale du système .....   | 7         |
| <b>Etat de l'art/veille</b>                              |   | <b>9</b>  |
| 1  | L'algorithme existant .....   | 9         |
| 1.1  | Représentations parcimonieuses .....  | 9         |
| 1.2  | Méthodes Monte Carlo Séquentielles.....   | 10        |
| 1.3  | Filtre particulaire et $l_1$ -minimization.....   | 10        |
| 1.4  | Suivi collaboratif avec plusieurs caméras utilisant des représentations<br>parcimonieuses ..... | 12        |
| 1.5  | L'algorithme de suivi .....   | 12        |
| 2  | Code Matlab.....  | 13        |
| 3  | Algorithme de suivi avec une seule caméra.....  | 14        |
| <b>Analyse et conception</b>                             |   | <b>15</b> |
| <b>Mise en oeuvre</b>                                    |   | <b>17</b> |
| 1  | Les outils et bibliothèques utilisées.....  | 17        |
| 2  | Achèvement .....  | 17        |
| 3  | Mise en oeuvre .....  | 17        |
| 4  | Limites .....   | 18        |
| 5  | Développement futur .....   | 18        |
| 6  | Résultats.....  | 19        |
| <b>Bilan et conclusion</b>                               |   | <b>25</b> |
| 1  | Planning du s10 .....   | 25        |
| 2  | Redards .....   | 26        |
| <b>Annexes</b>   |   | <b>27</b> |
| <b>A Glossaire</b>                                       |   | <b>28</b> |
| <b>B Description des interfaces externes du logiciel</b> |   | <b>29</b> |
| 1  | Interfaces matériel/logiciel .....  | 29        |
| 2  | Interfaces homme/machine .....  | 30        |
| 3  | Interfaces logiciel/logiciel .....  | 30        |
| <b>C Spécifications fonctionnelles</b>                   |   | <b>31</b> |
| 1  | Définition de la fonction Convertir de coordonnées 2D en 3D.....                                | 31        |
| 2  | Définition de la fonction Convertir de coordonnées 3D en 2D.....                                | 31        |

|          |   |           |
|----------|---|-----------|
| 3        | Définition de la fonction Convertir des images déformées en des images non déformées..... | 31        |
| 4        | Définition de la fonction Convertir des images non déformées en des images déformées..... | 32        |
| 5        | Définition de la fonction Interpolation de patch d'image.....                             | 32        |
| 6        | Définition de la fonction Ré-échantillonnage et génération de particules .....            | 32        |
| 7        | Définition de la fonction Calculer des coefficients de modèles .....                      | 33        |
| 8        | Définition de la fonction Codage parcimonie.....  | 33        |
| 9        | Définition de la fonction Calculer l'erreur de reconstruction.....                        | 33        |
| 10       | Définition de la fonction Calculer le niveau de sparsity .....                            | 33        |
| 11       | Définition de la fonction Construire le dictionnaire .....                                | 33        |
| 12       | Définition de la fonction Mettre à jour le dictionnaire .....                             | 34        |
| 13       | Définition de la fonction Obtenir les frames .....  | 34        |
| 14       | Définition de la fonction Sortie .....  | 34        |
| <b>D</b> | <b>Spécifications non fonctionnelles</b>  | <b>35</b> |
| 1        | Contraintes de développement et conception .....  | 35        |
| 2        | Contraintes de fonctionnement et d'exploitation .....                                     | 35        |
| 2.1      | Performances.....   | 35        |
| 2.2      | Capacités.....  | 35        |
| 2.3      | Sécurité .....  | 35        |
| <b>E</b> | <b>Cahier du developpeur</b>  | <b>36</b> |
| 1        | Intialisation.....  | 36        |
| 2        | Suivi de l'objet .....  | 36        |
| 3        | Changer la caméra pour suivre l'objet .....   | 36        |
| 4        | Sortir le résultat .....  | 37        |
| <b>F</b> | <b>Gestion de projet</b>  | <b>38</b> |
| 1        | Gestion de projet .....   | 38        |
| 2        | Planification S9.....   | 38        |
| <b>G</b> | <b>Doc d'installation et d'utilisation</b>  | <b>40</b> |
| 1        | installation d'image de système d'exploitation .....                                      | 40        |
| 2        | installation d'OpenCV .....   | 40        |
| 3        | Configurer le chemin de fichier .....   | 41        |
| <b>H</b> | <b>Cahier de test</b>   | <b>42</b> |
| 1        | Unit Test.....  | 42        |
| 1.1      | test_affparam2mat.....  | 42        |
| 1.2      | test_affparam2geom .....  | 43        |

|      |  |           |
|------|--|-----------|
| 1.3  | test_warping .....                           | 43        |
| 1.4  | test_distortedToUndistortedSensorCoord ..... | 44        |
| 1.5  | test_undistortedToDistortedSensorCoord ..... | 44        |
| 1.6  | test_imageToWorld .....                      | 45        |
| 1.7  | test_worldToImage .....                      | 46        |
| 1.8  | test_decompFista .....                       | 46        |
| 1.9  | test_classifyOutsideParticlePETS.....        | 47        |
| 1.10 | test_extractCandidates .....                 | 47        |
| 1.11 | test_fifo_stack_v2 .....                     | 48        |
| 1.12 | test_szPrePETS .....                         | 49        |
| 1.13 | test_bhattacharyya.....                      | 49        |
| 1.14 | test_motionCheck .....                       | 50        |
| 1.15 | test_updateDictionary .....                  | 50        |
| 1.16 | test_cam2camGroundTrackPETS.....             | 51        |
| 2    | Test Fonctionnel .....                       | 52        |
| 2.1  | Le résultat des coins de frame 1 .....       | 52        |
| 2.2  | Le résultat des coins de frame 2 .....       | 52        |
|      | <b>Webographie</b> .....                     | <b>53</b> |
|      | <b>Bibliographie</b> .....                   | <b>54</b> |

# Table des figures

## Description générale

|   |                                      |   |
|---|--------------------------------------|---|
| 1 | Diagramme de cas d'utilisation ..... | 5 |
| 2 | Diagramme d'activité .....           | 6 |
| 3 | Structure générale du système .....  | 7 |
| 4 | Diagramme de séquence .....          | 8 |

## Etat de l'art/veille

|   |  |    |
|---|--|----|
| 5 | Représentations parcimonieuses .....             | 10 |
| 6 | Bootstrap filtre[1] .....                        | 11 |
| 7 | Image déformée et image non déformée[WWW1] ..... | 13 |

## Analyse et conception

|   |                           |    |
|---|---------------------------|----|
| 8 | Diagramme de classe ..... | 15 |
|---|---------------------------|----|

## Mise en oeuvre

|    |                                  |    |
|----|----------------------------------|----|
| 9  | Le résultat du fichier txt ..... | 19 |
| 10 | Frame 1 .....                    | 20 |
| 11 | Frame 2 .....                    | 20 |
| 12 | Frame 3 .....                    | 21 |
| 13 | Frame 4 .....                    | 21 |
| 14 | Frame 5 .....                    | 22 |
| 15 | Frame 6 .....                    | 22 |
| 16 | Frame 7 .....                    | 23 |
| 17 | Frame 8 .....                    | 23 |
| 18 | Frame 9 .....                    | 24 |
| 19 | Frame 10 .....                   | 24 |

**Bilan et conclusion**

|    |                                   |    |
|----|-----------------------------------|----|
| 20 | Les tâches de S10 .....           | 25 |
| 21 | Le diagramme de gantt de S10..... | 26 |

**B Description des interfaces externes du logiciel**

|   |                                    |    |
|---|------------------------------------|----|
| 1 | Interfaces matériel/logiciel ..... | 29 |
|---|------------------------------------|----|

**C Spécifications fonctionnelles**

|   |                                |    |
|---|--------------------------------|----|
| 1 | Distortion d'image[WWW1] ..... | 32 |
|---|--------------------------------|----|

**E Cahier du developpeur**

|   |                           |    |
|---|---------------------------|----|
| 1 | Diagramme de classe ..... | 37 |
|---|---------------------------|----|

**F Gestion de projet**

|   |                                  |    |
|---|----------------------------------|----|
| 1 | Gestion du projet .....          | 38 |
| 2 | Le diagramme de gantt de S9..... | 39 |
| 3 | Les tâches de S9 .....           | 39 |

**H Cahier de test**

|    |  |    |
|----|--|----|
| 1  | test affparam2mat .....                      | 42 |
| 2  | test affparam2geom.....                      | 43 |
| 3  | test warping .....                           | 44 |
| 4  | test distortedToUndistortedSensorCoord ..... | 44 |
| 5  | test undistortedToDistortedSensorCoord ..... | 45 |
| 6  | test imageToWorld.....                       | 46 |
| 7  | test worldToImage .....                      | 46 |
| 8  | test decompFista .....                       | 47 |
| 9  | test classifyOutsideParticlePETS.....        | 47 |
| 10 | test extractCandidates .....                 | 48 |
| 11 | test fifo_stack_v2.....                      | 49 |
| 12 | test szPrePETS .....                         | 49 |
| 13 | test bhattacharyya .....                     | 50 |
| 14 | test motionCheck.....                        | 50 |
| 15 | test updateDictionary.....                   | 51 |
| 16 | test cam2camGroundTrackPETS.....             | 52 |
| 17 | des coordonnées des coins de Python .....    | 52 |
| 18 | des coordonnées des coins de Matlab .....    | 52 |
| 19 | des coordonnées des coins de Python .....    | 52 |
| 20 | des coordonnées des coins de Matlab .....    | 52 |



# Liste des tableaux

## **B Description des interfaces externes du logiciel**

|   |   |    |
|---|---|----|
| 1 | les spécifications de Raspberry Pi..... | 29 |
| 2 | choix de technologies .....             | 30 |



# Liste des Algorithmes

|   |                             |    |
|---|-----------------------------|----|
| 1 | L'algorithme de suivi ..... | 12 |
|---|-----------------------------|----|



# Introduction

## 1 L'acteur du projet est :

La maîtrise d'œuvre (MOE) : WANG Rujia, étudiant en 5<sup>ème</sup> année de l'École d'Ingénieur Polytech'Tours au sein du département Informatique, dans le cadre du PR&D ainsi que ses encadrants Quoc C. LE, D. CONTE, P. GAUCHER, M. HIDANE de l'équipe de recherche RFAI du Laboratoire Informatique de Tours.

## 2 Objectifs

L'objectif de ce projet est qu'implémenter un algorithme de suivi pour un réseau de cameras simple.

De plus, il y a trois objectifs principaux :

- Implémenter un algorithme de suivi pour un réseau de cameras simple
- Chaque caméra sera contrôlée par une Raspberry pi
- Communication entre caméras

Pour ce projet, l'algorithme de suivi utilise plusieurs cameras calibrés et frame- synchronisée avec de chevauchement des domaines de vision pour suivi le même objet.

Le système doit suivre un objet mobile dans une scène spécifique (hall du bâtiment, cours de l'école, une salle de cours...). Pour résoudre le problème des occultations (c'est-à-dire quand un objet est caché par un obstacle ou par d'autres objets), les caméras sont capables de communiquer entre elles pour effectuer un suivi complet sans interruption.

Dans un premier temps, l'algorithme de suivi sera implémenté sur 2 caméras indépendantes entre elles pour un suivi d'un seul objet. Ensuite, on fera communiquer les caméras afin de réaliser un suivi collaboratif.

## 3 Bases méthodologiques

Durant ce projet, le rapport de projet est réalisé à l'aide de Latex.

Pour la modélisation du système, le logiciel <Astarh> est utilisée. Ce logiciel est utilisé dans ce projet afin d'élaborer les diagrammes des cas d'utilisations ainsi que les diagrammes de classe. Parce que le logiciel est installé sur le Raspberry Pi, donc nous devons utiliser le système RASPBIAN.

Pour la programmation, selon les besoins, on a besoin d'utiliser Python3 et OpenCV3.

Pour réaliser l'algorithme de suivi, on a aussi besoin d'utiliser NumPy, SciPy et Matplotlib.

# Description générale

## 1 Environnement du projet

### 1.1 Raspberry Pi

Selon les besoins, c'est obligatoire d'utiliser Raspberry Pi[[WWW4](#)] .

Le Raspberry Pi est un nano-ordinateur monocarte à processeur ARM. Il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux-Debian et des logiciels compatibles.

Il dispose d'un processeur Broadcom BCM2837 64 bit à quatre cœurs ARM Cortex-A53 à 1,2 GHz et d'une puce Wifi 802.11n et Bluetooth 4.1 intégrée.

### 1.2 RASPBIAN

Raspbian est un système d'exploitation libre et gratuit basé sur Debian GNU/Linux optimisé pour fonctionner sur un Raspberry Pi.[[WWW5](#)]

Raspbian est le système d'exploitation officiel de la Fondation. Il est installé à l'aide de NOOBS. Raspbian est pré-installé avec beaucoup de logiciels pour la programmation et l'utilisation générale. Et NOOBS est un programme d'installation de système d'exploitation facile qui contient Raspbian.

Pour installer le système Raspbian dans la carte, on a besoin de télécharger l'image par <https://www.raspberrypi.org/downloads/raspbian/>. Après, on écrit l'image sur la carte SD.

