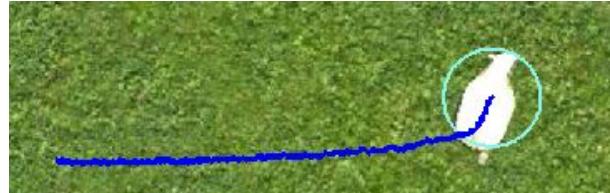


École Polytechnique de l'Université de Tours  
Spécialité Informatique  
64, Avenue Jean Portalis  
37200 TOURS, FRANCE  
Tél. +33 (0)2 47 36 14 14  
[www.polytech.univ-tours.fr](http://www.polytech.univ-tours.fr)

**Département Informatique**  
**5<sup>e</sup> année**  
**2014 - 2015**

**Rapport de projet fin d'étude**



**Étude comportement de l'animal (projet  
OVIN2A)**

**Encadrants**

Pascal Makris  
[pascal.makris@univ-tours.fr](mailto:pascal.makris@univ-tours.fr)  
Donatello Conte  
[donatello.conte@univ-tours.fr](mailto:donatello.conte@univ-tours.fr)

**Étudiants**

Lin SHEN  
[lin.shen@etu.univ-tours.fr](mailto:lin.shen@etu.univ-tours.fr)  
DI5 2014 - 2015



# Table des matières

---

<b>1 Remerciements</b>	<b>6</b>
<b>2 Introduction</b>	<b>7</b>
<b>3 Contexte du projet</b>	<b>8</b>
3.1 L'historique du projet . . . . .	8
3.1.1 Méthodes de base utilisées . . . . .	8
3.1.2 Algorithme de détection et suivie . . . . .	9
3.1.3 Interface du projet existant . . . . .	13
3.1.4 Problèmes du projet existant . . . . .	15
3.2 Les points à améliorer . . . . .	15
<b>4 Environnement de programmation</b>	<b>17</b>
4.1 Introduction du technique . . . . .	17
4.2 Configuration du environnement . . . . .	17
<b>5 Modélisation du projet</b>	<b>19</b>
5.1 Diagramme de classe . . . . .	19
5.2 Diagramme de séquence . . . . .	23
<b>6 Réalisation du projet</b>	<b>24</b>
6.1 Fonctionnalités réalisés . . . . .	24
6.1.1 Media Player . . . . .	24
6.2 Choisir objet . . . . .	26
6.3 Dialogue de déplacement max . . . . .	26
6.4 Dialogue d'information objet . . . . .	27
6.5 Enregistrer et ouvrir le fichier des coordonnées . . . . .	27
6.6 Emballage d'Exe . . . . .	28
6.7 Test . . . . .	29
6.8 Point à améliorer . . . . .	31
<b>7 Gestion de projet</b>	<b>32</b>
7.1 Méthodes de gestion de projet . . . . .	32
7.2 Outils de gestion de versions . . . . .	32
<b>8 Références</b>	<b>33</b>
<b>9 Conclusion</b>	<b>34</b>

# Table des figures

---

3.1	Pixel Addition . . . . .	8
3.2	Pixel Soustraction . . . . .	9
3.3	Choisit un objet vert . . . . .	10
3.4	Générer image binaire de première image de vidéo . . . . .	10
3.5	Générer image binaire de l'image courante de vidéo . . . . .	12
3.6	Image Addition . . . . .	12
3.7	Image Soustraction . . . . .	13
3.8	Erosion . . . . .	13
3.9	Interface de logiciel v1 . . . . .	14
3.10	Choisit l'objet de logiciel v1 . . . . .	14
3.11	Données de logiciel v1 . . . . .	15
3.12	Le design pattern MVC . . . . .	16
4.1	Configuration OpenCV . . . . .	18
4.2	Configuration OpenCV . . . . .	18
4.3	Configuration OpenCV . . . . .	18
5.1	Diagramme de classe . . . . .	19
5.2	Diagramme de classe de l'interface . . . . .	20
5.3	Diagramme de classe de la contrôleur . . . . .	21
5.4	Diagramme de classe du modèle . . . . .	22
5.5	Diagramme de séquence . . . . .	23
6.1	MainWindow . . . . .	25
6.2	MainWindow en modalité Visualiser . . . . .	25
6.3	Choisir l'objet . . . . .	26
6.4	Choisir l'objet . . . . .	26
6.5	Choisir l'objet . . . . .	27
6.6	Sauvegarder les informations des coordonnées . . . . .	28
6.7	Ouvrir les informations des coordonnées . . . . .	28
6.8	Dossier d'exe . . . . .	29
7.1	Git . . . . .	32

# Liste des tableaux

---

6.1	Test . . . . .	29
6.2	Test . . . . .	30

# **Remerciements**

---

Je tiens tout en premier lieu à remercier mon encadrant Monsieur Pascal Makris pour son aide précieuse tout au long de ce projet.

Je remercie aussi Monsieur Donatello Conte pour les conseils utiles pendant le projet.

Enfin je remercie aussi toutes les camarades pour m'aider de résoudre les différentes problèmes.

# Introduction

---

Dans le cadre de ma formation à Polytech tours, j'ai dû réaliser le projet de fin d'étude. Le projet fin d'étude a pour but de mettre en pratique mes compétences et c'est une conclusion de 3 ans. Le sujet est étude comportement de l'animal, plus précisément, c'est détecter et suivre les animaux dans la vidéo. Ce projet a été proposé par M. Pascal Makris, en collaboration avec M. Donatello Conte.

J'ai porté un grand intérêt à réaliser mon sujet qui est dans la domaine de traitement de l'image. Il y a une version existante. Mon travail consistait à améliorer la précédente version d'application en ajoutant certaine fonctionnalités et en mettant les codes sources plus propre.

Ce rapport est divisé par 5 grandes parties. Tout d'abord je vais vous présenter le contexte du projet, comprends la présentation du projet existant et les points à améliorer. Puis dans la seconde partie, je vous détaillerais l'environnement de programmation. Ensuite nous verrons la modélisation du projet. Enfin je vais vous présenter les étapes de réalisation de ce projet avec l'explication du code ainsi que la conclusion.

# Contexte du projet

---

Pour identifier les causes de mortalité des agneaux, il faut étudier leur comportements dans leur premières années. On a un caméra pour enregistrer les activités des agneaux. En utilisant notre logiciel, nous pouvons analyser les trajectoires des agneaux qui est différencié par la couleur. Un tel algorithme est déjà réalisé. Je vais apprendre les connaissances de traitement d'image et améliorer cette application. Par la suite, je vais vous présenter le projet existant et les améliorations.

## 3.1 L'historique du projet

### 3.1.1 Méthodes de base utilisées

Par la suite, je vais vous présenter certaines méthodes et théories de traitement d'images qu'on a utilisé dans ce projet. Vous pouvez les voir sur [Image Processing Operator Worksheets](#).

#### Pixel Addition

$\text{image} + \text{image}(\text{ou constante})$

Cet opérateur prend en entrée deux images de taille identique et produit en sortie une troisième image de la même taille que les deux premiers, dans lequel chaque valeur de pixel est la somme des valeurs de pixel correspondant de chacune des deux images d'entrée. Comme la figure suivante, on fait une addition des deux images de taille identique( $\text{image1}$  et  $\text{image2}$ ), on obtient  $\text{image3}$  de la même taille.

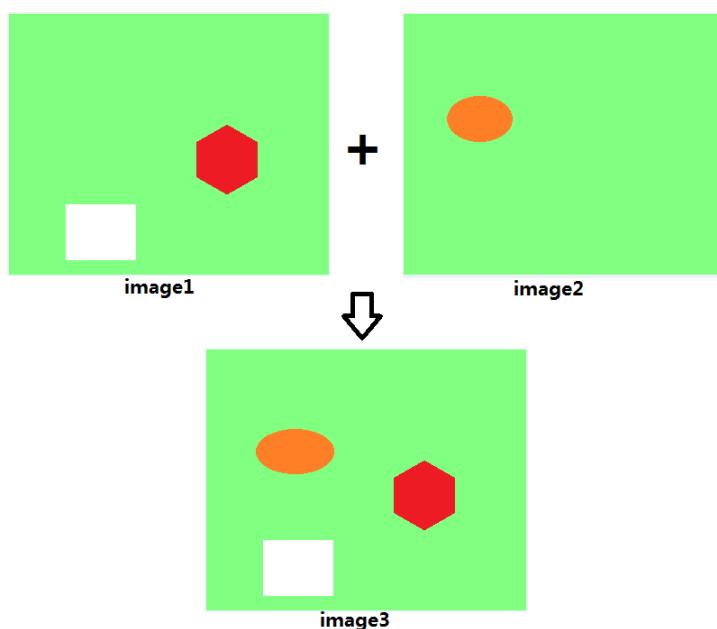


FIGURE 3.1 – Pixel Addition

## Pixel Soustraction

$\text{image} - \text{image}$

Cet opérateur prend deux images en entrée et produit en sortie une troisième image dont les valeurs de pixel sont ceux de la première image moins ceux de la deuxième image. Comme on peut voir la figure suivante, on fait la soustraction des deux images(image1 et image2), on obtient une troisième image.

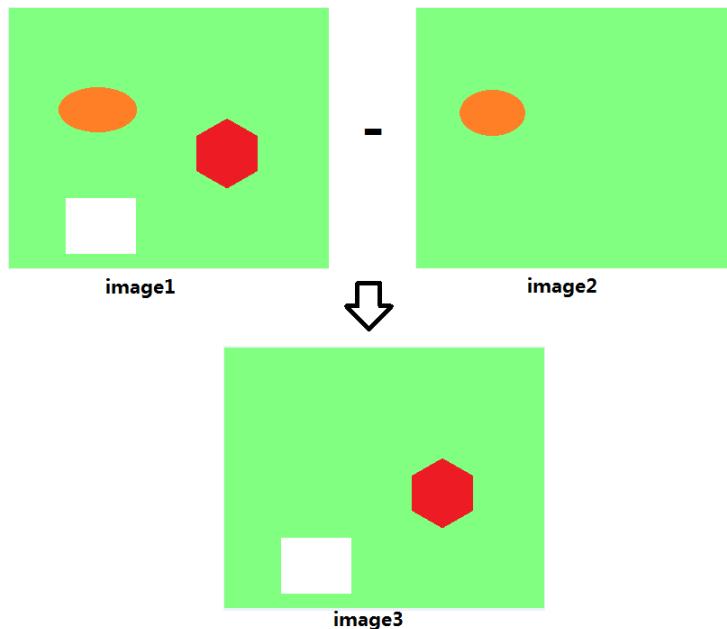


FIGURE 3.2 – Pixel Soustraction

## Thresholding–Seuillage

Le seuillage d'image est la méthode la plus simple de segmentation d'image. On sélectionne des pixels avec des valeurs données pour obtenir une image binaire.

