



Projet Tuteuré BOXi

RAMM TOUBA NICOLLE LEDEUX
en collaboration avec Two-i

INTRODUCTION



- Départ au Canada -> nécessité d'adapter le groupe et le calendrier
- Travail avec l'entreprise Two-i

Projet "BOX"

- Début du projet
- Organisation du groupe
- Problèmes rencontrés
- Étapes du développement
- Optimisation

PROBLÉMATIQUE ET CONTRAINTES

Problématique : proposer un service peu cher, simple d'utilisation , pratique et performant pour la détection de mouvements

Contraintes matérielles

- Raspberry pi 3 : puissance de calcul limitée
- Une solution pratique (doit pouvoir détecter les caméras une fois configurées)
- Un dispositif portatif et léger

Contraintes de performance

- Pouvoir analyser en direct le flux de plusieurs caméras
- Un minimum de 15 images par seconde
- Utilisation du langage C/C++ car plus rapide que Java

RECHERCHE

Les différentes solutions
HARDWARE

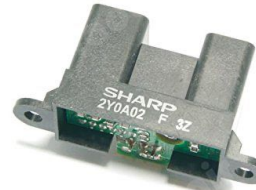
- FFM
- mot

Capteurs de Mouvement PIR



mandé car

ier
(C++)



Capteurs Infraréd



Capteurs Ultrasonore

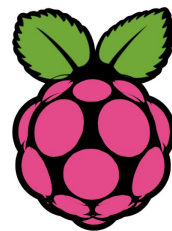
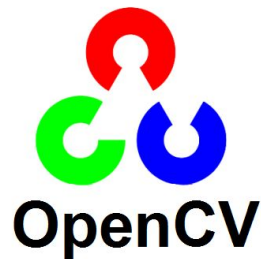
OUTILS UTILISÉS

Coté hardware :

- 2 Raspberry pi 3b +
- 2 Boitiers Trendnet
- 2 caméras IP (Foscam et Hikvision)
- 2 switch

Coté software :

- Librairie : OpenCV, fonctions et classes *VideoCapture*, *VideoWriter* et *Mat*
- Tests avec FFMPEG et LibVLC



Raspberry Pi

RÉPARTITION DU TRAVAIL

Arno

- Coordinateur (organisation, communication avec nos tuteurs, réunions et compte-rendus)
- Classe buffer

Manon

- Classe buffer en collaboration avec Arno
- Stockage, récupération des IP et ping