### 1.3 OpenCV

OpenCV (pour Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel. OpenCV peut être utilisé pour développer des programmes de traitement d'image en temps réel, de vision par ordinateur et de reconnaissance de formes.[[WWW3](#)]

Dans le projet, on va utiliser le OpenCV3 pour traiter des images et des vidéos.

Pour installer le OpenCV3 sur Raspberry Pi3, nous devons suivre les étapes ci-dessous :

- Agrandir le système de fichiers
- Installer les dépendances
- Télécharger le code source OpenCV3
- Créer python environnement virtuel
- Installer NumPy
- Compiler et installer OpenCV3

## 1.4 Python

Selon les besoins, on a besoin d'utiliser Python3.

Python est un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

## 1.5 NumPy

Dans ce projet, on a besoin d'utiliser NumPy pour traiter des données.

NumPy est une extension du langage Python. Il peut soutenir un grand nombre de tableau de plusieurs dimensions et d'opérations matricielles. De plus, il fournit un grand nombre de bibliothèques de fonctions mathématiques.

## 1.6 SciPy

SciPy est une bibliothèque Python open-source utilisée pour l'informatique scientifique et l'informatique technique.

SciPy contient des modules pour l'optimisation, l'interpolation, le traitement d'image, etc.

SciPy s'appuie sur l'objet tableau NumPy et fait partie de la pile NumPy qui comprend des outils comme Matplotlib et un ensemble de bibliothèques informatiques scientifiques. Cette pile NumPy a des utilisateurs similaires à d'autres applications telles que MATLAB.

## 1.7 Matplotlib

Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques. Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy. [\[WWW2\]](#)

# 2 Caractéristiques des utilisateurs

Le logiciel final sera utilisé par tout le monde qui n'ont pas forcément de connaissance en informatique.

L'application se voudra donc simple d'utilisation, intuitive et ergonomique.

Les utilisateurs doivent connaître les connaissances de comment utiliser le Raspberry Pi. Les utilisateurs peuvent tourner, stocker et analyser des vidéos.

### 3 Fonctionnalités du système

Selon les besoins, ce projet contient des fonctions suivantes :

- Lancer le Raspberry Pi : Les utilisateurs doivent lancer le Raspberry Pi pour réaliser toutes les autres opérations.
- Tourner et enregistrer la vidéo : Le logiciel doit être exécuté basé sur une vidéo. Les utilisateurs peuvent commencer à tourner la vidéo et enregistrer la vidéo par Raspberry Pi.
- Éteindre le Raspberry Pi : Les utilisateurs doivent fermer le Raspberry Pi après toutes les opérations pour protéger le système de Raspberry Pi.
- Démarrer le logiciel : Le logiciel peut réaliser la fonction de suivi et enregistrer les résultats de suivi automatiquement. L'utilisateur peut choisir un fichier des images. Le résultat de suivi est enregistré en tant que coordonnées dans un fichier texte.
- Visualiser la vidéo et la vidéo avec les résultats de suivi : Les utilisateurs peuvent visualiser la vidéo et observer les résultats de suivi par une interface. Cette partie est réalisé par un groupe de Projet SI.

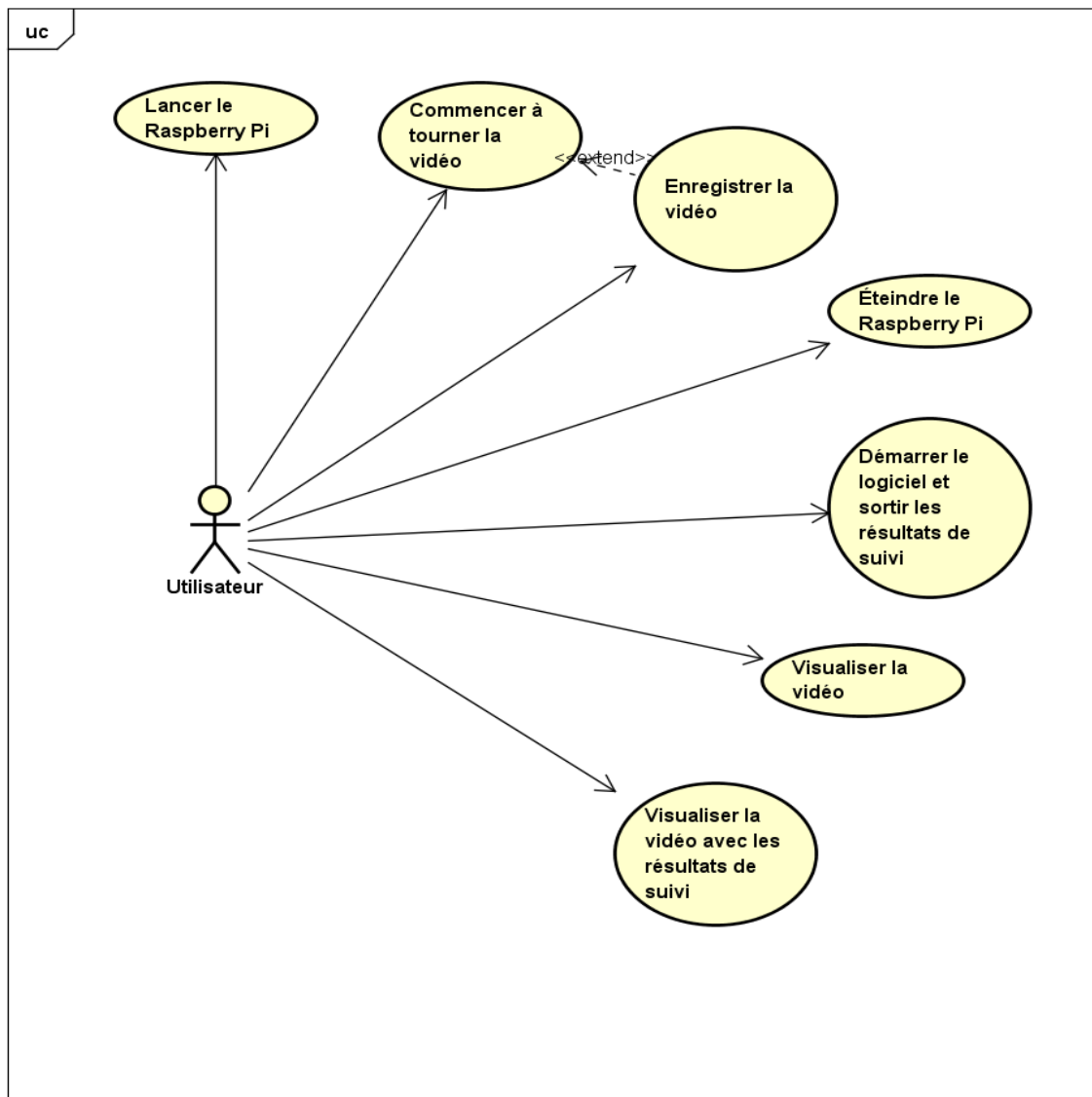


Figure 1 – Diagramme de cas d'utilisation

Tout d'abord, l'utilisateur a besoin de lancer le Raspberry pi. Ensuite, l'utilisateur peut commencer à tourner la vidéo par la commande console pour obtenir le vidéo ou le fichier des images. De plus, l'utilisateur peut choisir d'enregistrer le vidéo, d'éteindre la Raspberry Pi. Après toutes les opérations, l'utilisateur aura besoin d'éteindre le Raspberry Pi pour protéger le système de Raspberry Pi.

De plus, l'utilisateur peut choisir de démarrer le logiciel de suivi, le logiciel va analyser le fichier des images et enregistrer le résultat de suivi (des coordonnées de pixel et des tailles d'objet) dans un fichier texte automatiquement.

L'utilisateur peut regarder les vidéos stockés par une interface. Et cette interface est créé par un groupe de Projet SI. L'utilisateur peut choisir le vidéo ou le fichier des images et les résultats des coordonnées pour le visualiser.

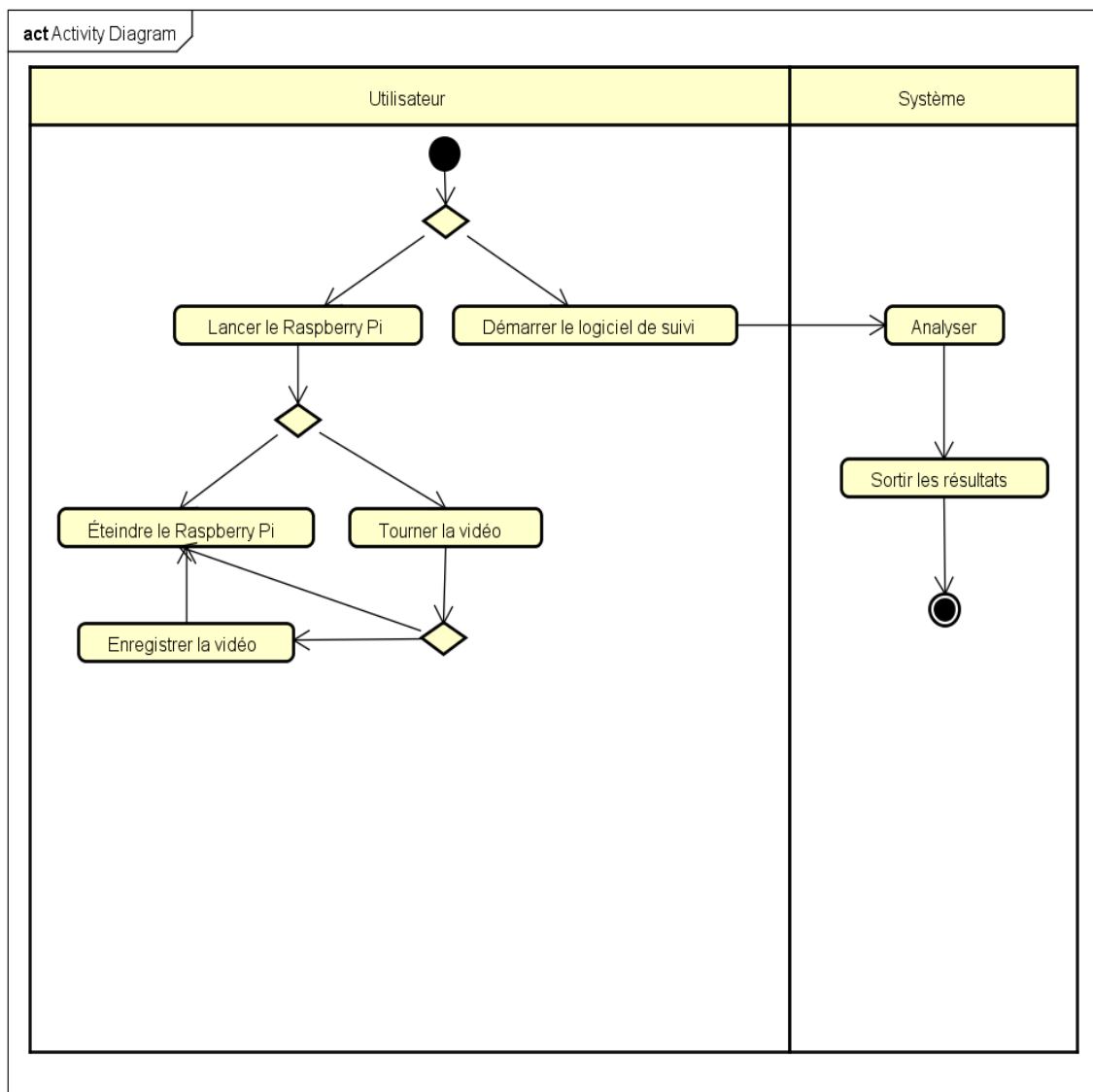


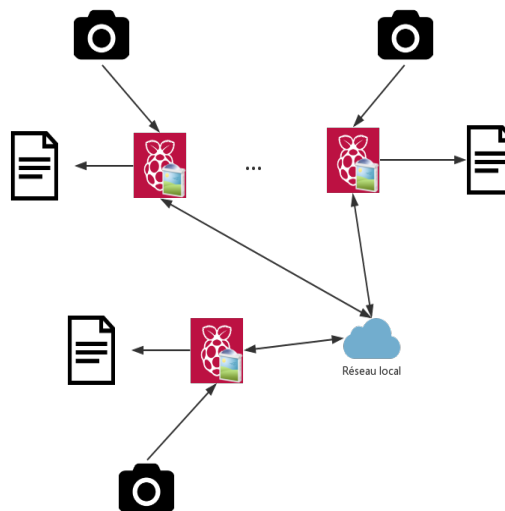
Figure 2 – Diagramme d'activité

## 4 Structure générale du système

La figure ci-dessus représente la structure de notre logiciel.

La caméra capture la vidéo, et le logiciel qui est installé dans le Raspberry Pi va traiter la vidéo frame par frame. Et il peut analyser la vidéo et stocker les résultats (les coordonnées de pixels et les tailles) d'objet suivi dans un fichier texte.