Le seuillage d'image remplace un à un les pixels d'une image à l'aide d'une valeur seuil fixée (par exemple 123). Ainsi, si un pixel à une valeur supérieure au seuil (par exemple 150), il prendra la valeur 255 (blanc), et si sa valeur est inférieure (par exemple 100), il prendra la valeur 0 (noir).

## Morphologie–L'érosion

L'effet de base de l'opérateur sur une image binaire est à éroder les limites des régions de pixels foreground. les zones de pixels de foreground diminuer en taille, et des trous dans ces zones deviennent plus grandes.

### 3.1.2 Algorithme de détection et suivie

Par la suite, je vous présente l'algorithme de détection et suivie. Cette algorithme est proposé par l'étudiant de l'an dernier. On garde cette algorithme. D'abord, c'est la partie d'initialisation.

### Algorithm 1 Initialisation

- 1: calculer les couleurs de trois canaux(BGR) d'objet choisis
- 2: *binary\_fond* :binariser première image de vidéo

1 Par exemple, on choisit un objet vert. On obtient 6 valeurs de couleurs de cet objet. Ces couleurs sont la base de l'algorithme.

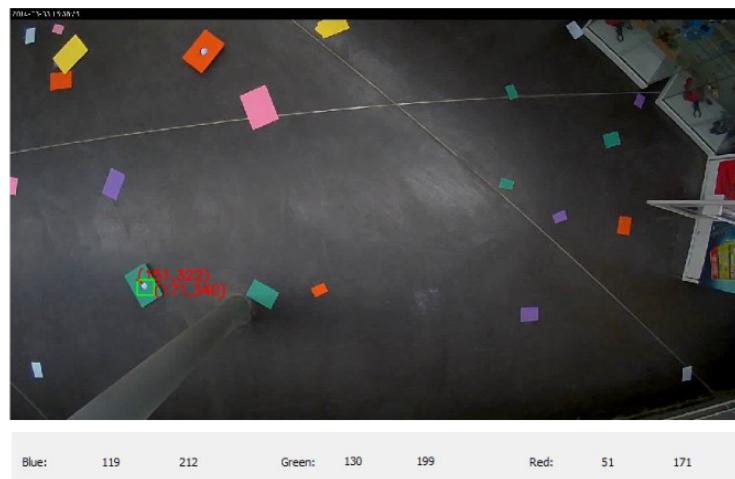


FIGURE 3.3 – Choisis un objet vert

2 Générer image binaire de première image de vidéo. L'objet qu'on choisit dans la vidéo sont vert, donc dans l'image binarisée, les parties blanches sont les objets vert, les autres parties deviennent noirs.

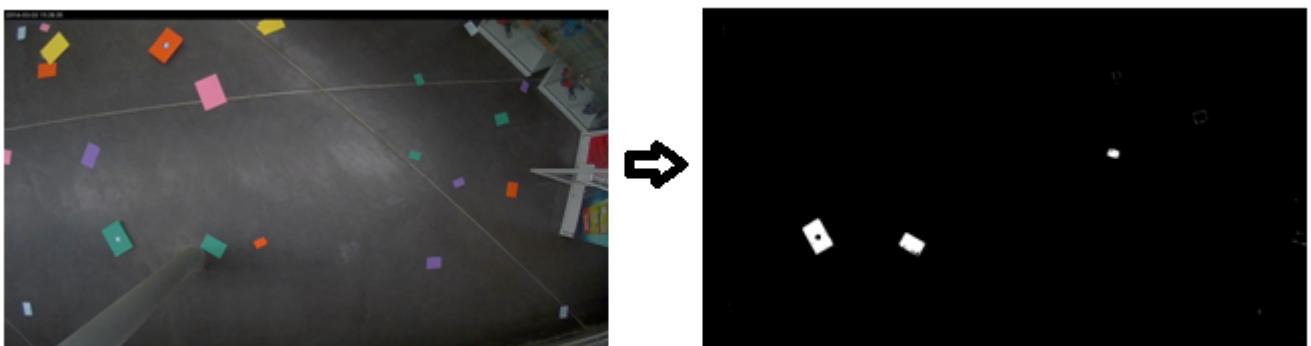


FIGURE 3.4 – Générer image binaire de première image de vidéo

L'algorithme suivant est l'algorithme principal de détection et suivie. L'entrée de l'algorithme est image de frame courant et numéro de frame courant. La sortie est une liste des points(x,y), la trajectoire de mouvement de l'objet.

D'abord je vous présente l'algorithme en générale, en suite je vais expliquer les détails.

---

**Algorithm 2** ALgorithme de détection et suivie

---

```

1: binary_frame : Générer image binaire d'image courante
2: Générer image qui contient seulement l'objet choisi dans le frame courant
   (binary_fond + binary_frame) – binary_fond = img_act(image de l'objet choisi)
3: clean_act : Faire l'érosion de image image_act
4: Rechercher les contours d'objet choisi
5: if (contours.size()>=1) then
6:   Localisez le center de objet et trouver le rayon d'objet
7:   if (trajectoire n'est pas vide) then
8:     if (les distances entre objet courant et objet de dernier frame < déplacement max) then
9:       Ajouter le point à trajectoire, modifier l'objet courant et dessiner le centre et le cercle
          d'objet
10:    end if
11:   else
12:     if pre_img == true then
13:       pre_img == false
14:       Ajouter le point à trajectoire
15:     else if Dans le dernier frame, l'objet déplace très rapidement then
16:       Re déterminer la distance et Ajouter le point à la trajectoire
17:     else
18:       if l'objet réapparaît then
19:         Ajouter le point à la trajectoire
20:       else
21:         Modifier l'image binaire
22:         count_refond++
23:       end if
24:     end if
25:   end if
26: else
27:   Vider la liste de trajectoire
28: end if
```

---

1 Générer l'image binaire du frame courant. L'objet choisi est objet vert. Donc les parties blanches sont les objets verts.

### Chapitre 3. Contexte du projet



FIGURE 3.5 – Générer image binaire de l'image courante de vidéo

**2**  $(\text{binary\_fond} + \text{binary\_frame}) - \text{binary\_fond} = \text{image de l'objet choisi}(\text{img\_act})$ .  
 La figure 3.6 est l'addition des images `bianry_fond` et `binary_frame`. On obtient l'image `binary_add`. La figure 3.7 est la soustraction des images `binary_add` et `binary_fond`.