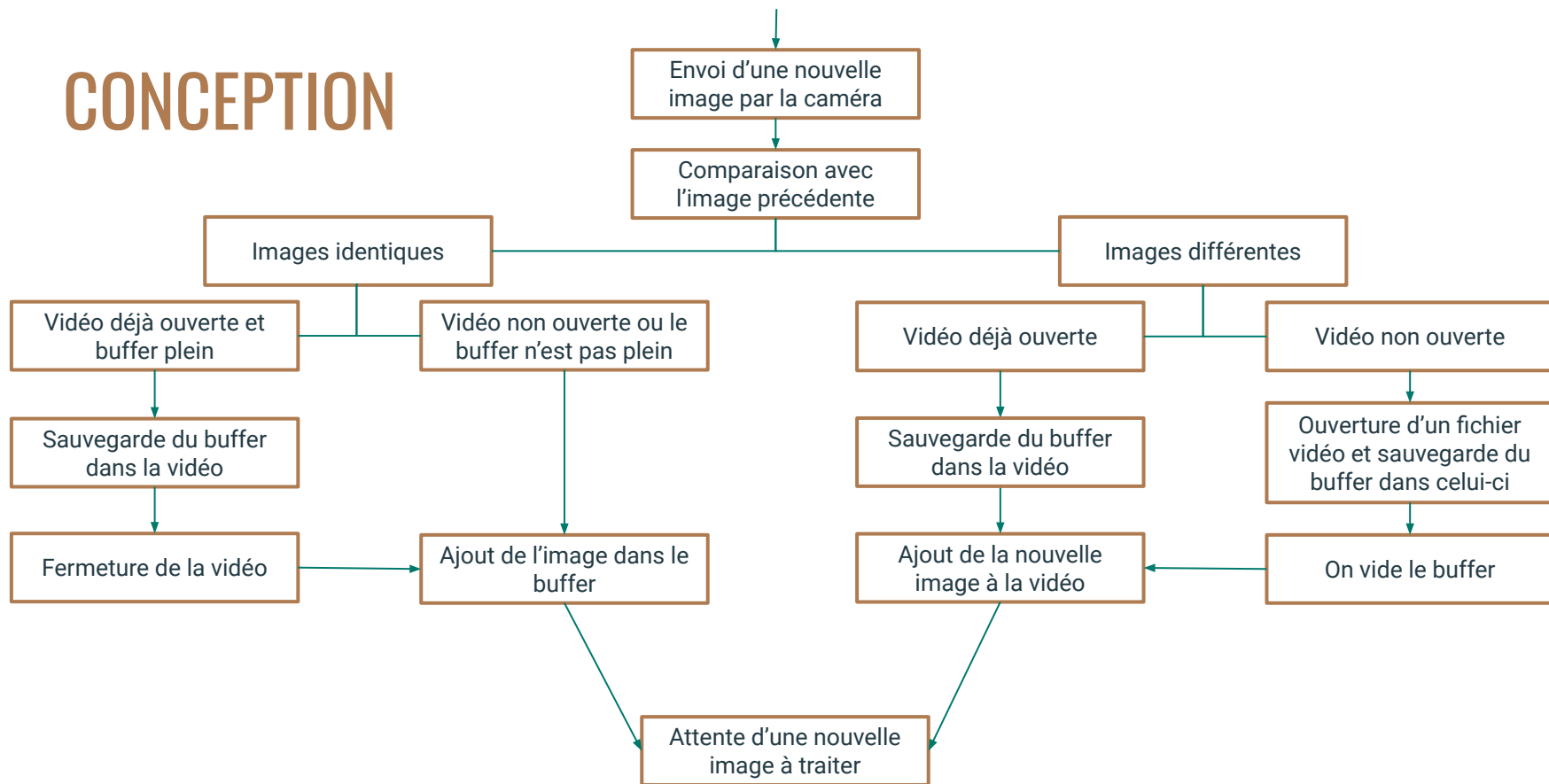
Flavien

- Détection du mouvement
- Multi threading
- Raspberry pi
- Optimisation

Xavier

- Conception et réalisation du poster
- Prototype des fonctions de récupération des IP

CONCEPTION



PRODUCTION

Requis :

1 caméra

Rapide et fluide



L'objectif :

3 caméras

Utiliser des classes

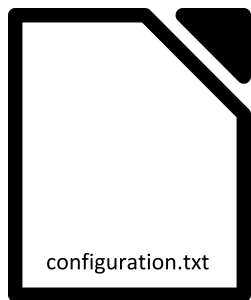
PRODUIT FINI

Multicaméra

Fichier d'ip

Écriture parallèle

Traitement : 1 images sur 4



Paramétrable

Algorithme de traitement
remplaçable

```
if (data->presenceMouvement())
```

POURSUITE DE PROJET

L'importance des fonctions OpenCV

Faire le propre algorithme

Le potentiel du compilateur g++

“PLUG AND PLAY ”

```
108 bool TraitementVideo::presenceMouvement(){
109     Mat image_diff, image_diff_gris, image_binaire;
110
111     //différence absolue des 2 images en input
112     absdiff(this->oldframe , this->newframe , image_diff);
113
114     //met l'image de différence en noir et blanc+ nuances de gris
115     cvtColor( image_diff, image_diff_gris, COLOR_BGR2GRAY );
116
117     //met l'image en noir et blanc en format binaire noir/blanc ajuster
118     threshold(image_diff_gris,image_binaire,this->seuil,255,0);
---
```

RETOUR SUR CETTE EXPÉRIENCE

Le côté technique

- Découverte du langage C++ et de ses librairies
- Gestion de plusieurs threads en parallèle

Tests et optimisation

- Ressources limitées sur Raspberry pi : nécessité de revoir notre code
- Nouveauté par rapport à nos projets précédents

Enjeux et contraintes des start-up

- Moyens limités, petite structure
- Nécessité de créer des projets rentables

Communication avec l'entreprise

- Préparation et compte-rendus des réunions
- Suivi du projet et communication par mail



Merci pour votre attention