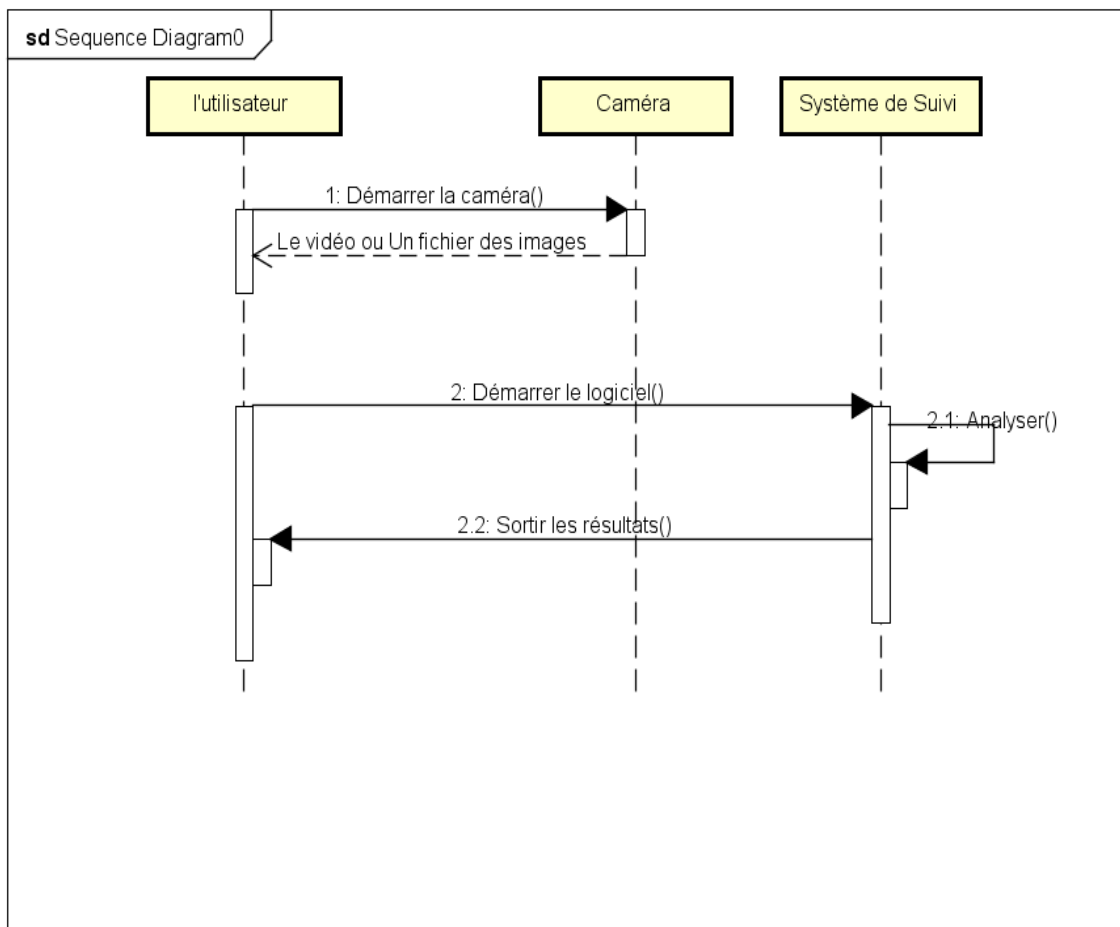
Et le Raspberry Pi peut communiquer avec les autres Raspberry Pi par un réseau local pour échanger des informations sur l'objet de suivi.



**Figure 3** – Structure générale du système

Le diagramme de séquence est ci-dessus.

Une fois que l'utilisateur a lancé le Raspberry Pi. L'utilisateur peut accéder la caméra du Raspberry Pi et obtenir des données de vidéo ou des frames par le console. Ensuite, l'utilisateur peut analyser les données et obtenir les résultats de suivi par le logiciel de suivi. De plus, les résultats de suivi sont stocké dans un fichier texte. Les utilisateurs peuvent utiliser le fichier texte et l'interface de Projet SI pour visualiser les images.



**Figure 4** – Diagramme de séquence



# Etat de l'art/veille

## 1 L'algorithme existant

Pour mon projet de Recherche & Développement, l'algorithme est existant, et il est créé par monsieur Quoc C. LE. Le contenu de cette section est décrit dans l'article «Object Tracking by Sparse Representations on Collaborative Cameras»[2] de monsieur Quoc C. LE.

Malgré les progrès impressionnants réalisés au cours de la dernière décennie, le suivi visuel représente toujours un challenge majeur.

Le suivi d'objet peut être classé en différents sous-domaines (le suivi d'objet unique contre de plusieurs objets et le suivi unique contre multi-caméra). Dans ce projet, l'algorithme est l'algorithme de suivi qui utilise plusieurs cameras calibrés et frame- synchronisée avec de chevauchement des domaines de vision pour suivi le même objet.

L'algorithme de suivi est basé sur des représentations parcimonieuses et le filtre particulaire. Dans notre cadre, chaque caméra calcule son propre résultat de suivi en utilisant son flux vidéo et, éventuellement, des flux vidéo d'autres caméras.

Chaque caméra a un dictionnaire qui est un ensemble de templates (cible et trivial). Un ensemble de candidats (patch d'image, résultat de suivi) est généré par le filtre particulaire. Chaque candidat est représenté comme une combinaison linéaire parcimonieuses de templates cibles et triviaux. Et cet ensemble de templates est appelé dictionnaire. Le candidat ayant le moins d'erreur de reconstruction est sélectionné comme l'objet.

### 1.1 Représentations parcimonieuses

L'objet cible est représenté comme une combinaison linéaire parcimonieuse d'un ensemble de templates. Ces templates appartiennent à l'un des deux groupes : les templates cibles et les templates triviaux. Les templates triviaux sont utilisés pour gérer l'occlusion en utilisant un ensemble de vecteurs unitaires. Les coefficients de représentation pour chaque candidat sont obtenus en résolvant un problème de  $l_1$ -minimisation

$$y = [T, E] \begin{bmatrix} a \\ b \end{bmatrix}$$

Un résultat de suivi,  $y \in \mathbb{R}^d$  est un patch d'image dans le cadre actuel.

$T = [t_1 \dots t_n] \in \mathbb{R}^{d \times n}$  ( $d \gg n$ ) contenant  $n$  templates cibles où  $t_i \in \mathbb{R}^d$

$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)^T \in \mathbb{R}^n$  est appelé un vecteur de coefficient cible

$E = [e_1 \dots e_d] \in \mathbb{R}^{d \times d}$  contenant  $d$  templates triviaux où  $e_i \in \mathbb{R}^d$  est une impulsion de Dirac à l'emplacement  $i$

$b = (b_1, b_2, \dots, b_d)^T \in \mathbb{R}^d$  est appelé un vecteur de coefficient trivial

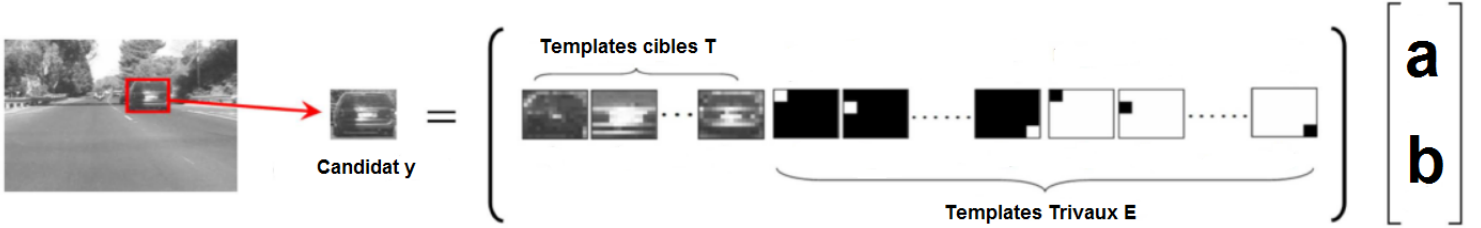


Figure 5 – Représentations parcimonieuses

Dans le cas de l'occlusion, un nombre limité de coefficients triviaux seront activés, alors que le vecteur de coefficient reste parcimonieux. Un bon candidat cible sera représenté par une représentation parcimonieuse. Mais, un mauvais candidat cible sera représenté par une représentation dense.

## 1.2 Méthodes Monte Carlo Séquentielles

Les méthodes monte carlo séquentielles sont un ensemble de méthodes basées sur la simulation qui fournissent une approche pratique et attrayante pour le calcul des distributions postérieures.[1]

On va le utiliser pour la première étape de l'algorithme existant.

## 1.3 Filtre particulaire et $l_1$ -minimization

Nous utilisons le cadre de filtre particulaire populaire qui considère le suivi comme une estimation des états pour un modèle d'espace d'état de séries chronologiques.

Dans le problème de filtrage classique, un système est décrit comme suivant :

$$x_t = f_t(x_{t-1}, v_t) \quad (1)$$

$$y_t = h_t(x_t, W_t) \quad (2)$$

Où  $x_t$  représente l'état caché à l'instant  $t$  du système,  $y_t$  représente l'état de mesure extrait de la séquence d'images,  $f_t$  modélise l'évolution temporelle de  $x_t$ ,  $h_t$  est l'équation de mesure,  $v_t$  et  $w_t$  sont des bruits blancs indépendants.

Le modèle d'état caché est :  $x_t = (x_t, y_t, s_t, \alpha_t)$ , où  $x_t$ ,  $y_t$  sont des coordonnées de pixels,  $s_t$  est le paramètre d'échelle de la boîte englobante et  $\alpha_t$  est la rotation d'angle de la boîte à l'instant  $t$ .

Le filtre bayésien non linéaire consiste à estimer la fonction de densité de filtrage postérieure  $p(x_t | y_{1:t})$  (où  $y_{1:t}$  désigne la série de mesures de l'instant 1 à l'instant  $t$ ), exprimée par :

$$p(x_t | y_{1:t}) = \frac{p(y_t | x_t) p(x_t | y_{1:t-1})}{\int_x p(y_t | x'_t) p(x'_t | y_{1:t-1}) dx'} \quad (3)$$

Où  $\chi$  est l'espace d'état. La densité antérieure est définie comme :

$$p(x_t|y_{1:t-1}) = \int_{\chi} p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \quad (4)$$

Les filtres à particules sont utilisés pour approcher la fonction de densité de filtrage postérieure d'une somme pondérée de N masses de Dirac  $\delta_{x_t^{(n)}}$  centrées sur des réalisations d'états hypothétiques  $\{x_t^{(n)}\}_{n=1}^N$  de l'état  $x_t$ , aussi appelé particules.

Par conséquent, la distribution de filtrage  $\mathbb{P}(dx_t|y_{1:t})$  est récursivement approchée par la distribution empirique  $P_N(dx_t|y_{1:t}) = \sum_{n=1}^N w_t^{(n)} \delta_{x_t^{(n)}}$ , où  $x_t^{(n)}$  est la  $n^{th}$  particule et  $w_t^{(n)}$  est le poids.

Si une approximation de  $\mathbb{P}(dx_t|y_{1:t})$  est connue, le processus est divisé en trois étapes principales

- 1. L'étape de diffusion consiste à estimer  $p(x_t|y_{1:t-1})$  en propageant l'essaim de particules  $\{x_t^{(n)}, w_t^{(n)}\}_{n=1}^N$  en utilisant une fonction d'importance  $q(x_t|x_{0:t}^n, y_t)$
- 2. L'étape de mise à jour calcule ensuite de nouveaux poids de particules en utilisant la nouvelle observation  $y_t$ , comme :

$$w_t^{(n)} \propto w_{t-1}^{(n)} \frac{p(y_t|x_t^{(n)})p(x_t^{(n)}|x_{t-1}^{(n)})}{q(x_t^{(n)}|x_{0:t}^n, y_t)}, \text{ tel que } \sum_{i=1}^N w_t^{(n)} = 1 \quad (5)$$

- 3. Les techniques de rééchantillonnage sont utilisées pour éviter les problèmes de dégénérescence des particules, par exemple au filtre classique de rééchantillonnage d'importance séquentielle (SIR)

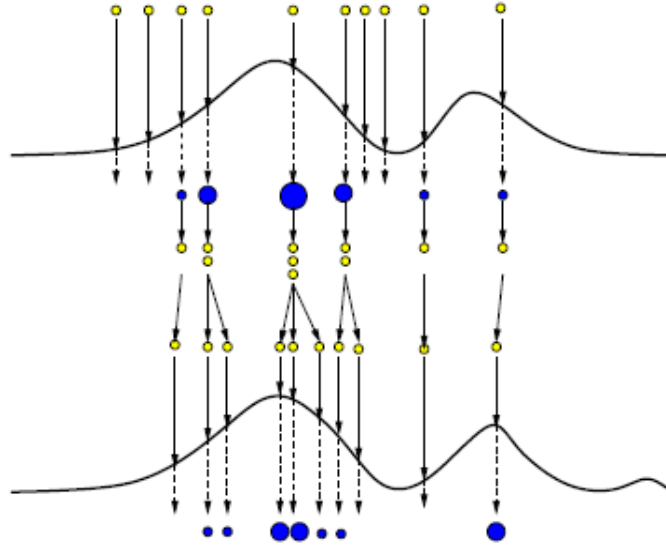


Figure 6 – Bootstrap filtre[1]

En notant  $D = [T, E]$  chaque candidat cible  $y^{(n)}$  est faiblement approximée comme une combinaison linéaire clairsemée d'éléments dans  $D$  en résolvant un problème des  $l_1$  moindres carrés non négatifs normalisé :

$$[a_*^{(n)}, b_*^{(n)}] = \arg \min_{a, b \geq 0} \left\| \begin{bmatrix} a \\ b \end{bmatrix} \right\|_1 + \lambda \left\| [T, E] \begin{bmatrix} a \\ b \end{bmatrix} - y^{(n)} \right\|_2^2 \quad (6)$$

Parmi tous les candidats  $y^{(n)}$ , celui qui a le moins d'erreur de projection sur les modèles cibles est sélectionné, c'est-à-dire  $y^* = y_{i^*}$ , où

$$i^* = \arg \min_{1 \leq n \leq N} \|Ta_*^{(n)} - y^{(n)}\|_2 \quad (7)$$

## 1.4 Suivi collaboratif avec plusieurs caméras utilisant des représentations parcimonieuses

Contrairement au suivi avec une seule caméra, le suivi avec plusieurs caméras peut résoudre le problème de l'occlusion où les cibles intéressantes sont fréquemment masquées par l'environnement des autres objets.