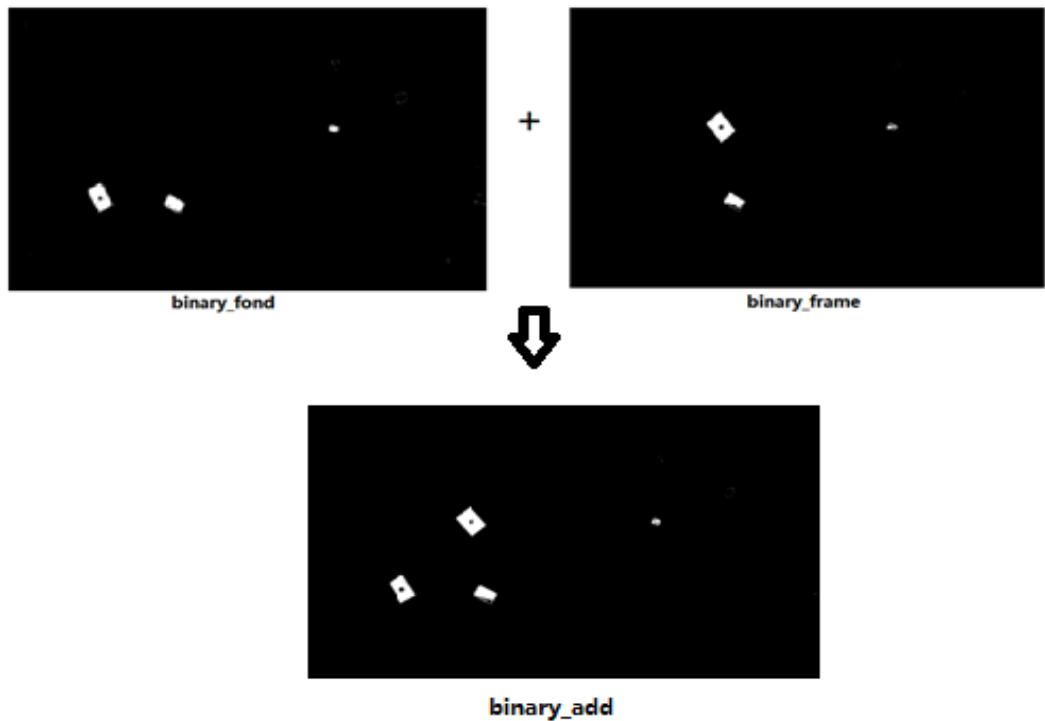


FIGURE 3.6 – Image Addition

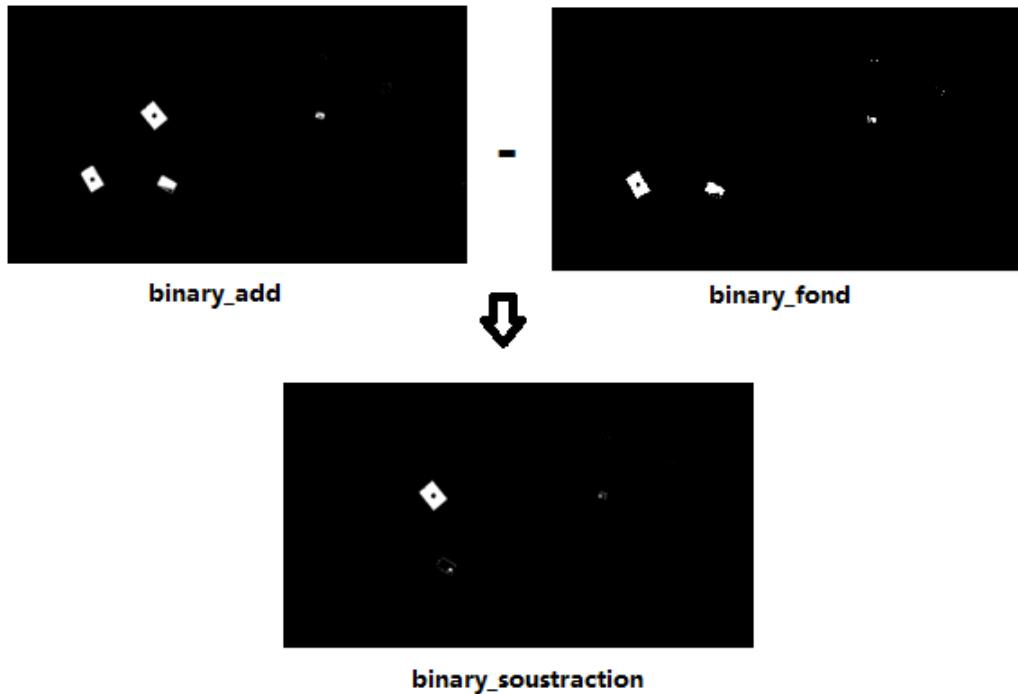


FIGURE 3.7 – Image Soustraction

**3** Après l'érosion, on obtient une image plus propre.

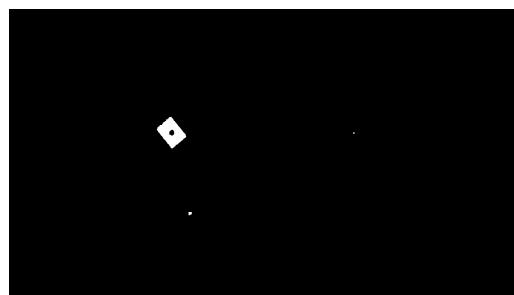


FIGURE 3.8 – Erosion

### 3.1.3 Interface du projet existant

Voici la figure ci-dessous, c'est l'interface du logiciel ancien.

Il comprends les fonctionnalités ci-dessous :

- Ouvrir la vidéo
- Saisir le numéro de frame qui comprends l'objet suivie et ouvrir l'image pour choisir l'objet, comme la figure 3.10
- Générer les histogrammes de trois canaux
- Définir la vitesse, le premier frame et le répertoire pour sauvegarder les données de trajectoire. Les données de trajectoire est comme la figure 3.11

### Chapitre 3. Contexte du projet

- Afficher la vidéo avec trajectoire et les couleurs d'objet suivie
- Arrêter la vidéo

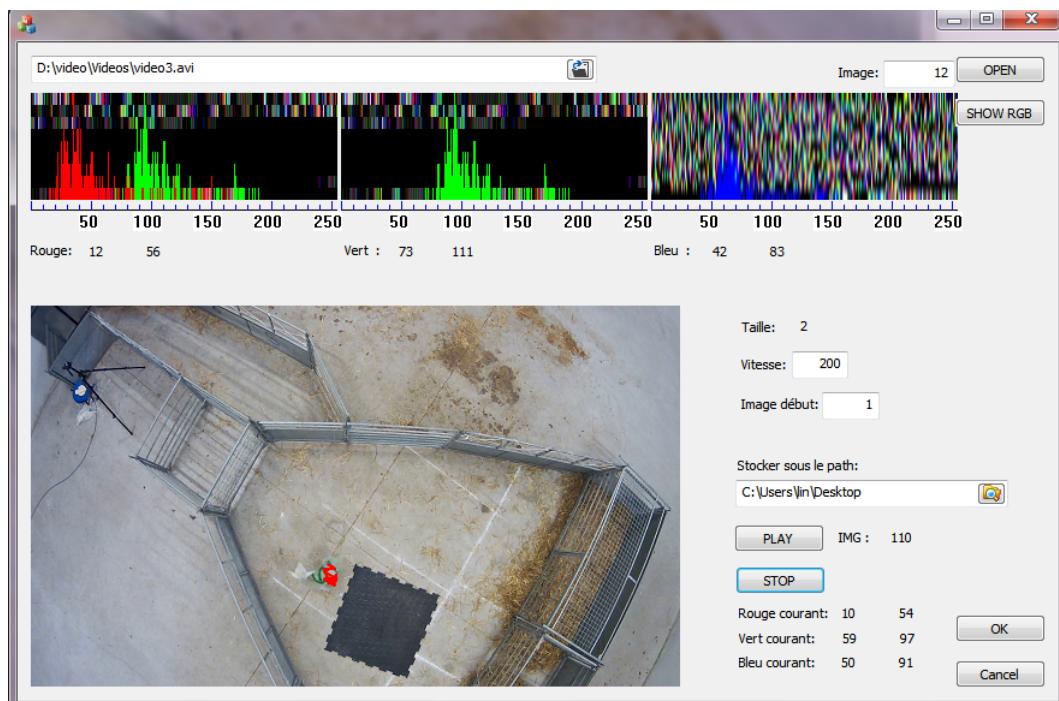


FIGURE 3.9 – Interface de logiciel v1

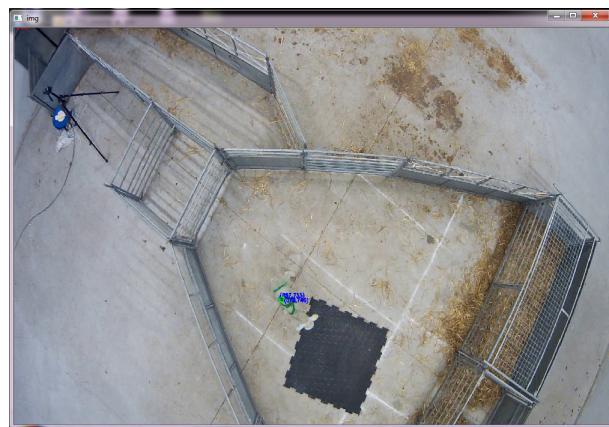


FIGURE 3.10 – Choisit l'objet de logiciel v1

14-04-2015-17h02m39s.txt - Bloc-notes			
Fichier	Édition	Format	Affichage ?
X: 890.500000	Y:744.750000	Tue Apr 14 17:02:39 2015	
X: 896.000000	Y:746.500000	Tue Apr 14 17:02:39 2015	
X: 893.000000	Y:740.000000	Tue Apr 14 17:02:40 2015	
X: 877.500000	Y:733.500000	Tue Apr 14 17:02:40 2015	
X: 899.500000	Y:746.000000	Tue Apr 14 17:02:40 2015	
X: 865.000000	Y:745.500000	Tue Apr 14 17:02:40 2015	
X: 879.714294	Y:748.000000	Tue Apr 14 17:02:40 2015	
X: 867.000000	Y:754.000000	Tue Apr 14 17:02:40 2015	
X: 901.000000	Y:772.277771	Tue Apr 14 17:02:40 2015	
X: 901.441162	Y:776.882324	Tue Apr 14 17:02:40 2015	
X: 900.500000	Y:777.000000	Tue Apr 14 17:02:40 2015	
X: 901.599976	Y:781.200012	Tue Apr 14 17:02:41 2015	
X: 890.500000	Y:787.500000	Tue Apr 14 17:02:41 2015	
X: 900.000000	Y:789.000000	Tue Apr 14 17:02:41 2015	
X: 899.500000	Y:790.423096	Tue Apr 14 17:02:41 2015	
X: 869.000000	Y:772.000000	Tue Apr 14 17:02:41 2015	

FIGURE 3.11 – Données de logiciel v1

### 3.1.4 Problèmes du projet existant

Il existe quelque problèmes sur le projet existant.

- Comme on veut voir sur la figure 3.9, les histogrammes ne sont pas clairs.
- Il faut saisir le numéro du frame pour ouvrir une image et choisir l'objet. Ce n'est pas facile à utiliser.
- On peut seulement jouer la vidéo. On ne peut pas la reculer et avancer.
- Quand on arrête la vidéo et rejouer, la vidéo recommence.
- Tous les codes sont dans la même fichier .cpp, il n'y a pas de structure. Ce n'est pas très facile à modifier.

## 3.2 Les points à améliorer

### L'amélioration 1

Mettre les codes sources en design pattern MVC(Modèle-Vue-Contrôleur). On crée une nouvelle structure pour mettre les codes plus propre et plus facile pour une optimisation supplémentaire, comme detecter 2 agneaux en même temps, changer l'algorithme de détection et suivie.

Le design pattern Modèle-Vue-Contrôleur (MVC), qui permettra de séparer distinctement interface graphique, les traitements et les évènements. Le modèle représente le cœur de l'application : traitements des données. La vue reçoit toute les actions de l'utilisateur et présente les résultats renvoyés par le modèle. Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de la vue et enclenche les actions à effectuer.

Voici la figure suivante. En résumé, lorsqu'un client envoie une requête à l'application :

- la requête envoyée depuis la vue est analysée par le contrôleur ;
- le contrôleur demande au modèle approprié d'effectuer les traitements et notifie à la vue que la requête est traitée ;
- la vue notifiée fait une requête au modèle pour se mettre à jour.

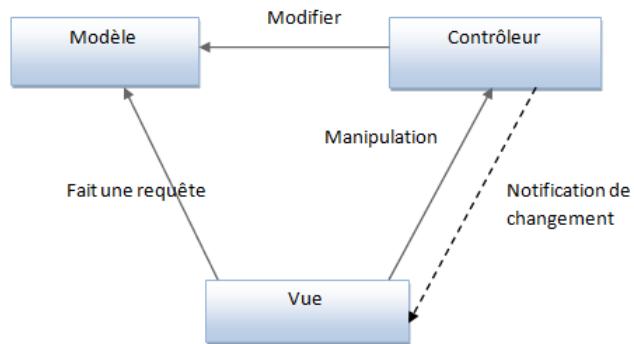


FIGURE 3.12 – Le design pattern MVC

### L'amélioration 2

Créer une nouvelle interface pour faciliter les utilisations. Il y a seulement une interface dans l'application existante. On peut mettre les histogrammes et les réglages des paramètres dans autres interface. On aussi ajoute une partie pour afficher en temps réel les données de trajectoire.

### L'amélioration 3

Ajoute la fonctionnalité pour reculer, avancer et suspendre la vidéo, comme le lecteur vidéo. C'est plus pratique pour choisir la vidéo et étudier la trajectoire.

### L'amélioration 4

Corriger les bugs dans les codes. Modifier les codes de générer les histogrammes. C'est une partie importante, pour générer la plage de couleur.

# Environnement de programmation

---

Langages de programmation que j'ai utilisé est C++ avec une bibliothèque graphique libre OpenCV(Open Source Computer Vision Library). L'outil de programmation est Qt. Par la suite, je vais vous présenter les outils utilisés et la configuration du environnement.

## 4.1 Introduction du technique

**OpenCV :**

OpenCV (Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. Elle propose nombreuses fonctionnalités de traitement d'images et traitement vidéos. Elle comprend les bibliothèques de C, C ++ et Python. Je utilise la partie de C++.

**Qt :**

Qt est un framework d'application de multi-plateforme qui est largement utilisé pour le développement de logiciels qui peut être exécuté sur différentes plates-formes logicielles et matérielles avec peu ou pas de changement dans la base de code sous-jacent.

**Dependency walker**

Dependency walker est un programme gratuit pour Microsoft Windows utilisé pour lister les fonctions importées et exportées d'un fichier exécutable portable. Je utilise ce logiciel pour examiner les dll manqués.

**FilePackager**

FilePackager permet de créer des fichiers Self-Extractor zip.

## 4.2 Configuration du environnement

Pour Qt, il y a plusieurs compilateurs. Je utilise la compilateurs de visual studio. Donc il faut installer visual studio d'abord.

1. Télécharger OpenCV(<http://opencv.org/downloads.html>). Je utilise la version 2.4.9.  
Télécharger Qt(<http://www.qt.io/download-open-source/>). Je utilise «Qt 5.3 for Windows 32-bit (VS 2013, 705 MB)».  
Télécharger visual studio 2013.
2. Installer visual studio.
3. Décompresser le fichier OpenCV dans la répertoire "C ://opencv". Configurer les variables d'environnement de OpenCV(Propriétés d'ordinateur->paramètres systèmes avancées->variables d'environnement). Ajouter une nouvelle variable d'environnement s'appelle OPENCV, comme la figure suivante. Modifier la variable d'environnement "Path", ajouter "%OPENCV%//x86/vc12/bin".
4. Installer Qt.

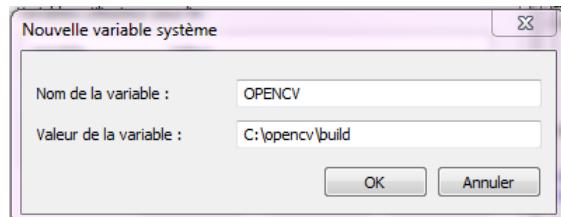


FIGURE 4.1 – Configuration OpenCV

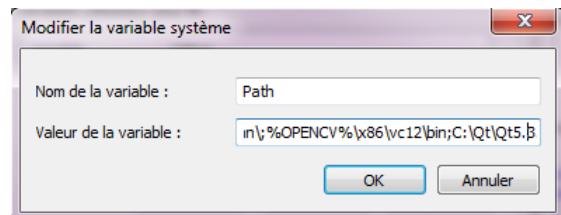


FIGURE 4.2 – Configuration OpenCV

- Ouvrir Qt et créer un nouveau projet «Application Qt4 en console». Dans le fichier .pro, ajouter la configuration de OpenCV comme la figure suivante.

```

44 INCLUDEPATH += C:/opencv/build/include
45 LIBPATH += C:/opencv/build/x86/vc12/lib\
46
47
48 CONFIG(release,debug|release){
49 CONFIG += embed_manifest_exe
50 QMAKE_LFLAGS_WINDOWS += /MANIFESTUAC:\"level='\`requireAdministrator\' uiAccess='false'\`"
51 LIBS += C:/opencv/build/x86/vc12/lib/opencv_calib3d249.lib
52 LIBS += C:/opencv/build/x86/vc12/lib/opencv_contrib249.lib
53 LIBS += C:/opencv/build/x86/vc12/lib/opencv_features2d249.lib
54 LIBS += C:/opencv/build/x86/vc12/lib/opencv_flann249.lib
55 LIBS += C:/opencv/build/x86/vc12/lib/opencv_highgui249.lib
56 LIBS += C:/opencv/build/x86/vc12/lib/opencv_imgproc249.lib
57 LIBS += C:/opencv/build/x86/vc12/lib/opencv_objdetect249.lib
58 LIBS += C:/opencv/build/x86/vc12/lib/opencv_video249.lib
59 LIBS += C:/opencv/build/x86/vc12/lib/opencv_core249.lib
60 }
61
62 CONFIG(debug, debug|release){
63 LIBS += C:/opencv/build/x86/vc12/lib/opencv_calib3d249d.lib
64 LIBS += C:/opencv/build/x86/vc12/lib/opencv_contrib249d.lib
65 LIBS += C:/opencv/build/x86/vc12/lib/opencv_features2d249d.lib
66 LIBS += C:/opencv/build/x86/vc12/lib/opencv_flann249d.lib
67 LIBS += C:/opencv/build/x86/vc12/lib/opencv_highgui249d.lib
68 LIBS += C:/opencv/build/x86/vc12/lib/opencv_imgproc249d.lib
69 LIBS += C:/opencv/build/x86/vc12/lib/opencv_objdetect249d.lib
70 LIBS += C:/opencv/build/x86/vc12/lib/opencv_video249d.lib
71 LIBS += C:/opencv/build/x86/vc12/lib/opencv_core249d.lib
72 }

```

FIGURE 4.3 – Configuration OpenCV

# Modélisation du projet

---

La modélisation est une partie importante pour ce projet. Dans le projet existant, la structure de code n'est pas propre. Ce n'est pas facile à l'améliorer. J'ai créé une nouvelle structure de code. J'ai divisé les codes à plusieurs classes.

## 5.1 Diagramme de classe

La figure suivante est le diagramme de classe en général. Les classes en bleu sont les classes de l'interface, comprends «MainWindow» et deux dialogues. Les classes en gris sont les classes de l'algorithme de détection et suivie. Les classes en rouge sont le contrôleur.

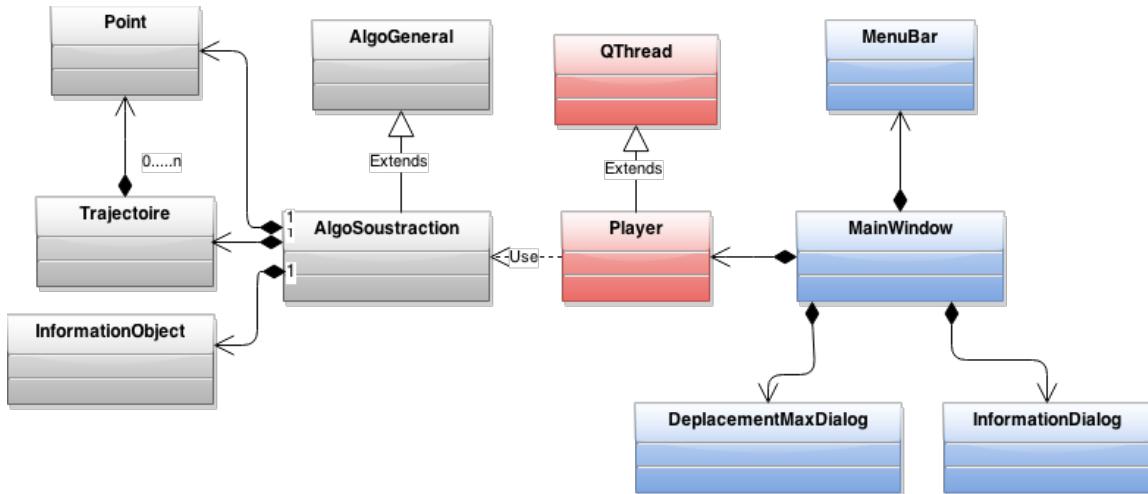


FIGURE 5.1 – Diagramme de classe

## Chapitre 5. Modélisation du projet

Dans les classes de l'interface, «`DeplacementMaxDialog`» est pour saisir la paramètre de l'algorithme. «`InformationDialog`» est pour afficher les histogrammes de couleur de l'objet choisi. «`MainWindow`» est pour afficher la vidéo, choisir l'objet à suivre, afficher la vidéo avec trajectoire et afficher les coordonnées.