Parmi tous les candidats, le candidat ayant le moins d'erreur de reconstruction est sélectionné comme l'objet. Si la représentation est assez parcimonieuse, le candidat est sélectionné, sinon, chaque candidat local est projeté dans les autres caméras. Dans tous les candidats de toutes les caméras, le candidat ayant le moins d'erreur de reconstruction est sélectionné et le candidat est projeté vers la caméra initiale.

L'algorithme est que chaque caméra communique les centres de ses candidats locaux à d'autres caméras, afin d'aider à choisir le meilleur quand l'occlusion se produit.

## 1.5 L'algorithme de suivi

```

Input : Input :  $K$  cameras  $C_1, \dots, C_K$ ,  $T$  : number of frames,  $N$  : number of candidates,  $\varepsilon$ 
          sparsity level for occlusion detection,  $\lambda$  : parameter used in (6),  $p$  : number of
          frames used to initialize tracking
Output : Complete object tracking for each camera  $C_K$ ,  $1 \leq k \leq K$ 
Initialization
for  $k \leftarrow 1$  to  $K$  do
    Automatically detect target in the first  $p$  frames;
    Construct dictionary  $D_k$  by concatenating all the targets in Camera  $k$ ; -
end
Main loop
for  $t \leftarrow p + 1$  to  $T$  do
    for  $k \leftarrow 1$  to  $K$  do
        Draw the particles following equation (4), then get the patches  $y_{t,k}^{(n)}$ ,  $1 \leq n \leq N$ ;
        Sparse coding for each  $y_{t,k}^{(n)}$  as in (6);
        Compute the reconstruction error for each candidate by considering only target
        templates; Select the candidate  $y_{t,k}^{(n^*)}$  with the minimum error according to (7);
        Compute the sparsity level  $s_{t,k}$  as the ratio between the non-zero target templates
        used in  $y_{t,k}^{(n^*)}$  and the total number of target template;
    end
    if  $s_{t,k} < \varepsilon$  then
        — Select the candidate  $n^*$  as the target;
    else
        — Project the pixel coordinates of local candidates on the 3D space;
        Project these coordinates onto the cameras  $C_{k'}$ ,  $k' \neq k$  and get the corresponding
        bounding boxes;
        Sparse coding of each candidate using (6);
        Select the candidate with minimum reconstruction error using only target
        templates as the target- -
    end
end

```

**Algorithme 1 :** L'algorithme de suivi

## 2 Code Matlab

Afin de m'aider à mieux comprendre ce projet, monsieur Quoccuong Le a me donné des codes Matlab.

Par exemple, le code suivant est utilisé pour convertir des images déformées en des images non déformées

```
function [Xu,Yu] = distortedToUndistortedSensorCoord (Xd,Yd,kappa1)
% correction des pixels
distortion_factor = 1 + kappa1 * (Xd.*Xd + Yd.*Yd);
Xu = Xd.* distortion_factor;
Yu = Yd.* distortion_factor;
end
```



Figure 7 – Image déformée et image non déformée[WWW1]

Par exemple, le code suivant est utilisé pour de convertir des coordonnées 2D en 3D.

```
function w = image2world(px_cam, zp, cam)
ones_array = ones(1, size(px_cam, 2));
Xd = cam.mDpx*(px_cam(1,:) - cam.mCx)/cam.mSx;
Yd = cam.mDpy*(px_cam(2,:) - cam.mCy);
[Xu,Yu] = distortedToUndistortedSensorCoord(Xd, Yd,cam.mKappa1);
X = [Xu;Yu;cam.mFocal*ones_array];
tmp1 = cam.mR\X;
tmp2 = cam.mR\[cam.mTx;cam.mTy;cam.mTz];
m = (zp+tmp2(3))./tmp1(3,:);
w = tmp1.* repmat(m,3,1)-repmat(tmp2,1, size(px_cam, 2));
end
```

### 3 Algorithme de suivi avec une seule caméra

Pour mieux comprendre le projet, j'ai aussi étudié l'algorithme de suivi avec une seule caméra. Et j'ai bien lit l'article "Robust Visual Tracking and Vehicle Classification via Sparse Representation" [3]

L'algorithme de suivi est aussi basé sur des représentations parcimonieuses et le filtre particulière.

Un candidat cible est représenté par une combinaison linéaire de l'ensemble de templates, des templates cibles (obtenus à partir des images précédentes) et des templates triviaux. Et les coefficients de représentation pour chaque candidat sont aussi obtenus en résolvant un problème de  $l_1$ -minimisation. Ensuite, le candidat avec le plus petit erreur de projection est choisi comme résultat de suivi. Après, le filtre particulière est utilisé pour le suivi.

Cette algorithme est un peu différent de notre algorithme.

Pour des représentations parcimonieuses, il y a des templates triviaux négatifs et des templates triviaux positifs.

$$y = [T, E, -E] \begin{bmatrix} a \\ b^+ \\ b^- \end{bmatrix}$$

Dans cette algorithme, l'ensemble de templates cible T est mis à jour par respecter des coefficients du résultat de suivi pour s'adapter à l'illumination et des variations de forme. Mais dans notre algorithme, on a utilisé le filtre Bootstrap pour s'adapter à l'illumination et des variations de forme.

# Analyse et conception

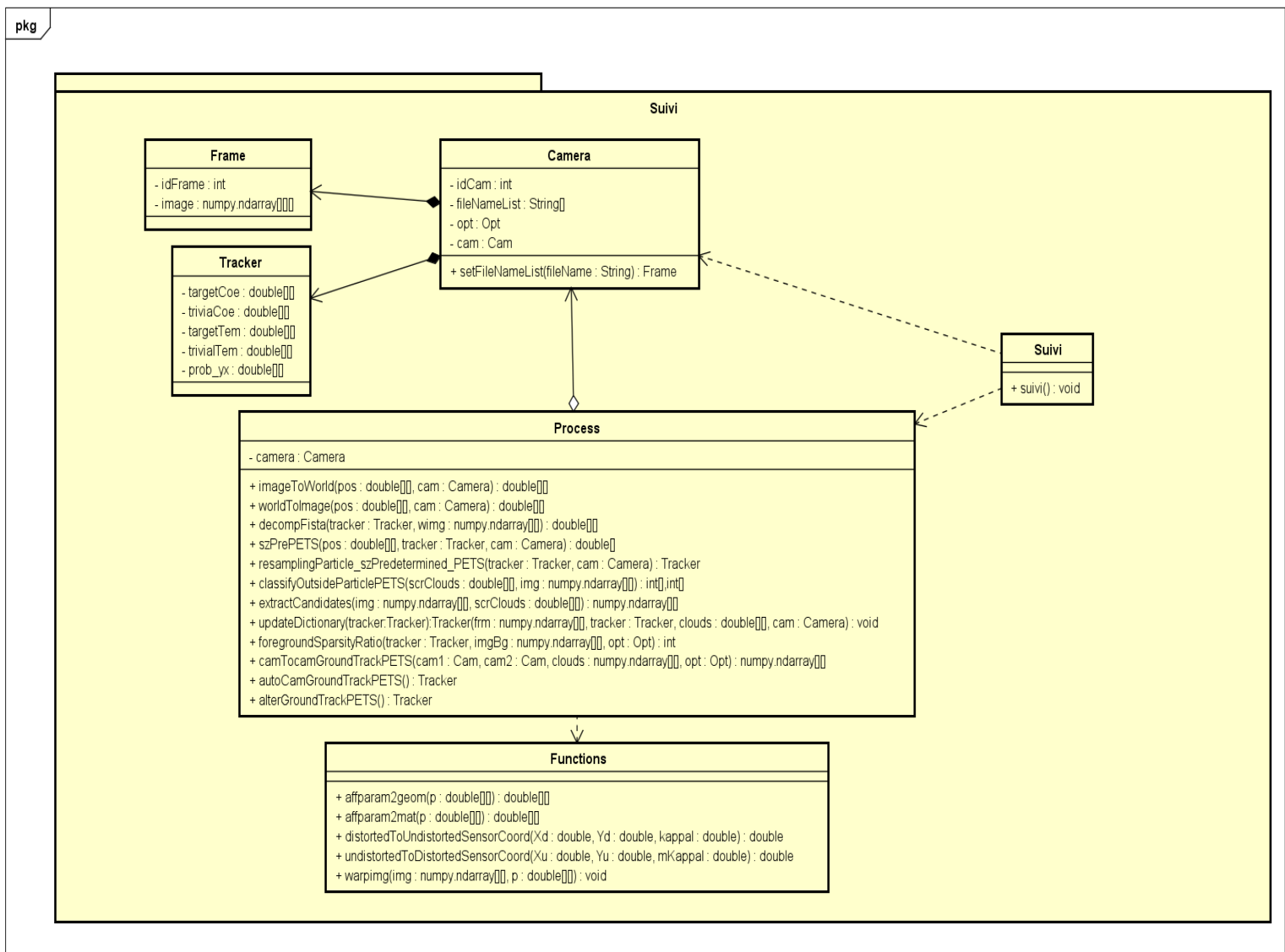


Figure 8 – Diagramme de classe

L'interface de notre projet est programmée par un groupe de Projet SI. Pour le visualiser, il faut sortir les coordonnées d'objet dans un fichier texte.

Parce que le contrôleur est utilisé pour lier la partie vue et la partie modèle. Et il n'y a pas l'interface dans ce projet, alors, je n'utilise pas MVC.

Il y a trois modèles dans ce projets : Frame, Tracker, Camera.

Tracker contient les particules, le dictionnaire qui contient des informations des templates cibles et triviaux, les coefficients de représentations parcimonieuses, les poids, etc.

Frame contient l'id de frame et les valeurs de pixels d'image. Parce que si nous lisons beaucoup d'images à une fois, cela prendrait beaucoup de temps et beaucoup de mémoire. Nous choisissons donc de lire les noms de toutes les images à une fois et ensuite de lire les images à chaque fois par le nom et le chemin de l'image. Il peut gagner le temps.

Camera contient Tracker et Frame. De plus, il contient aussi Opt(les paramètres de Tracker) et Cam(les paramètres de camera, par exemples :la position)

Functions contient des fonctions qui sont servies pour les fonctions de « Process ».

Process contient des fonctions qui représentent les étapes de l'algorithme de suivi.

Fonction suivi() appelle toutes les fonctions dans Process. De plus, elle initialise le Tracker, les paramètres Opt, les paramètres Cam, elle peut aussi sortir les coordonnées de l'objet vers le fichier texte.



# Mise en oeuvre

## 1 Les outils et librairies utilisées

Dans ce projet, j'utilise les outils et les bibliothèques suivants :

- Raspberry Pi  
Installer l'image de système d'exploitation et configurer l'environnement.  
Tourner des vidéos et prendre des photos.
- Pycharm  
Développer et tester le logiciel
- Les bibliothèques utilisées  
OpenCV, NumPy, SciPy et Matplotlib (Expliqué en détail dans la partie Environnement du projet)

## 2 Achèvement

Il y a trois objectifs principaux :

- Implémenter un algorithme de suivi pour un réseau de cameras simple
- Chaque caméra sera contrôlée par une Raspberry Pi
- Communication entre caméras

J'ai complété les tâches suivantes :

- Implémenter l'algorithme de suivi pour un réseau de cameras simple
- Tourner des vidéos et prendre des photos par une Raspberry Pi

## 3 Mise en oeuvre

Tout d'abord, j'ai initialisé le contenu du modèle : Camera, Frame, Tracker.

Pour le modèle Camera, j'ai écrit sa fonction d'initialisation. j'ai aussi écrit des fonctions qui sont utilisés pour calculer quelques paramètres du Camera.

Pour le modèle Frame, j'ai écrit sa fonction d'initialisation. Je peux aussi tourner des vidéos et prendre des photos par une Raspberry Pi pour obtenir des images des frames.

Pour le modèle Tracker, j'ai écrit sa fonction d'initialisation. Les paramètres de Tracker seront défini par la fonction `suivi()` pour réaliser l'algorithme de suivi.

Après, j'ai écrit quelques fonctions de base, `affparam2geom`, `affparam2mat`, `distortedToUndistortedSensorCoord`, `undistortedToDistortedSensorCoord`, `imageToWorld` et `worldToImage`.

Dans cette partie, au début, je n'étais pas très claire sur la structure de l'entrée et de la sortie. Alors, j'ai bien lu le code Matlab, et appris sérieusement Matlab. J'ai bien étudié les différences entre les deux langages. Par exemple, Matlab traite une matrice colonne par colonne, mais Python traite une matrice ligne par ligne. J'ai besoin d'essayer chaque fonction de python pour m'assurer qu'il peut s'exécuter et accomplir la même fonction que Python. Par exemple, la fonction `transpose` est différente en Matlab et en Python. De plus, les fonctions `math.sin()` et `numpy.sin()` ne sont pas la même en Python. De plus, `matrice(:)` en Matlab et `matrice[:]` en python représentent complètement différents.

Ensuite, j'ai écrit la fonction `warpimg` et des fonctions dans la classe `Process`.

Dans cette partie, pour réaliser la fonction `warpimg()`, j'ai rencontré de sérieuses difficultés. J'ai essayé plusieurs méthodes pour atteindre la fonction d'interpolation désirée. Mais j'ai échoué. Donc, finalement, j'ai trouvé la fonction `interp2` de la bibliothèque en Matlab. Et j'ai réalisé la même fonction `interp2` en Python pour atteindre la fonction d'interpolation.

J'ai implémenté des fonctions dans l'ordre de l'algorithme. Tout d'abord, c'est l'initialisation d'objet. Après, il y a des fonctions de jugement et de prétraitement. Ensuite, j'ai implémenté la fonction qui est utilisé pour ré-échantillonnage et génération de particules et la fonction codage parcimonie. Il y a beaucoup de choses à faire attention à. Par exemple, la fonction `cholesky` de `numpy` et la fonction `cholesky` de `scipy` ne sont pas la même, il faut bien étudier et tester les fonctions. Et pour traiter des images, parfois `opencv` est meilleur, parfois `Matplotlib` est meilleur.

Enfin, J'ai complété la fonction principale par appeler les fonctions dans la classe `Process`.

Basé sur la suggestion de M. Ronan, j'ai changé mon diagramme de classe, changé la structure de mon code et refaçoné mon code.

## 4 Limites

Dans ce logiciel, il y a un problème majeur qui doit être optimisé. Ce logiciel prend beaucoup de temps et ne peut pas être effectuée en temps réel. Tout d'abord, Matlab lit les images plus rapidement que Python(avec OpenCV). Ensuite, pour implémenter cet algorithme, de nombreuses boucles sont utilisées, les boucles prennent de temps. Enfin, pour réaliser une même fonctions, il y a des façons différents. Il faut tester quelle méthode est la plus efficace.

## 5 Développement futur

Pour continuer ce projet, il y a des choses à faire suivants :

- Optimiser ce logiciel pour gagner le temps

Pour optimiser ce logiciel :

Dans le logiciel, j'ai utilisé plusieurs façons pour obtenir l'indice maximal, il faut tester et trouver le façon plus efficace.

Dans le logiciel, il y a beaucoup de boucles, il faut essayer de réduire le nombre de boucles.

Dans le logiciel, pour assigner les valeurs à une matrice par une autre matrice, j'ai utilisé `copy` et `deepcopy`, et il prend beaucoup de temps.

- Installer dans Raspberry Pi  
Besoin de définir l'entrée comme l'image prise par la caméra de Raspberry Pi.
- Communication entre les Raspberry Pi  
Il faut trouver une façon de communication, par exemple : wifi ou bluetooth.
- Le logiciel rejoint l'interface

## 6 Résultats

C'est le résultat que j'obtiens dans le fichier txt.

```

Frame 1
1 239 644 321 685
Frame 2
1 234 644 322 685
Frame 3
1 239 637 321 678
Frame 4
1 240 628 323 669
Frame 5
1 238 612 323 653
Frame 6
1 236 603 319 647
Frame 7
1 234 595 317 636
Frame 8
1 239 584 321 626
Frame 9
1 235 578 318 619
Frame 10
1 237 568 318 609
Frame 11
1 238 561 319 602
Frame 12
1 231 541 311 582
Frame 13
1 231 535 309 574

```

**Figure 9** – *Le résultat du fichier txt*

J'utilise l'interface et le fichier sortie qui contient les coordonnées de l'objet pour visualiser les résultats.



Figure 10 – Frame 1



Figure 11 – Frame 2





Figure 12 – Frame 3



Figure 13 – Frame 4





Figure 14 – Frame 5



Figure 15 – Frame 6





Figure 16 – Frame 7



Figure 17 – Frame 8





Figure 18 – Frame 9



Figure 19 – Frame 10



# Bilan et conclusion

## 1 Planning du s10

|                     |          |         |
|---------------------|----------|---------|
| Code_Model          | 17-12-20 | 18-1-10 |
| Code_Init           | 18-1-10  | 18-1-18 |
| Code_Function_A     | 18-1-18  | 18-1-25 |
| Code_Fuction_B      | 18-1-25  | 18-2-12 |
| Code_Algorithm      | 18-2-12  | 18-2-16 |
| Test                | 18-2-16  | 18-3-16 |
| Rapport de S10      | 18-3-16  | 18-3-30 |
| Soutenance          | 18-3-30  | 18-4-4  |
| Lire le code Matlab | 17-12-20 | 18-2-16 |

Figure 20 – Les tâches de S10

Pour le semestre 10, j'ai les tâches suivantes :

Tâche Code\_Model : Complétez le code de la partie Model

Tâche Code\_Init : Complétez le code de l'initialisation

Tâche Code\_Function\_A : Complétez les méthodes "worldToImage", "imageToWorld", "undistortedToDistorted" et "distortedToUndistorted" de la classe Functions et de la classe Process

Tâche Code\_Function\_B : Complétez les autres méthodes de la classe Process

Tâche Code\_Algorithm : Complétez le code de la classe suivi

Tâche Test : Tester et optimiser le code

Tâche Rapport de S10 : Écrire le rapport de PR&D

Tâche Soutenance : Préparer la soutenance

Tâche Lire le code Matlab : Afin de mieux comprendre cet algorithme et de mieux programmer, j'ai besoin de bien lire et bien comprendre le code Matlab.

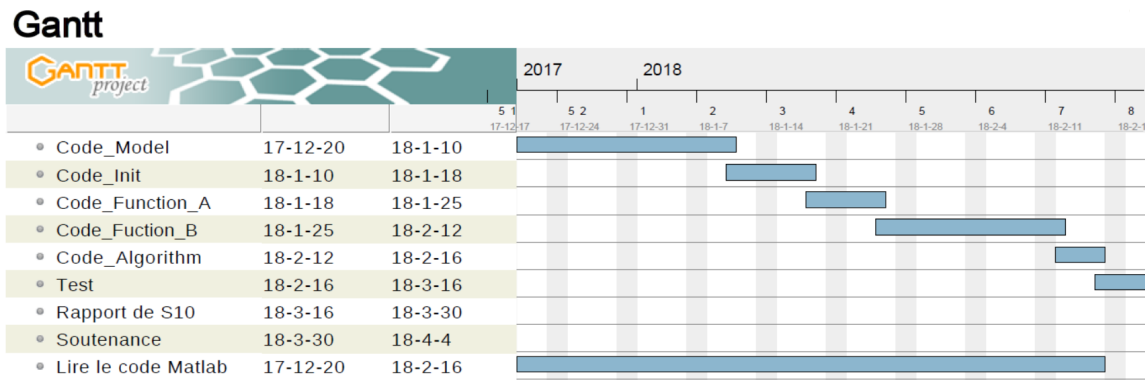


Figure 21 – Le diagramme de gantt de S10

## 2 Redards

J'avais un peu de retard dans la phase Code\_Function\_A. Parce qu'il a quelques différences entre python et matlab. J'ai essayé 6 façons de réaliser la fonction warping. Mais j'ai échoué parce qu'il n'y a pas de fonction qui peut réaliser la même fonctionnalité de interp2 de Matlab qui est utilisé dans la fonction warping. Enfin, pour bien réaliser la fonctionnalité de la fonction warping, je choisis de traduire la fonction interp2 de matlab en python.

## Annexes



## Glossaire

NOOBS : New Out Of the Box Software

SoC : System on Chip

CPU : Central Processing Unit

# B

## Description des interfaces externes du logiciel

### 1 Interfaces matériel/logiciel

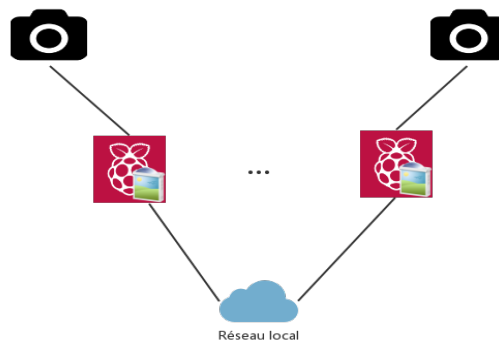


Figure 1 – Interfaces matériel/logiciel

Le logiciel de suivi est installé sur chaque Raspberry Pi. Et chaque Raspberry Pi est connecté à une caméra. De plus, chaque Raspberry Pi est connecté au réseau local pour échanger des informations.

Le logiciel fonctionnera sur une machine Raspberry Pi ayant les spécifications suivantes

Table 1 – les spécifications de Raspberry Pi

|                         |                                   |
|-------------------------|-----------------------------------|
| SoC                     | Broadcom BCM2837                  |
| CPU                     | 1,2 GHz quadricœur ARM Cortex-A53 |
| Mémoire                 | 1 Go                              |
| Nombre de ports USB 2.0 | 4                                 |
| Unité de lecture        | MicroSD                           |
| Puissance nominale      | 800 mA (4 W)                      |

## 2 Interfaces homme/machine

La visualisation est le travail du Projet SI. Pour la visualisation, le logiciel va générer un fichier texte contenant les coordonnées et les tailles d'objet de suivi. Et les utilisateurs ont besoin de commencer à tourner la vidéo et enregistrer la vidéo par des commandes console. Alors, dans mon PR&D, je n'ai pas besoin d'écrire l'interface.

## 3 Interfaces logiciel/logiciel

Selon les besoins, ce logiciel n'utilise pas la base donnée. Il obtient les données à partir de caméra de Raspberry Pi. Donc ce logiciel ne dépend d'aucun autre logiciel.

Et des choix de technologies sont présentés dans le tableau suivant :

**Table 2 – choix de technologies**

|                          |              |
|--------------------------|--------------|
| Composant serveur        | Nom, Version |
| Système d'exploitation   | RASPBIAN     |
| Composant applicatif     |              |
| Langage de programmation | python       |
| Bibliothèque             | OpenCV       |
| Extension                | NumPy        |

# C

## Spécifications fonctionnelles

### 1 Définition de la fonction Convertir de coordonnées 2D en 3D

Présentation :

Permet de convertir les coordonnées 2D sur le plan d'image vers 3D sur plan z du monde réel. Dans l'algorithme, si le candidat sélectionné est plus grand du niveau de sparsity de détection d'occlusion, on a besoin de projeter l'objet dans le monde réel.

Description :

Entrée : coordonnées  $[x, y]'$  en pixel, la hauteur du plan z, paramètres de calibration de caméra.

Sortie : coordonnées  $[x, y, z]'$  en mm.

### 2 Définition de la fonction Convertir de coordonnées 3D en 2D

Présentation :

Permet de convertir les coordonnées 3D sur plan z du monde réel vers 2D sur plan d'image.

Description :

Entrée : coordonnées  $[x, y, z]'$  en mm, paramètres de calibration de caméra.

Sortie : coordonnées  $[x, y]'$  en pixel.

### 3 Définition de la fonction Convertir des images déformées en des images non déformées

Présentation :

Permet de corriger la distorsion de l'objectif, mesurer la taille d'un objet dans les unités du monde réel pour reconstruction de scène en monde réel. Cette fonction est utilisée pour la fonction « convertir de coordonnées 2D en 3D ».

Description :

Entrée : coordonnées  $[x, y]'$  en pixel et paramètres d'étalonnage de la caméra.

Sortie : coordonnées  $[x, y]'$  en mm.

## 4 Définition de la fonction Convertir des images non déformées en des images déformées

Présentation :

Cette fonction effectue la fonction inverse de la fonction « Convertir des images déformées en des images non déformées »

Description :

Entrée : coordonnées  $[x, y]'$  en mm et Paramètres d'étalonnage de la caméra.

Sortie : coordonnées  $[x, y]'$  en pixel.