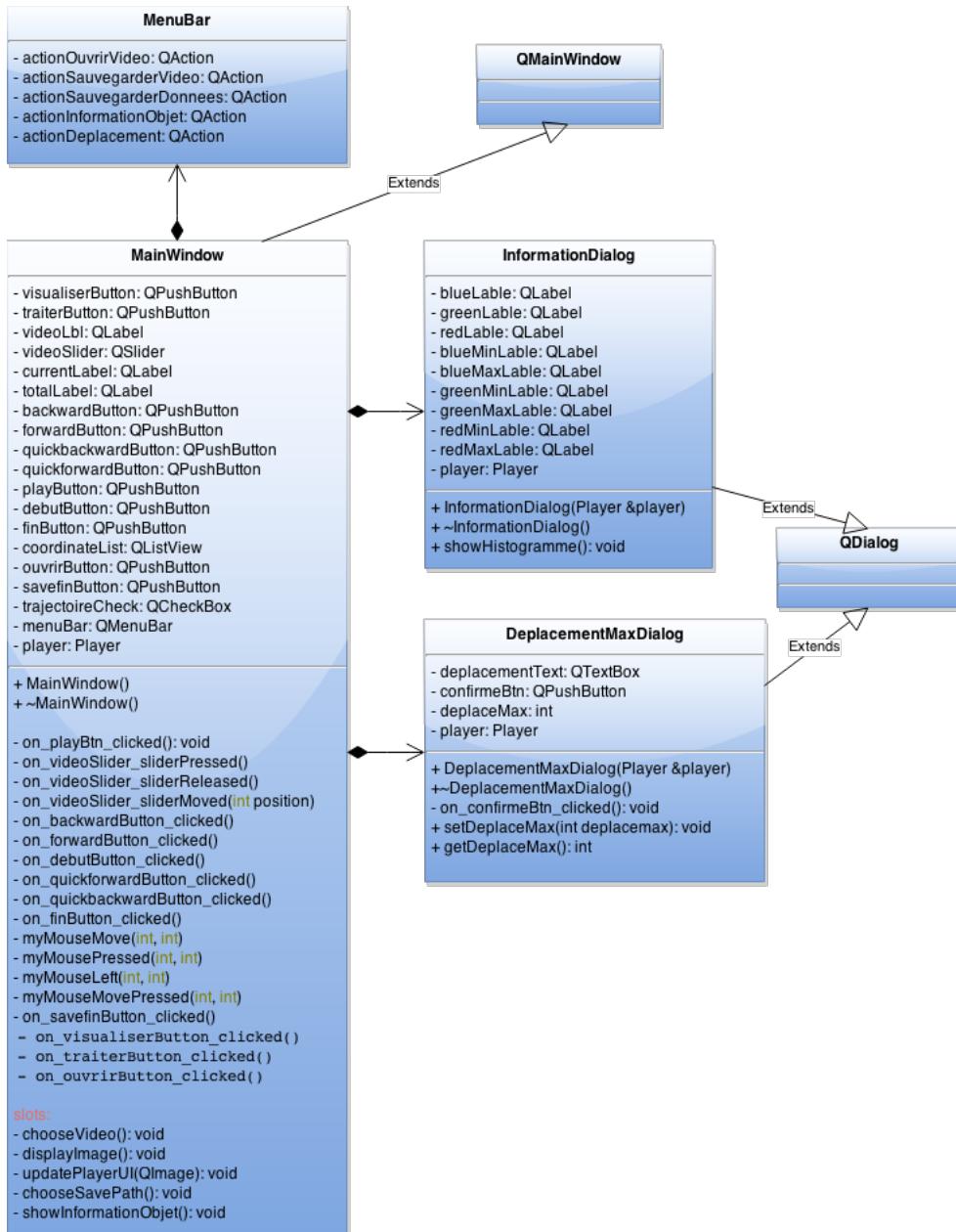


FIGURE 5.2 – Diagramme de classe de l'interface

Dans les classes en rouge, si dans la modalité "tratier", on récupère les frames de vidéo et donne les images à l'interface pour afficher la vidéo. Si dans la modalité "visualiser", on récupère les frames de vidéo et les donne à classe «AlgoSoustraction» pour calculer la trajectoire, après on donne l'image traitée et les informations des coordonnées à l'interface pour afficher.

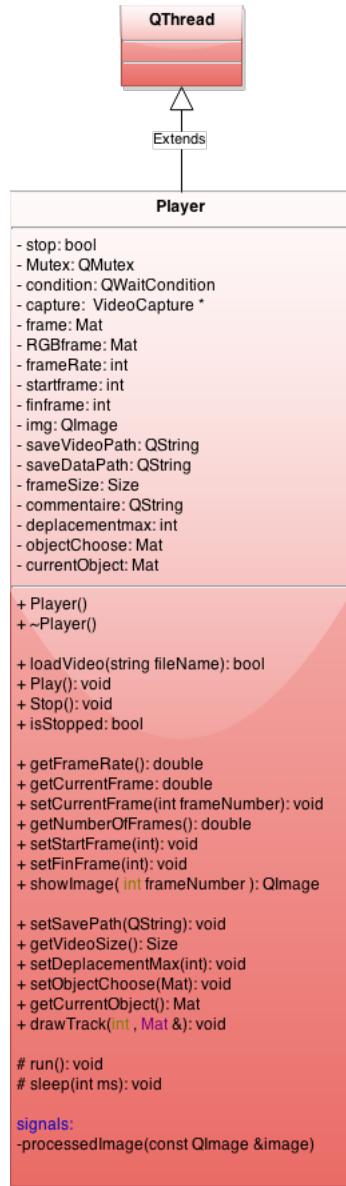


FIGURE 5.3 – Diagramme de classe de la contrôleur

## Chapitre 5. Modélisation du projet

Dans les classes du modèle, on a 5 classes.

La classe «Point» est les coordonnées du centre de l'objet dans chaque frame et l'heure correspondante.

La classe «Trajectoire» est l'ensemble des points. Si on veut suivre plusieurs agneaux, on peut initialiser plusieurs objets de cette classe.

La classe «InformationObject» est pour stocker les informations de l'objet, comme les coordonnées du centre, le rayon et les contours de l'objet. Dans la future amélioration, cette classe peut-être utilisable.

La classe «AlgoSoustraction» est l'algorithme de détection et suivie. Elle hérite la classe «AlgoGeneral». Si on veut change l'algorithme de détection et suivie, on peut facilement ajouter une nouvelle classe qui hérite «AlgoGeneral». Dans la classe «AlgoGeneral», la méthode «Detecter(Mat & currentFrame)» est la méthode virtuelle. Dans la classe fille, on peut utiliser cette méthode pour l'algorithme de détection et suivie. La méthode «calculeHistogram(Mat & src)» est pour générer les histogrammes de trois canaux. La méthode «GenerateBinaryImage(Mat &)» est pour générer l'image binarisée.

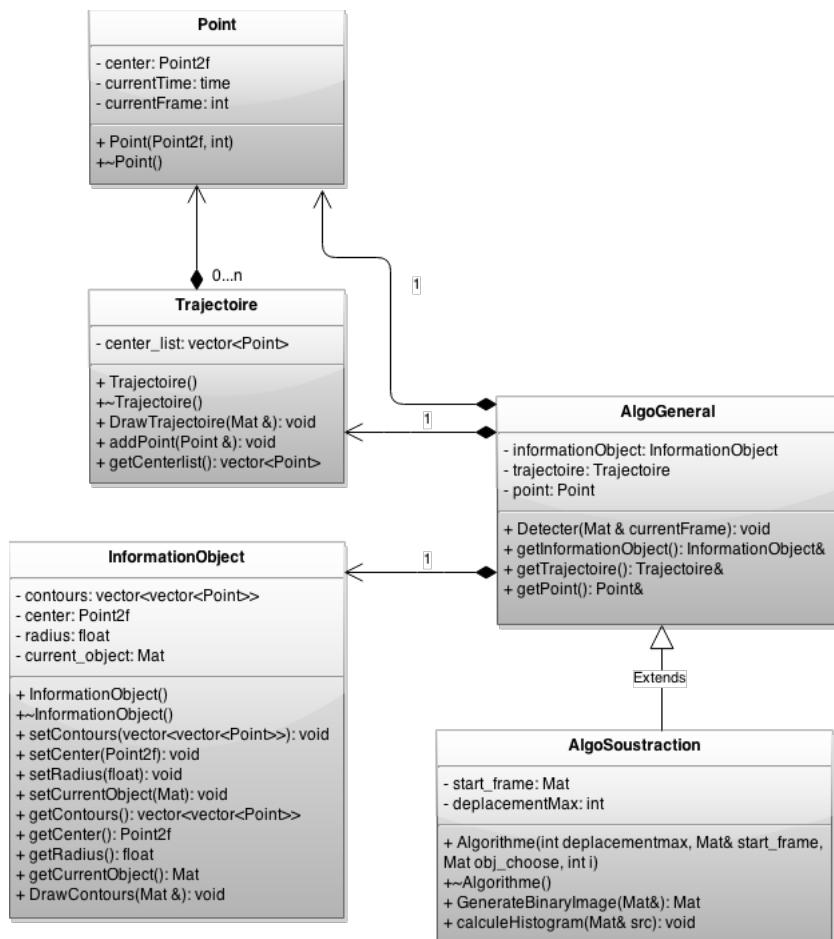


FIGURE 5.4 – Diagramme de classe du modèle

## 5.2 Diagramme de séquence

J'ai créé un diagramme de séquence. Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language. Ce diagramme de séquence explique la procédure pour afficher la vidéo avec trajctoire. D'abord, La classe «Player» appelle la méthode «Detecter(Mat&)» pour détecter l'objet choisi dans le frame courant. Et puis, la classe «AlgoSoustraction» ajouter le point dans la trajectoire. Ensuite, «Player» récupère la trajectoire et dessiner la trajectoire dans l'image courante. A la fin, «Player» retourne l'image traitée à «MainWindow».