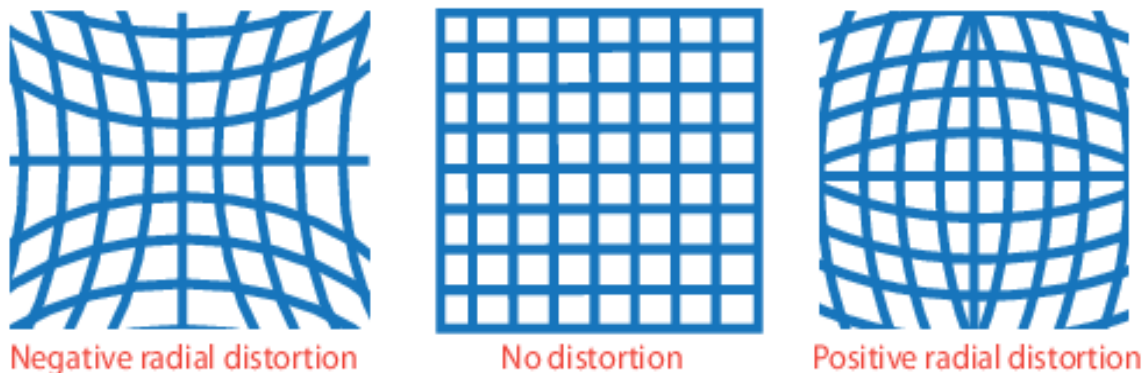


Figure 1 – Distortion d'image[WWW1]

## 5 Définition de la fonction Interpolation de patch d'image

Présentation :

Permet d'extraire un patch d'image avec la taille souhaitée.

Description :

Entrée :  $p[p_x, p_y, s_x, s_y, \theta]'$ , où  $p_x$  et  $p_y$  sont  $\theta$  coordonnées du centre de la boîte englobant  $s_x$  et  $s_y$  sont la taille de la boîte avec  $x$  (longueur) et  $y$  (hauteur) dimensions,  $\theta$  est la rotation d'angle de la boîte.

La taille du patch souhaité  $d \times d$  (e. g [32×32])

Sortie : Patch de taille  $d \times d$

## 6 Définition de la fonction Ré-échantillonnage et génération de particules

Présentation :

C'est la première étape dans la boucle principale d'algorithme. Permet de ré-échantillonner et générer les particules selon leur poids.

Description :

Entrée :  $p[p_x, p_y, s_x, s_y, \theta]'$  particules au frame  $t$

Sortie :  $p[p_x, p_y, s_x, s_y, \theta]'$  particules au frame  $t + 1$



## 7 Définition de la fonction Calculer des coefficients de modèles

Présentation :

Permet de calculer des coefficients de modèles, qui est utilisé pour représentation parcimonieuse.

Description :

Entrée : le dictionnaire, patch d'image.

Sortie : Deux tableaux de type double pour stocker les coefficients.

## 8 Définition de la fonction Codage parcimonie

Présentation :

Permet d'utiliser les coefficients de modèles et de faire la représentation parcimonieuse.

Description :

Entrée : le modèle caméra qui contient des informations de frame, dictionnaire, candidat, etc.

Sortie : tracker, les particules, la probabilité.

## 9 Définition de la fonction Calculer l'erreur de reconstruction

Présentation :

Permet de calculer l'erreur de reconstruction pour chaque candidat en ne considérant que les modèles cibles.

Description :

Entrée : le modèle caméra qui contient des informations de dictionnaire, coefficients, etc.

Sortie : le modèle tracker qui a le moins d'erreur de reconstruction.

## 10 Définition de la fonction Calculer le niveau de sparsity

Présentation :

C'est la cinquième étape dans la boucle principale d'algorithme. Permet de calculer le niveau de sparsity du candidat qui a le moins d'erreur de reconstruction.

Description :

Entrée : le modèle tracker qui contient des informations coefficients.

Sortie : le niveau de sparsity.

## 11 Définition de la fonction Construire le dictionnaire

Présentation :

Permet de construire le dictionnaire pour représentations parcimonieuse.

Description :

Entrée : les pièces d'images que le logiciel détecte.

## 12 Définition de la fonction Mettre à jour le dictionnaire

Présentation :

Permet de mettre à jour le dictionnaire.

Description :

Sortie : le modèle tracker qui contient des informations sur le dictionnaire.

## 13 Définition de la fonction Obtenir les frames

Présentation :

Permet d'obtenir des images par le Raspberry Pi et les stocker dans le modèle caméra.

Description :

Entrée : des images qui est obtenu par le Raspberry Pi.

## 14 Définition de la fonction Sortie

Présentation :

Permet de stocker les coordonnées d'objet de suivi dans le fichier texte. Et le fichier texte est utilisé pour la visualisation.

Description :

Entrée : des coordonnées d'objet.

# D

## Spécifications non fonctionnelles

### 1 Contraintes de développement et conception

Le logiciel va être exécuter sur le Raspberry Pi.

Et le langage de programmation que l'on va adopter pour le développement est Python.

Les bibliothèques de programmes imposées sont NumPy, OpenCV, SciPy, etc.

### 2 Contraintes de fonctionnement et d'exploitation

#### 2.1 Performances

Ce logiciel devrait pouvoir traiter la vidéo en temps réel et afficher le résultat en temps réel.

#### 2.2 Capacités

Pour installer le OpenCV sur le Raspberry Pi, le SD card doit être 8GB au minimum.

La taille de vidéo que ce logiciel peut théoriquement analyser n'est pas sûr pour l'instant. Mais la taille de vidéo ne peut pas être plus grande de la mémoire du Raspberry Pi.

Ce projet est basé sur le Raspberry Pi. Et la caméra du Raspberry a des capacités limitées. la caméra peut prendre des images  $2592 \times 1944$  au maximum. Et la caméra peut prendre la vidéo 1080p30, 720p60 et  $640 \times 480$ p60/90.

#### 2.3 Sécurité

Le logiciel ne sera pas sensible aux failles de sécurité.

# E

## Cahier du developpeur

Pour le diagramme de cas d'utilisation, le diagramme d'activité et le diagramme de séquence. Vous pouvez lire les parties Structure générale du système et fonctionnalités du système.

Le projet n'utilise pas MVC, parce que le contrôleur est utilisé pour lier le vue et le modèle. Mais dans ce projet, l'interface est réalisée par un groupe SI. Pour réaliser la visualisation, on a besoin de sortir un fichier texte qui contient des coordonnées d'objet.

Il y a quelques étapes importantes.

- Initialisation
- Suivi de l'objet
- Changer la caméra pour suivre l'objet
- Sortir le résultat

### 1 Intialisation

Permet d'initialiser des paramètres de la caméra (Cam) et des paramètres de l'objet de suivi (Opt). De plus, permet d'initialiser l'objet de suivi et ajouter l'objet de suivi dans le système.

### 2 Suivi de l'objet

Permet de suivre l'objet avec un seule caméra.

Le suivi de l'objet est réalisé par des fonctions dans « Process ». Et toutes les fonctions de « Functions » sont servies pour les fonctions de « Process ».

Pour savoir les fonctionnalités de chaque fonctions, vous pouvez lire la partie spécifications fonctionnelles.

### 3 Changer la caméra pour suivre l'objet

Lorsqu'il y a des obstacles, les deux caméras échangent des informations. L'autre caméra suivra les objets par fonction « alterGroundTrackPETS ».

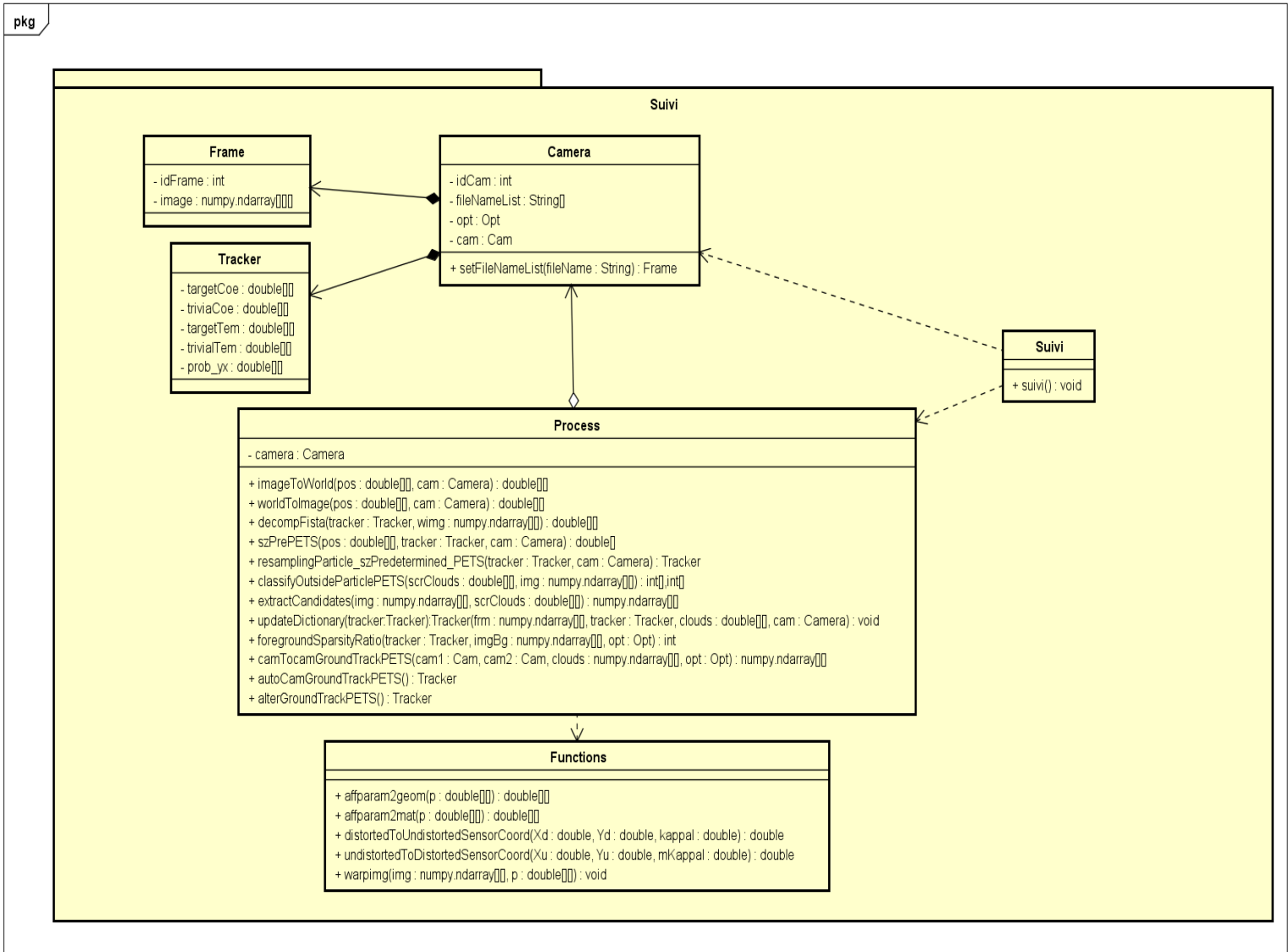


Figure 1 – Diagramme de classe

## 4 Sortir le résultat

Le logiciel va sortir le résultat vers un fichier Texte. L'utilisateur peut le visualiser par l'interface qui est réalisé par le groupe de projet libre.

# F

## Gestion de projet

### 1 Gestion de projet

Trello est un outil de gestion de projet en ligne.

Je utilise Trello pour lister les tâches. De plus, je utilise Trello pour voir l'avancement.

Je peut savoir les tâches à faire, les tâches en cours et les tâches terminés par Trello. Alors, je peut savoir qu'est-ce que je suis retard.

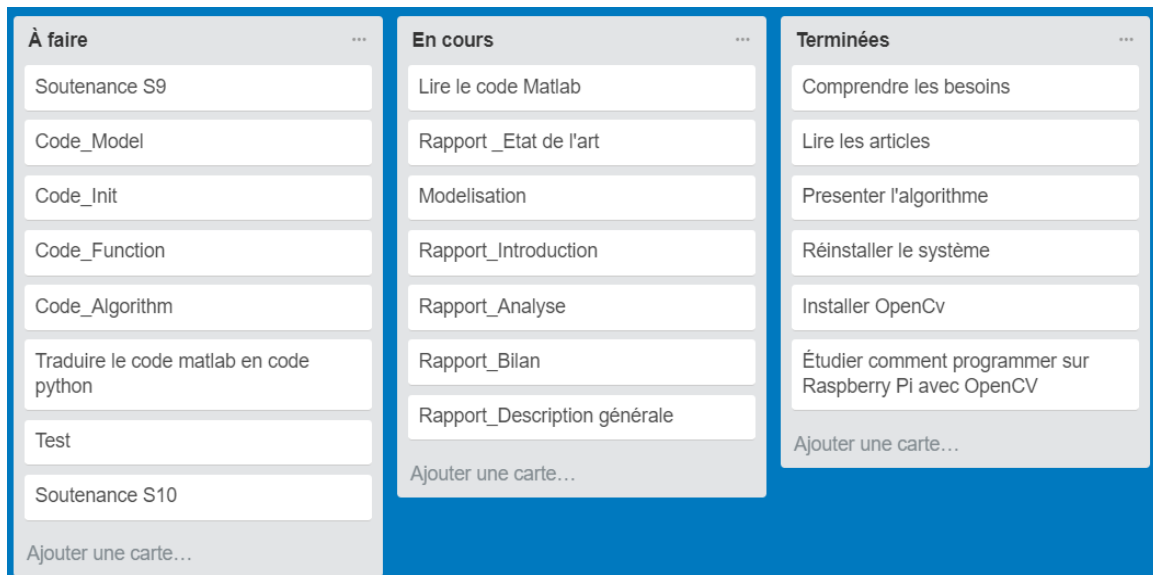


Figure 1 – Gestion du projet

### 2 Planification S9

Dans la semestre 9, le projet commence le 21/09/2017 à comprendre les besoins et le sujet de projet. Et le projet finit le 13/09/2017 à faire la présentation

## Gantt

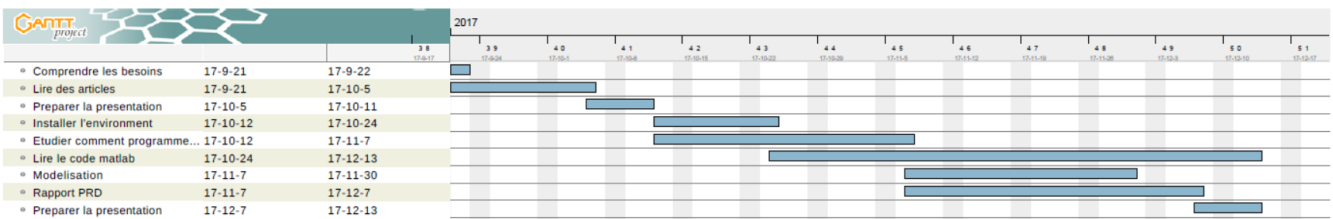


Figure 2 – Le diagramme de gantt de S9

|                                   |          |          |
|-----------------------------------|----------|----------|
| Comprendre les besoins            | 17-9-21  | 17-9-22  |
| Lire des articles                 | 17-9-21  | 17-10-5  |
| Preparer la presentation          | 17-10-5  | 17-10-11 |
| Installer l'environnement         | 17-10-12 | 17-10-24 |
| Etudier comment programmer sur RP | 17-10-12 | 17-11-7  |
| Lire le code matlab               | 17-10-24 | 17-12-13 |
| Modelisation                      | 17-11-7  | 17-11-30 |
| Rapport PRD                       | 17-11-7  | 17-12-7  |
| Preparer la presentation          | 17-12-7  | 17-12-13 |

Figure 3 – Les tâches de S9

Au cours de semestre 9, j'ai terminé chaque mission comme prévu. J'ai également rencontré des problèmes.

Difficultés rencontrées :

J'ai passé beaucoup de temps à installer OpenCV. Parce que la carte SD n'est pas assez grande, chaque fois je dois supprimer un logiciel pour faire de la place pour l'installation.



# Doc d'installation et d'utilisation

## 1 installation d'image de système d'exploitation

Pour installer une image du système d'exploitation Raspberry Pi sur une carte SD, il y a des étapes suivantes :

- Télécharger l'image : Les images officielles des systèmes d'exploitation peuvent être téléchargées à partir du Web Raspberry Pi.
- Ecrire l'image sur la carte SD avec Etcher.

## 2 installation d'OpenCV

Pour installer OpenCV 3 sur un Raspberry Pi, il y a des étapes suivantes :

- Développez le système de fichiers pour inclure tout l'espace disponible sur votre carte micro-SD.
- Installer les dépendances avec les commandes  
mettre à jour tous les paquets existants :

```
sudo apt-get update  
sudo apt-get upgrade
```

installer CMake :

```
sudo apt-get install build-essential cmake pkg-config
```

installer I/O paquets :

```
sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev  
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev  
sudo apt-get install libxvidcore-dev libx264-dev
```

installer sous-module highgui, qui est utilisé pour afficher des images sur écran et créer des interfaces graphiques :

```
sudo apt-get install libgtk2.0-dev
```

d'autre dépendance :



```
sudo apt-get install libatlas-base-dev gfortran
```

installer python :

```
sudo apt-get install python2.7-dev python3-dev
```

installer pip :

```
wget https://bootstrap.pypa.io/get-pip.py
```

```
sudo python get-pip.py
```

#### — Configurer les environnements virtuels

```
sudo pip install virtualenv virtualenvwrapper
```

```
sudo rm -rf ~/.cache/pip
```

```
export WORKON_HOME=$HOME/.virtualenvs
```

```
source /usr/local/bin/virtualenvwrapper.sh
```

```
echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.profile
```

```
echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.profile
```

```
echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.profile
```

```
source ~/.profile
```

```
mkvirtualenv cv -p python2
```

```
mkvirtualenv cv -p python3
```

```
source ~/.profile
```

```
workon cv
```

#### — Compiler et installer OpenCV

```
$ cd ~/opencv-3.1.0/
```

```
$ mkdir build
```

```
$ cd build
```

```
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
```

```
-D CMAKE_INSTALL_PREFIX=/usr/local \
```

```
-D INSTALL_PYTHON_EXAMPLES=ON \
```

```
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.1.0/modules \
```

```
-D BUILD_EXAMPLES=ON ..
```

```
make -j4
```

## 3 Configurer le chemin de fichier

#### — le chemin d'image background

```
bg_view1 = mpimg.imread("BG\\00000001.jpg")
```

#### — le chemin des fichiers des images

```
files = os.listdir("Image\View_001")
```

Pour tester le projet et les fonctions. Je compare les résultats de python avec les résultats de matlab. Le but du test est de vérifier que le logiciel respecte ses spécifications. Le test consiste à vérifier que le logiciel réalise les fonctions attendues et ce correctement.

## 1 Unit Test

### 1.1 test\_affparam2mat

test\_affparam2mat convertit 6 paramètres affines (x, y, th, scale, aspect, skew) en une matrice 2\*3

L'entrée de matlab :

```
[665;280;2.5625;0;2;0]
```

Le résultat de matlab :

```
[665;280;2.5625;0;0;5.125]
```

L'entrée de python :

```
np.array([[665],[280],[2.5625],[0],[2],[0]])
```

Le résultat de python :

```
np.array([[665],[280],[2.5625],[0],[0],[5.125]])
```

```
Launching unittests with arguments python -m unittest Test.Test.test_affparam2mat in C:\Users\WRJ\PycharmProjects\PRD_FinalVersion
```

```
Ran 1 test in 0.001s
```

```
OK
```

**Figure 1** – test affparam2mat

## 1.2 test\_affparam2geom

affparam2geom convertit une matrice 2\*3 en 6 paramètres affines (x, y, th, scale, aspect, skew)

L'entrée de matlab :

```
[665;280;2.5625;0;0;5.125]
```

Le résultat de matlab :

```
[665;280;2.5625;0;2;0]
```

L'entrée de python :

```
np.array([[665],[280],[2.5625],[0],[0],[5.125]])
```

Le résultat de python :

```
np.array([[665],[280],[2.5625],[0],[2],[0]])
```

Résultat de test :

```
Launching unittests with arguments python -m unittest Test.Test.test_affparam2geom in C:\Users\WRJ\PycharmProjects\PRD_FinalVersion
```

```
Ran 1 test in 0.006s
```

```
OK
```

**Figure 2 – test affparam2geom**

## 1.3 test\_warpimg

Rééchantillonnage d'image en utilisant le mappage et l'interpolation linéaire

Pour tester cette fonction, je génère l'entrée de matlab dans un fichier texte et utilise la matrice stockée dans ce fichier comme l'entrée de python. Ensuite, je génère la sortie de matlab dans un fichier texte et compare la sortie de python avec le contenu de ce fichier texte.

L'entrée de matlab

warpimgInput.txt

Le résultat de matlab

warpimgOutput.txt

L'entrée de python

```
np.loadtxt('test_txt/warpimgInput.txt')
```

Le résultat de python

```
warpimg()
```

Résultat de test :

Launching unittests with arguments python -m unittest Test.Test.test\_warping in C:\Users\WRJ\PycharmProjects\PRD\_FinalVersion

Ran 1 test in 0.523s

OK

**Figure 3 – test warping**

## 1.4 test\_distortedToUndistortedSensorCoord

Convertir des images déformées en des images non déformées. Permet de corriger la distorsion de l'objectif, mesurer la taille d'un objet dans les unités du monde réel pour reconstruction de scène en monde réel.

L'entrée de matlab

```
Xd = 1.597463883662186
Yd = 0.178715772628500
mKappa = 0.005111304363900
```

Le résultat de matlab

```
Xu = 1.618561177843289
Yu = 0.181076025820142
```

L'entrée de python

```
Xd = 1.597463883662186
Yd = 0.178715772628500
mKappa = 0.005111304363900
```

Le résultat de python

```
np.array([1.618561177843289,0.181076025820142])
```

Résultat de test :

Launching unittests with arguments python -m unittest Test.Test.test\_distortedToUndistortedSensorCoord in C:\Users\WRJ\PycharmProjects\PRD\_FinalVersion

Ran 1 test in 0.001s

OK

**Figure 4 – test distortedToUndistortedSensorCoord**

## 1.5 test\_undistortedToDistortedSensorCoord

Convertir des images non déformées en des images déformées. Cette fonction effectue la fonction inverse de la fonction « Convertir des images déformées en des images non déformées »

L'entrée de matlab

```
Xu = 1.687931837186443
Yu = 0.180500559982071
mKappa = 0.005111304363900
```

Le résultat de matlab

```
Xd = 1.664107860120738
Yd = 0.177952920849600
```

L'entrée de python

```
Xu = 1.687931837186443
Yu = 0.180500559982071
mKappa = 0.005111304363900
```

Le résultat de python

```
np.array([1.664107860120738,0.177952920849600])
```

Résultat de test :

```
Launching unittests with arguments python -m unittest Test.Test.test_undistortedToDistortedSensorCoord in C:\Users\WRJ\PycharmProjects\PRD_FinalVersion
```

```
Ran 1 test in 0.001s
```

```
OK
```

**Figure 5** – *test\_undistortedToDistortedSensorCoord*

## 1.6 test\_imageToWorld

Permet de convertir les coordonnées 2D sur le plan d'image vers 3D sur plan z du monde réel.

Dans l'algorithme, si le candidat sélectionné est plus grand de le niveau de sparsity de détection d'occlusion, on a besoin de projeter l'objet dans le monde réel.

L'entrée de matlab

```
[U,V] = meshgrid(-393+1:4:393,-288+1:4:288);
U = U + 393;
V = V + 288;
[U(:)';V(:)']
```

Le résultat de matlab

posp1.txt

L'entrée de python

```
[U,V] = np.meshgrid(np.arange(-392,393,4),np.arange(-287,288,4))
U = U +393
V = V +288
np.vstack((np.reshape(np.transpose(U), -1), np.reshape(np.transpose(V), -1)))
```

Le résultat de python

```
np.loadtxt('test_txt/posp1.txt')
```

Résultat de test :

```

Launching unittests with arguments python -m unittest Test.Test.test_imageToWorld in C:\Users\WRJ\PycharmProjects\PRD_FinalVersion

Ran 1 test in 0.172s

OK

```

Figure 6 – test *imageToWorld*

## 1.7 test\_worldToImage

Permet de convertir les coordonnées 3D sur plan z du monde réel vers 2D sur plan d'image.

Pour tester cette fonction, je génère l'entrée de matlab dans un fichier texte et utilise la matrice stockée dans ce fichier comme l'entrée de python. Ensuite, je génère la sortie de matlab dans un fichier texte et compare la sortie de python avec le contenu de ce fichier texte.

L'entrée de matlab

wToIInput.txt

Le résultat de matlab

wToIOutput.txt

L'entrée de python

```
np.loadtxt('test_txt/wToIInput.txt')
```

Le résultat de python

```
worldToImage(testInput, cam1)
```

Résultat de test :

```

Launching unittests with arguments python -m unittest Test.Test.test_worldToImage in C:\Users\WRJ\PycharmProjects\PRD_FinalVersion

Ran 1 test in 0.076s

OK

```

Figure 7 – test *worldToImage*

## 1.8 test\_decompFista

C'est l'algorithme de Fast Iterative Shrinkage-Thresholding. Il est utilisé pour calculer le coefficient des modèles.

Pour tester cette fonction, je génère l'entrée de matlab dans un fichier texte et utilise la matrice stockée dans ce fichier comme l'entrée de python. Ensuite, je génère la sortie de matlab dans un fichier texte et compare la sortie de python avec le contenu de ce fichier texte.

L'entrée de matlab

fistaInput.txt

B.txt

Le résultat de matlab

fistaOutput.txt

L'entrée de python

```
np.loadtxt('test_txt/fistaInput.txt')
np.loadtxt('test_txt/B.txt')
```

Le résultat de python

```
decompFista(B, testInput, 10, 100)
```

Résultat de test :

```
Launching unittests with arguments python -m unittest Test.Test.test_decompFista in C:\Users\WRJ\PycharmProjects\PRD_FinalVersion
```

```
Ran 1 test in 11.152s
```

```
OK
```

**Figure 8** – *test decompFista*

## 1.9 test\_classifyOutsideParticlePETS

Permet de classer si les particules sont toujours dans la vue de la caméra ou non.