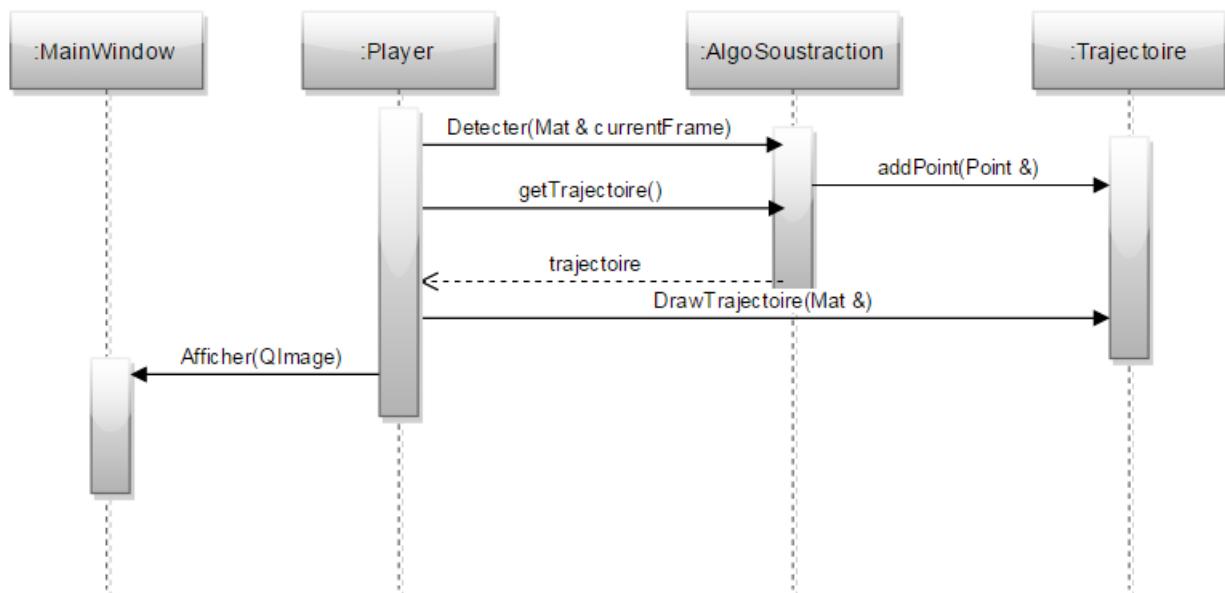


FIGURE 5.5 – Diagramme de séquence

# Réalisation du projet

---

Dans ce chapitre, je vais présenter les fonctionnalités réalisés et les tests. Il y a 2 modalités, une "Traiter", une autre "Visualiser".

La modalité "Traiter" est pour choisir l'objet à suivre et exécuter l'algorithme de détection et suivre. Après "Traiter", on a les informations de coordonnée et on peut les sauvegarder. Dans la modalités "Traiter", si on Checkbox "Trajectoire" n'est pas croché, on affiche la vidéo originale, sinon, on affiche la vidéo avec trajectoire. Quand on affiche la vidéo originale, on choisit une vidéo dans le répertoire et on peut avancer et reculer la vidéo , choisir l'objet à suivre et définir le début et la fin de vidéo. Après on croche Checkbox "Trajectoire", l'application va afficher la vidéo avec trajectoire et les coordonnées.

Dans la modalité "Visualiser", on peut choisir une vidéo et un fichier des coordonnées correspondant . Après, l'application va afficher la vidéo avec trajectoire. Et on peut avancer et reculer la vidéo traitée pour bien étudier les comportements des agneaux.

## 6.1 Fonctionnalités réalisés

### 6.1.1 Media Player

Le premier fonctionnalité qu'on réalise est Media Player. On peut afficher la vidéo frame par frame, ainsi qu'avancer, reculer et faire la pause. C'est pratique pour s'arrêter dans un frame.

#### La modalité "Traiter"

Dans la modalité "Traiter", au début, sur la bar de menu «Fichier», on peut choisir une vidéo dans le répertoire.

Ensuite, je vais expliquer les fonctionnalités des widgets en chiffre rouge.

1. Reculer la vidéo(10 frame)
2. Reculer la vidéo(1 frame)
3. Afficher le frame courant
4. Slider pour déplacer la vidéo
5. Afficher le frame total
6. Avancer la vidéo(10 frame)
7. Avancer la vidéo(1 frame)
8. Suspendre et redémarrer la vidéo.Quand la vidéo est arrêtée, le bouton est «Play», sinon, le bouton est «Stop»
9. Le bouton «Début» et «Fin» est pour définir le début et la fin de vidéo qu'on veut afficher après

Dans la modalité "Traiter", si on croche checkbox «Trajectoire», c'est-à-dire on veut suivre l'objet, on ne peut pas déplacer la vidéo lorsque la vidéo est en cours de lecture.

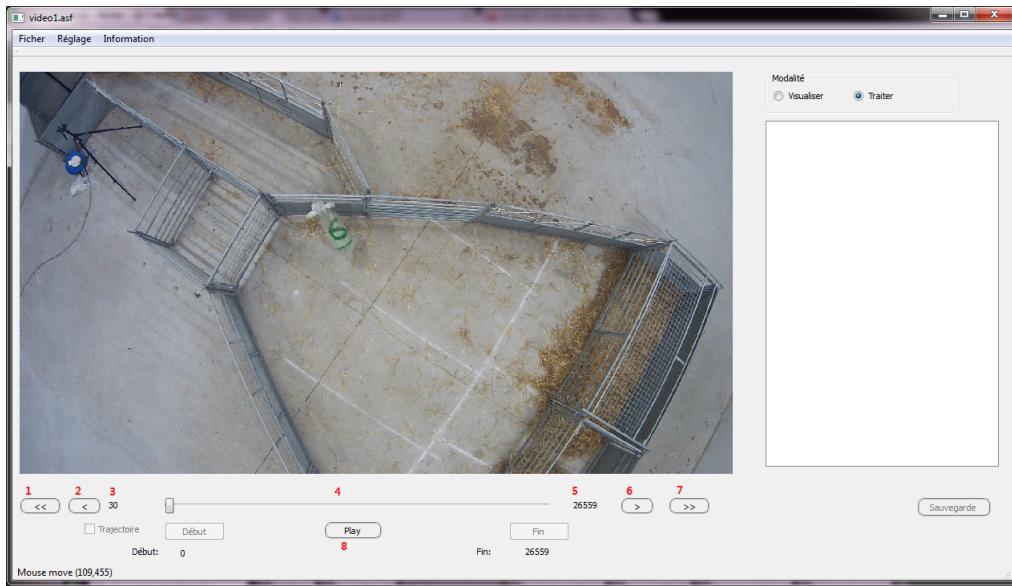


FIGURE 6.1 – MainWindow

### La modalité "Visualiser"

Dans la modalité "Visualiser", on peut aussi afficher, avancer et reculer la vidéo. La widget à droite permet d'afficher les informations de coordonnées du centre de l'objet. La première colonne est le nombre de frame, la deuxième et la troisième sont les coordonnées(x,y). L'affichage de vidéo et les coordonnées sont synchronisés. Par exemple, quand la vidéo affiche 29ème frame, on affiche les informations de coordonnée dans le 29ème frame. Quand on déplace la vidéo en utilisant les boutons, la vidéo et les informations de coordonnée sont synchronisées aussi. Par exemple, maintenant la vidéo est à 29ème frame, on la déplace à 25ème, on affiche les coordonnées de premier au 15ème frame dans ListView.

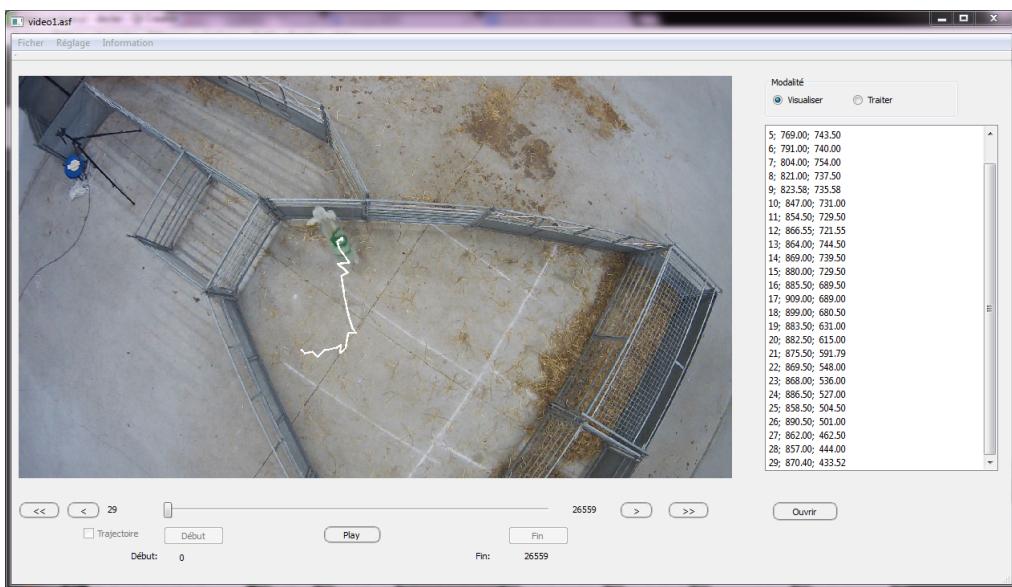


FIGURE 6.2 – MainWindow en modalité Visualiser

### 6.2 Choisir objet

Dans le projet existant, on saisit un nombre de frame et on ouvre une image dans autre fenêtre pour choisir l'objet.

Je modifie le moyen de choisir l'objet. On peut déplacer la vidéo dans un frame avec Slider ou bouton. Après on peut directement choisir l'objet dans MainWindow. Quand on déplace la souris sur VideoLabel, on affiche les coordonnées sur la barre de statut. Quand on appuie sur la souris, dans la fenêtre, elle affiche un point avec les coordonnées. Quand on appuie sur la souris pendant le déplacement, on peut dessiner un rectangle sur l'objet. Quand on laisse la souris, on dessine un point avec les coordonnées.

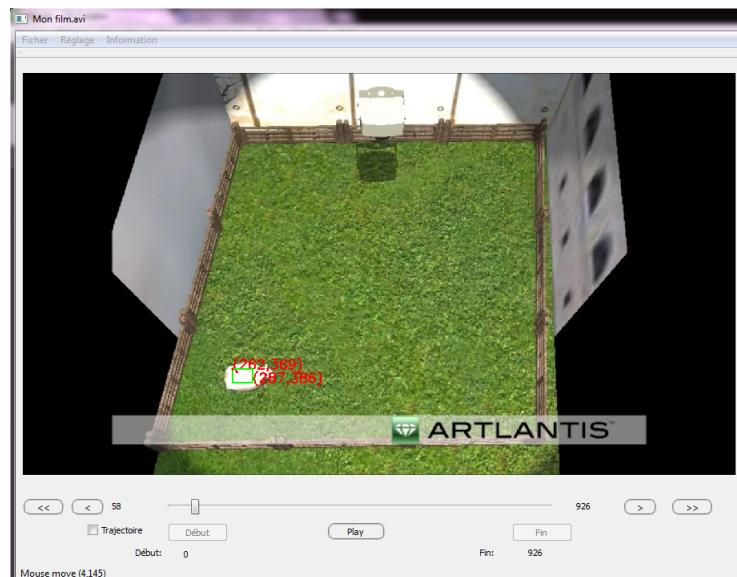


FIGURE 6.3 – Choisir l'objet

### 6.3 Dialogue de déplacement max

Dans le bar de menu «Réglage», on peut définir le paramètre de déplacement maximal. Ce paramètre est utilisé dans l'algorithme de détection et suivie pour distinguer les objets avec la couleur similaire.