Pour tester cette fonction, je génère l'entrée de matlab dans un fichier texte et utilise la matrice stockée dans ce fichier comme l'entrée de python. Ensuite, je génère la sortie de matlab dans un fichier texte et compare la sortie de python avec le contenu de ce fichier texte.

L'entrée de matlab

```
scrCloudsClassify.txt
```

Le résultat de matlab

```
cloudsInClassify.txt
```

L'entrée de python

```
np.loadtxt('test_txt/scrCloudsClassify.txt')
```

Le résultat de python

```
np.loadtxt('test_txt/cloudsInClassify.txt')
```

Résultat de test :

```
Launching unittests with arguments python -m unittest Test.Test.test_classifyOutsideParticlePETS in C:\Users\WRJ\PycharmProjects\PRD_FinalVersion
```

```
Ran 1 test in 0.084s
```

```
OK
```

**Figure 9** – *test classifyOutsideParticlePETS*

## 1.10 test\_extractCandidates

Permet d'extraire un patch d'image avec la taille déterminée, qui est utilisé pour la représentation parcimonieuse.

Pour tester cette fonction, je génère l'entrée de matlab dans un fichier texte et utilise la matrice stockée dans ce fichier comme l'entrée de python. Ensuite, je génère la sortie de matlab dans un fichier texte et compare la sortie de python avec le contenu de ce fichier texte.

L'entrée de matlab

frmInput.txt

scrCloudsInput.txt

Le résultat de matlab

extractCandidatesOutput.txt

L'entrée de python

```
np.loadtxt('test_txt/frmInput.txt')
np.loadtxt('test_txt/scrCloudsInput.txt')
```

Le résultat de python

wings (un patch d'image)

Résultat de test :

```
Launching unittests with arguments python -m unittest Test.Test.test_extractCandidates in C:\Users\WRJ\PycharmProjects\PRD_FinalVersion
```

```
Ran 1 test in 0.677s
```

```
OK
```

Figure 10 – *test extractCandidates*

## 1.11 test\_fifo\_stack\_v2

Elle est utilisée dans la fonctions « SC\_ExtractingFeaturesPETS »

L'entrée de matlab

```
velocity = [0,0,0,0;0,0,0,0]
Veloc = [-1.036205146157517e+02
; 1.267660673840273e+02]
```

Le résultat de matlab

```
Velocity = [-103.6205,0,0,0
; 126.7661,0,0,0]
```

L'entrée de python

```
velocity = np.array([[0,0,0,0],[0,0,0,0]])
veloc = np.array([[-1.036205146157517e+02],[1.267660673840273e+02]])
```

Le résultat de python

```
velocityOutput = np.array([[-103.6205,0,0,0],[126.7661,0,0,0]])
```

Résultat de test :



```

Launching unittests with arguments python -m unittest Test.Test.test_fifo_stack_v2 in C:\Users\WRJ\PycharmProjects\PRD_FinalVersion

Ran 1 test in 0.001s

OK

```

Figure 11 – *test\_fifo\_stack\_v2*

### 1.12 test\_szPrePETS

Permet de calculer un paramètre qui est utilisé dans la fonction

« resamplingParticle\_szPredetermined\_PETS »

L'entrée de matlab

Des paramètres de tracker

Le résultat de matlab

2.5625

L'entrée de python

Les mêmes paramètres de Matlab

Le résultat de python

2.5625

Résultat de test :

```

Launching unittests with arguments python -m unittest Test.Test.test_szPrePETS in C:\Users\WRJ\PycharmProjects\PRD_FinalVersion

Ran 1 test in 0.004s

OK

```

Figure 12 – *test szPrePETS*

### 1.13 test\_bhattacharyya

Permet de calculer la distance de bhattacharyya

Pour tester cette fonction, je génère l'entrée de matlab dans un fichier texte et utilise la matrice stockée dans ce fichier comme l'entrée de python.

L'entrée de matlab

X1.txt

X2.txt

Le résultat de matlab

0.187932320222243

L'entrée de python

```

np.loadtxt('test_txt/X1.txt')
np.loadtxt('test_txt/X2.txt')

```

Le résultat de python

0.187932320222243

Résultat de test :

Launching unittests with arguments python -m unittest Test.Test.test\_bhattacharyya in C:\Users\WRJ\PycharmProjects\PRD\_FinalVersion

Ran 1 test in 0.071s

OK

**Figure 13 – test bhattacharyya**

## 1.14 test\_motionCheck

Permet de vérifier s'il existe un objet dans la boîte englobante via l'algorithme de soustraction d'arrière-plan.

Pour tester cette fonction, je génère l'entrée de matlab dans un fichier texte et utilise la matrice stockée dans ce fichier comme l'entrée de python.

L'entrée de matlab

subtr\_img.txt

Le résultat de matlab

1

L'entrée de python

```
np.loadtxt('test_txt/subtr_img.txt')
est = np.array([[665],[280],[2.5625],[0],[0],[5.125]])
```

Le résultat de python

1

Résultat de test :

Launching unittests with arguments python -m unittest Test.Test.test\_motionCheck in C:\Users\WRJ\PycharmProjects\PRD\_FinalVersion

Ran 1 test in 0.535s

OK

**Figure 14 – test motionCheck**

## 1.15 test\_updateDictionary

Permet de mettre à jour le dictionnaire

Pour tester cette fonction, je génère l'entrée de matlab dans un fichier texte et utilise la matrice stockée dans ce fichier comme l'entrée de python. Ensuite, je génère la sortie de matlab dans un fichier texte et compare la sortie de python avec le contenu de ce fichier texte.

L'entrée de matlab

wimgUpdateDictionary.txt

templatesUpdateDictionary.txt

BUpdateDictionary.txt

weightUpdateDictionary.txt

Le résultat de matlab

weightUpdateDictionaryResult.txt

L'entrée de python

```
wimg = np.loadtxt('test_txt/wimgUpdateDictionary.txt')
templates = np.loadtxt('test_txt/templatesUpdateDictionary.txt')
B = np.loadtxt('test_txt/BUpdateDictionary.txt')
weight = np.loadtxt('test_txt/weightUpdateDictionary.txt')
```

Le résultat de python

```
updateDictionary(tracker, sig_up, 0)
```

Résultat de test :

```
Launching unittests with arguments python -m unittest Test.Test.test_updateDictionary in C:\Users\WRJ\PycharmProjects\PRD_FinalVersion
```

```
Ran 1 test in 0.204s
```

```
OK
```

**Figure 15** – *test updateDictionary*

## 1.16 test\_cam2camGroundTrackPETS

Permet de calculer des particules. Elle est utilisée par la fonction «alterGroundTrackPETS»

Pour tester cette fonction, je génère l'entrée de matlab dans un fichier texte et utilise la matrice stockée dans ce fichier comme l'entrée de python. Ensuite, je génère la sortie de matlab dans un fichier texte et compare la sortie de python avec le contenu de ce fichier texte.

L'entrée de matlab

cloudsCam2camGroundTrackPETS.txt

Le résultat de matlab

cloudsCam2camGroundTrackPETSResult.txt

L'entrée de python

```
np.loadtxt('test_txt/cloudsCam2camGroundTrackPETS.txt')
```

Le résultat de python

```
camTocamGroundTrackPETS(cam1, cam8, cloudsCam2camGroundTrackPETS, opt)
```

Résultat de test :

Launching unittests with arguments python -m unittest Test.Test.test\_cam2camGroundTrackPETS in C:\Users\WRJ\PycharmProjects\PRD\_FinalVersion

Ran 1 test in 0.085s

OK

**Figure 16** – *test cam2camGroundTrackPETS*

## 2 Test Fonctionnel

Je compare les résultats des coordonnées de l'objet de python avec les résultats des coordonnées de l'objet de matlab.

Parce que les positions sont générées aléatoirement, il sera un peu de différence.

### 2.1 Le résultat des coins de frame 1

```
[[644.5 685.5 685.5 644.5 644.5]
 [239. 239. 321. 321. 239. ]]
```

**Figure 17** – *des coordonnées des coins de Python*

corners =

```
644.5000 685.5000 685.5000 644.5000 644.5000
239.0000 239.0000 321.0000 321.0000 239.0000
```

**Figure 18** – *des coordonnées des coins de Matlab*

### 2.2 Le résultat des coins de frame 2

```
[[644.25596615 685.01746454 685.01746454 644.25596615 644.25596615]
 [238.70087611 238.70087611 320.79396144 320.79396144 238.70087611]]
```

**Figure 19** – *des coordonnées des coins de Python*

corners =

```
644.8421 684.4679 684.4679 644.8421 644.8421
241.1057 241.1057 320.5245 320.5245 241.1057
```

**Figure 20** – *des coordonnées des coins de Matlab*



# Webographie

- [WWW1] *Camera Calibration*. URL : <https://fr.mathworks.com/help/vision/ug/camera-calibration.html?requestedDomain=www.mathworks.com> (visité le 05/12/2017).
- [WWW2] *Matplotlib*. URL : <https://fr.wikipedia.org/wiki/Matplotlib> (visité le 03/12/2018)■
- [WWW3] *OPENCV*. URL : <https://fr.wikipedia.org/wiki/OpenCV> (visité le 05/12/2017).
- [WWW4] *Raspberry Pi*. URL : [https://fr.wikipedia.org/wiki/Raspberry\\_Pi](https://fr.wikipedia.org/wiki/Raspberry_Pi) (visité le 05/12/2017).
- [WWW5] *RASPBIAN*. URL : <https://fr.wikipedia.org/wiki/Raspbian> (visité le 05/12/2017)■



## Bibliographie

- [1] N DE FREITAS, A DOUCET et N GORDON. « An introduction to sequential Monte Carlo methods ». In : *SMC Practice*. Springer Verlag (2001).
- [2] Quoccuong LE. « Object Tracking by Sparse Representations on Collaborative Cameras ». 2017.
- [3] Xue MEI et Haibin LING. « Robust visual tracking and vehicle classification via sparse representation ». In : *IEEE transactions on pattern analysis and machine intelligence* 33.11 (2011), p. 2259–2272.

# Projet Recherche et Développement polytech : Implémentation du algorithme de suivi

Rujia Wang

Encadrement : Quoccuong Le, Donatello Conte, Pierre Gaucher et Moncef Hidane

## Objectifs

Les objectifs sont:

- Implémenter un algorithme de suivi pour un réseau de cameras simple
- Chaque caméra sera contrôlée par une Raspberry pi
- Communication entre caméras



**POLYTECH**  
TOURS

## Mise en œuvre

Dans la partie Recherche, j'ai étudié l'algorithme de suivi avec plusieurs caméras ou avec une seule caméra. J'ai étudié les méthodes et les conceptions relatives.

Dans la partie Développement, J'aurai besoin d'implémenter le logiciel pour suivre un objet avec plusieurs caméras.



**POLYTECH**  
TOURS

## Résultats attendus

- Le logiciel suivi pour un réseau de cameras simple
- Communication entre caméras



**POLYTECH**  
TOURS



# Projet Recherche et Développement polytech : Implémentation du algorithme de suivi

Rujia Wang

Encadrement : Quoccuong Le, Donatello Conte, Pierre Gaucher et Moncef Hidane

## Objectifs

Les objectifs sont:

- Implémenter un algorithme de suivi pour un réseau de cameras simple
- Chaque caméra sera contrôlée par une Raspberry pi
- Communication entre caméras

## Mise en œuvre

Dans la partie Recherche, j'ai étudié l'algorithme de suivi avec plusieurs caméras ou avec une seule caméra. J'ai étudié les méthodes et les conceptions relatives. Dans la partie Développement, J'aurai besoin d'implémenter le logiciel pour suivre un objet avec plusieurs caméras.

## Résultats attendus

- Le logiciel suivi pour un réseau de cameras simple
- Communication entre caméras



POLYTECH<sup>®</sup>  
TOURS



POLYTECH<sup>®</sup>  
TOURS



POLYTECH<sup>®</sup>  
TOURS





# Projet Recherche et Développement polytech

## Implémentation du algorithme de suivi

### Résumé

Projet Recherche & Développement est très important et obligatoire pour obtenir le diplôme d'ingénieur.

L'objectif de ce projet est qu'implémenter un algorithme de suivi pour un réseau de cameras simple. On le réalise par utiliser le Raspberry Pi et le langage Python.

L'algorithme de suivi est basé sur des représentations parcimonieuses et le filtre particulaire.

### Mots-clés

algorithme de suivi, Raspberry Pi, Python, multi-caméra, représentations parcimonieuses, filtre particulaire

### Abstract

Project Research & Development is very important and necessary to obtain the engineering degree.

The objective of this project is to implement a tracking algorithm for a simple camera network. It is realized by using the Raspberry Pi and the Python.

The tracking algorithm is based on sparse representations and particle filter.

### Keywords

tracking algorithm, Raspberry Pi, Python, multi-camera, sparse representations, particle filter

### Tuteurs académiques

Quoccuong LE  
Donatello CONTE  
Pierre GAUCHER  
Moncef HIDANE

### Étudiant

Rujia WANG (DI5)