Par exemple, on définit le déplacement maximal à 150 pixel et on choisit un objet vert à suivre. Si dans une image, on détecte plusieurs objets verts, on calcule la distance entre l'objet dans le frame courant et dans le frame dernier et prend l'objet dont la distance est inférieure à 150.

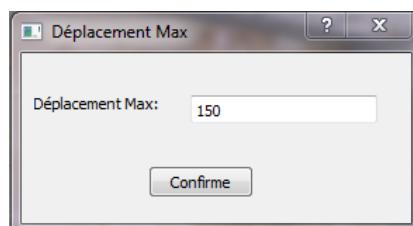


FIGURE 6.4 – Choisir l'objet

## 6.4 Dialogue d'information objet

Dans le bar de menu «Information», on peut ouvrir un dialogue pour afficher la couleur de l'objet choisi. C'est les histogrammes de trois canaux(Bleu, vert, rouge). L'abscisse de histogramme est l'intensité, de 0 à 255, l'ordonnée est les nombre de pixels de cette intensité.

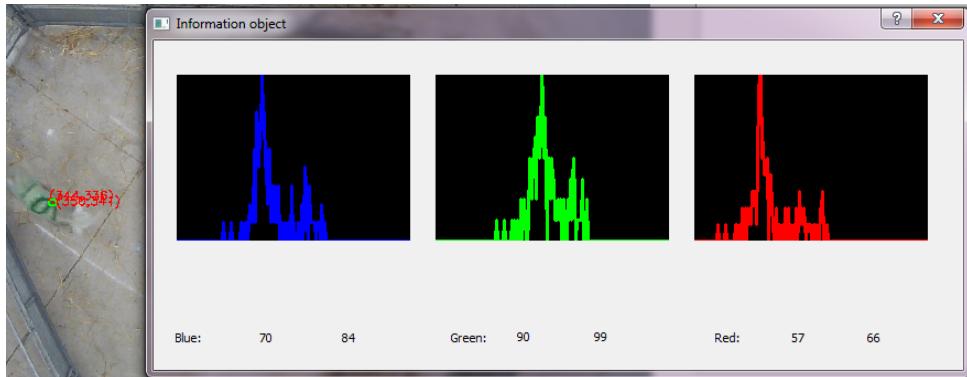


FIGURE 6.5 – Choisir l'objet

Les valeurs au dessous sont la limite minimale et la limite maximale de chaque couleur dans l'abscisse. Il y des formules pour calculer ces valeurs. On prend le canal rouge comme exemple. «thresh\_red\_1» est la valeur minimale sur l'abscisse, «thresh\_red\_2» est la valeur maximale sur l'abscise. «RedVal» est la valeur sur l'abscise dont valeur sur l'ordonnée est maximale.

---

**Algorithm 3** Calcule la limite minimale et maximale

```

1: thresh_red_1=(int)(thresh_red_1-15)
2: thresh_red_2=(int)(thresh_red_2+15)
3: if (thresh_red_2-RedVal)>3*(RedVal-thresh_red_1) then
4:   thresh_red_2=RedVal+3*(RedVal-thresh_red_1)
5: end if
6: if (RedVal-thresh_red_1)>3*(thresh_red_2-RedVal) then
7:   thresh_red_1=RedVal-3*(thresh_red_2-RedVal)
8: end if
9: if (thresh_red_1<=0) then
10:  thresh_red_1=1
11: end if
12: if (thresh_red_2>=256) then
13:  thresh_red_2=255
14: end if

```

---

## 6.5 Enregistrer et ouvrir le fichier des coordonnées

Dans la modalité "Traiter", après détection, on peut sauvegarder les informations des coordonnées si on les s'intéresse. Le répertoire par défaut est le dossier avec le même nom de la vidéo. Le type de fichier est csv.

## Chapitre 6. Réalisation du projet

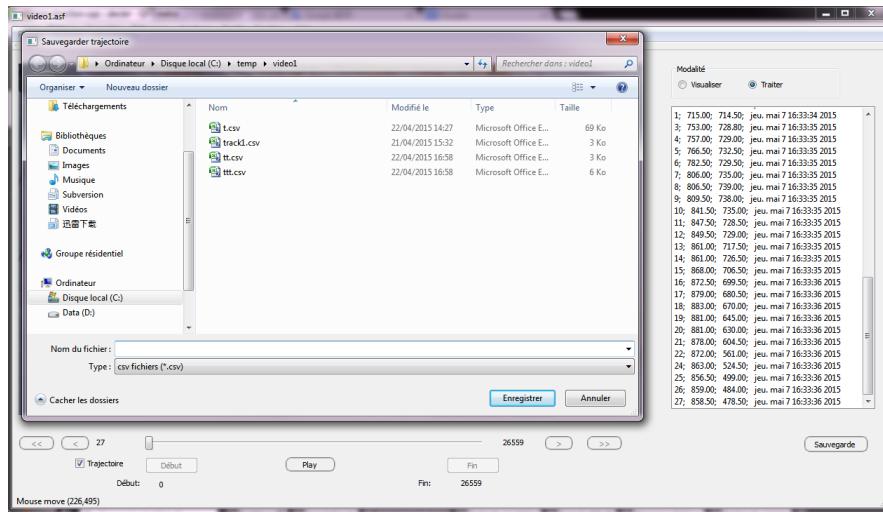


FIGURE 6.6 – Sauvegarder les informations des coordonnées

Dans la modalité "Visualiser", il faut ouvrir un fichier csv. Le répertoire par défaut est le dossier avec le même nom de la vidéo.

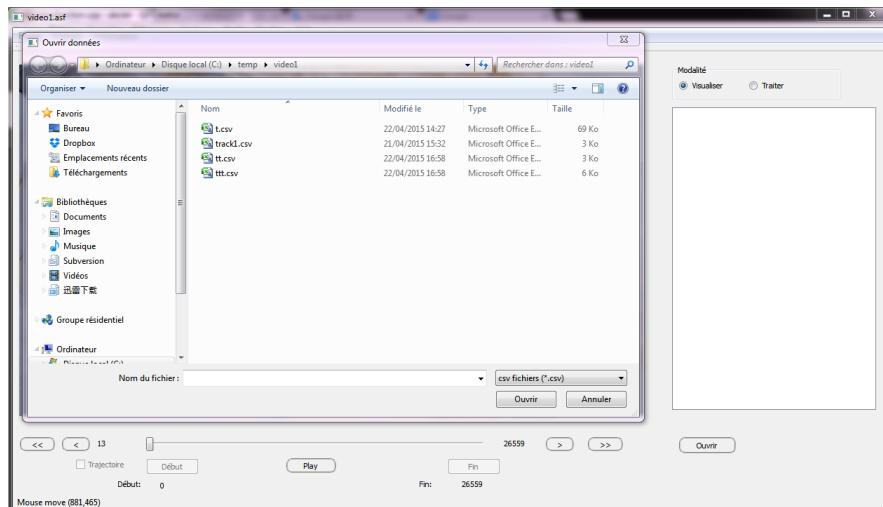


FIGURE 6.7 – Ouvrir les informations des coordonnées

## 6.6 Emballage d'Exe

On utilise "Dependency walker" pour examiner les dll manqués. On recherche ces dll dans notre répertoire de Qt et les ajoute dans notre dossier.

On ajoute la dossier «platforms» avec qwindows.dll qui se trouve dans « Répertoire d'installation de Qt/Qt5.3.2/5.3/msvc2013/plugins/platforms». On ajoute «msvcp100.dll» et «msvcp120.dll». Après ajouter toutes les dll, on a le dossier ci-dessous.

A la fin, on utilise «FilePacker» pour emballage d'Exe.

 decter.exe	21/04/2015 17:29	Application	153 Ko
 decter1.exe	13/04/2015 09:29	Application	159 Ko
 icudt52.dll	13/04/2015 13:48	Extension de l'app...	10 347 Ko
 icuin52.dll	13/04/2015 13:45	Extension de l'app...	999 Ko
 icuuc52.dll	13/04/2015 13:46	Extension de l'app...	804 Ko
 libEGL.dll	13/04/2015 14:53	Extension de l'app...	53 Ko
 libGLESv2.dll	13/04/2015 13:43	Extension de l'app...	633 Ko
 mozglue.dll	13/04/2015 13:45	Extension de l'app...	128 Ko
 mstcp100.dll	18/03/2010 08:15	Extension de l'app...	412 Ko
 mstcp120.dll	05/10/2013 02:38	Extension de l'app...	445 Ko
 msocr100.dll	18/03/2010 08:15	Extension de l'app...	753 Ko
 msocr120.dll	05/10/2013 02:38	Extension de l'app...	949 Ko
 opencv_core249.dll	15/04/2014 11:14	Extension de l'app...	2 086 Ko
 opencv_ffmpeg249.dll	20/12/2013 11:32	Extension de l'app...	10 289 Ko
 opencv_highgui249.dll	15/04/2014 11:15	Extension de l'app...	2 097 Ko
 opencv_imgproc249.dll	15/04/2014 11:14	Extension de l'app...	1 833 Ko
 Qt5Core.dll	16/10/2014 09:07	Extension de l'app...	3 987 Ko
 Qt5Gui.dll	11/09/2014 13:41	Extension de l'app...	3 158 Ko
 Qt5Widgets.dll	11/09/2014 13:44	Extension de l'app...	4 196 Ko

FIGURE 6.8 – Dossier d'exe

## 6.7 Test

Nom de la vidéo	Format	Durée	Taille	Environnement	Objet à suivre	Résultat
Mouton.avi	avi	30s	8515ko	bergerie verte : un seule mouton	mouton	Très bien
Mouton.mp4	mp4	30s	9.9Mo	bergerie verte : un seule mouton	mouton	Très bien
Robot.asf	asf	6m1s	544Mo	Hall du DI : plusieurs robot carré en couleur différente, quelques passants, couleur d'interférence	un robot vert	Moyen
Robot.asf	asf	6m1s	544Mo	Hall du DI : plusieurs robot carré en couleur différente, quelques passants, couleur d'interférence	un robot orange	Bien
Robot.avi	avi	6m1s	178Mo	Hall du DI : plusieurs robot carré en couleur différente, quelques passants, couleur d'interférence	un robot vert	Moyen
Robot.avi	avi	6m1s	178Mo	Hall du DI : plusieurs robot carré en couleur différente, quelques passants, couleur d'interférence	un robot orange	Bien
birds.mp4	mp4	5m37s	23,1Mo	Nature, environnement qui change : plusieurs oiseaux	un oiseau	Mal

TABLE 6.1 – Test

## Chapitre 6. Réalisation du projet

Nom de la vidéo	Format	Durée	Taille	Environnement	Objet à suivre	Résultat
video1.asf	ASF	4m25s	938Mo	bergerie gris : un mouton gris avec la peinture verte en dos, mouvement rapide, quelques passants	mouton	Mal
video1.avi	AVI	4m25s	472Mo	bergerie gris : un mouton gris avec la peinture verte en dos, mouvement rapide, quelques passants	mouton	Moyen
video2.asf	ASF	4m04s	875Mo	bergerie gris : un mouton gris avec la peinture verte en dos	mouton	moyen
video2.avi	AVI	4m25s	434Mo	bergerie gris : un mouton gris avec la peinture verte en dos	mouton	bien
video3.asf	ASF	1m44s	388Mo	bergerie gris avec un tapis noir : un mouton gris avec la peinture verte en dos, mouvement lent	mouton	moyen
video3.avi	AVI	1m44s	185Mo	bergerie gris avec un tapis noir : un mouton gris avec la peinture verte en dos, mouvement lent	mouton	bien

TABLE 6.2 – Test

Parmi ces vidéo, les 6 dernières vidéos sont la vidéo réelle.

- **Format de la vidéo :** Pour la vidéo du format asf, elle parfois s'arrête brusquement. Pour la vidéo du format avi et mp4, ça fonctionne bien.
- **Durée de la vidéo :** La durée n'est pas un problème.
- **Taille de la vidéo :** La taille de la vidéo affecte la vitesse de lecture.
- **Environnement :** Le fond de la vidéo est important pour détection. Lorsque les modifications de fond, cette algorithme de détection ne marche pas, comme la vidéo «birds.mp4». Mais pour ce projet, la caméra est fixe et le fond est fixe. Donc ça ne pose pas de problème.  
Si la couleur de l'objet à suivre et la couleur de fond sont similaires, ça ne marche pas bien. Pour les 6 dernières vidéos, l'agneau est petit et la couleur est gris comme la couleur de fond. C'est difficile de bien choisir la couleur verte sur le dos du mouton. Si on ne peut pas bien choisir la zone de vert, on ne peut pas bien suivre le mouton.
- **Objet à suivre :** Si le mouton déplace très rapidement, parfois, on ne peut pas bien détecter.

## 6.8 Point à améliorer

- Faire un zoom out pour bien choisir le petit objet.
- Maintenant l'algorithme dépend du paramètre «déplacement maximal». On pourra améliorer l'algorithme de détection et suivie sans ce paramètre.
- On pourra améliorer l'algorithme pour s'adapter le changement du fond.
- Pour la vidéo asf, la vidéo parfois s'arrête brusquement, je ne trouve pas la solution pour résoudre ce problème.
- Améliorer l'efficacité de détection.

# Gestion de projet

---

## 7.1 Méthodes de gestion de projet

J'utilise la méthode traditionnelle du cycle en V qui a des phases de conception, développements, tests et intégration. Elle limite un retour aux étapes précédentes. Parce que c'est un projet existant, les besoins sont clairs. D'abord j'ai étudié les notions et le projet existant pour bien comprendre la besoin. Et puis j'ai fait la spécification. En suite j'ai fait la modélisation et réalisation. Dans la phase de réalisation, j'ai codé module par module. J'ai fait le test pour chaque module avant j'ai passé au module suivant. A la fin, j'ai écrit toutes les documentations.

## 7.2 Outils de gestion de versions

J'utilise Git pour la gestion de versions. Git est un logiciel de gestion de versions décentralisé. C'est un outil libre, simple et performant. La figure suivante est la contribution sur Git.

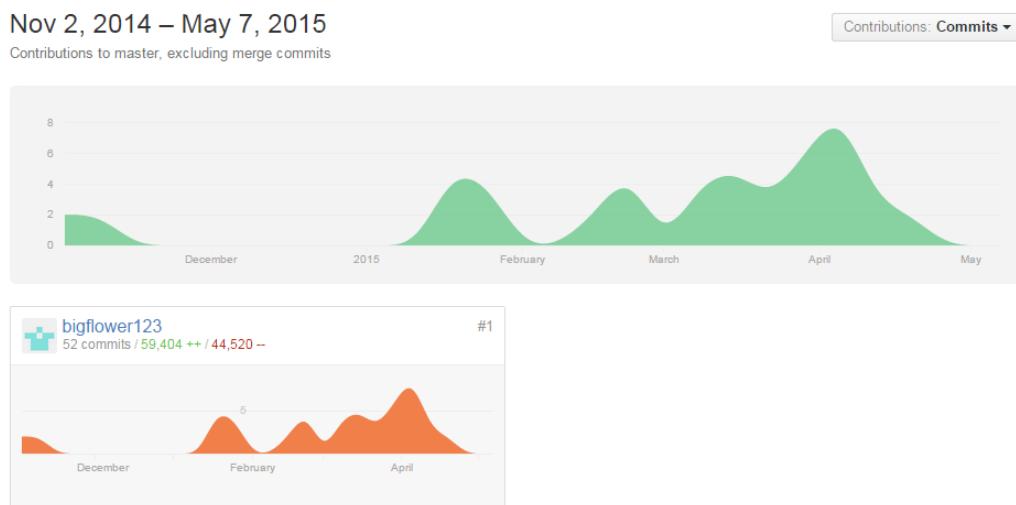


FIGURE 7.1 – Git

# Références

---

1. Opencv documentation  
<http://docs.opencv.org/master/index.html>
2. The Hypermedia Image Processing Reference  
[http://homepages.inf.ed.ac.uk/rbf/HIPR2/hipr\\_top.htm](http://homepages.inf.ed.ac.uk/rbf/HIPR2/hipr_top.htm)
3. XingXU, Rapport et codes de Projet de Fin d'Etude 2013/2014, Détection et suivi d'animaux dans les vidéos

# Conclusion

---

Pendant 7 mois, j'ai fait un projet de fin d'étude, étude comportement de l'animal. Plus précisément, l'objectif est l'amélioration d'un projet existant pour détection et suivre l'agneau dans la vidéo.

Ce projet m'a permis d'approfondir mes connaissances dans les domaines de traitement de l'image. Au début, j'ai consacré beaucoup de temps à comprendre le projet existant. Tout au long du projet, M.Mascal et M.Conte m'a donné beaucoup de conseils utiles. Pour la fonctionnalité de sauvegarde, j'ai enregistré la vidéo avec trajectoire. Mais ça prend beaucoup d'espace. Après la modification, j'ai enregistré seulement le fichier csv. Dans la modalité "Visualiser", j'ai dessiné la trajectoire en temps réel. C'est plus pratique.

Grâce aux tests avec plusieurs vidéos, il toujours reste quelques problèmes à améliorer.

Dans la vidéo réelle, le mouton est très petit et c'est difficile de choisir la zone verte sur le dos. Dans ce cas, on pourra ajouter un fonctionnalité de zoom in pour la zone du mouton.

De plus, l'algorithme dépend du paramètre «déplacement maximal». Ce n'est pas facile à définir. On pourra améliorer ce algorithme sans utiliser ce paramètre.

Si le fond de la vidéo change, on ne peut pas bien détecter l'objet. Ce problème doit encore être discutés. Mais pour notre projet, la caméra est fixe, donc le fond de la vidéo est toujours fixe. Ce algorithme ne pose pas de problème.

Pour la vidéo asf, la vidéo parfois s'arrête brusquement, je ne trouve pas la solution pour résoudre ce problème.

Ce projet a donc été pour moi une expérience autant agréable qu'enrichissante au cours de laquelle j'ai appris la gestion de projet et le un usage de Opencv/C++ et Qt. J'ai aussi appris les connaissances sur le traitement d'image numérique.



# Étude comportement de l'animal(projet OVIN2A)

---

Département Informatique  
5<sup>e</sup> année  
2014 - 2015

Rapport de projet fin d'étude

**Résumé :** Mon sujet de projet de fin d'étude fut de créer une application de détection et suivie l'animal dans la vidéo qui est basé sur un projet existant. Plus précisément mon travail consistait à améliorer l'interface et créer une nouvelle structure de code en gardant l'algorithme de détection. J'ai appris les connaissances sur le traitement d'image numérique.

**Mots clefs :** Détection et suivie, Opencv, Qt

**Abstract:** My subject of graduation Project was to create an application to detect and follow the animal in the video which is based on an existing project. Specifically my job was to improve the interface and create a new code structure by keeping the algorithm. I learned the knowledge of digital image processing.

**Keywords:** Detection and track, Opencv, Qt

---

**Encadrants**  
Pascal Makris  
[pascal.makris@univ-tours.fr](mailto:pascal.makris@univ-tours.fr)  
Donatello Conte  
[donatello.conte@univ-tours.fr](mailto:donatello.conte@univ-tours.fr)

**Étudiants**  
Lin SHEN  
[lin.shen@etu.univ-tours.fr](mailto:lin.shen@etu.univ-tours.fr)  
DI5 2014 - 2015