



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

Département Informatique

[illegible]

TABLE DES MATIERES

Table des matières	3
Cahier de spécification Système	5
1. Introduction	5
2. Contexte de la réalisation	5
2.1. Contexte	5
2.2. Objectifs	5
2.3. Hypothèses.....	5
2.4. Bases méthodologiques	5
A) Méthodes de travail	5
B) Outil de gestion de versions	6
C) Outil de gestion de projet.....	6
3. Description générale.....	6
3.1. Environnement du projet.....	6
3.2. Caractéristiques des utilisateurs	7
3.3. Fonctionnalités et structure générale du système.....	7
3.4. Contraintes de développement, d'exploitation et de maintenance	7
4. Description des interfaces externes du logiciel	8
4.1. Interfaces matériel/logiciel	8
4.2. Interfaces homme/machine	9
5. Architecture générale du système.....	9
6. Description des fonctionnalités	10
6.1. Définition de la fonction Détection	10
6.2. Définition de la fonction Suivi	11
7. Conditions de fonctionnement	11
7.1. Performances	11
7.2. Capacités	12
7.3. Sécurité	12
Plan de développement	13
8. Découpage du projet en tâches	13
8.1. Tâche apprentissage	13
8.1.1. Description de la tâche.....	13
8.1.2. Estimation de charge.....	13
8.2.1. Description de la tâche.....	13
8.2.2. Estimation de charge.....	13
8.3.1. Description de la tâche.....	13
8.3.2. Estimation de charge.....	13
8.4.1. Description de la tâche.....	13
8.4.2. Estimation de charge.....	13

8.5.1.	Description de la tâche.....	13
8.5.2.	Estimation de charge.....	13
8.6.1.	Description de la tâche.....	14
8.6.2.	Estimation de charge.....	14
8.7.1.	Description de la tâche.....	14
8.7.2.	Estimation de charge.....	14
8.8.1.	Description de la tâche.....	14
8.8.2.	Estimation de charge.....	14
8.9.1.	Description de la tâche.....	14
8.9.2.	Estimation de charge.....	14
8.10.1.	Description de la tâche	14
8.10.2.	Estimation de charge	14
8.11.1.	Description de la tâche	14
8.11.2.	Estimation de charge	15
8.12.1.	Description de la tâche	15
9.	Planning(Gantt)	15
	Glossaire.....	16
	Bibliographie	17
	Index.....	18

CAHIER DE SPECIFICATION SYSTEME

1. Introduction

Ce document a pour but de définir précisément le projet de fin d'étude (PFE) suivant : Détection et suivi d'animaux dans les vidéos. Il a été proposé par M. Pascal Makris, en collaboration avec M. Donatello Conte. Nous parlerons du contexte, du problème posé, des objectifs ainsi que la manière dont nous comptons répartir le travail tout au long du projet.

2. Contexte de la réalisation

2.1. Contexte

Dans le cadre du projet régional **OVIN2A** qui vise à identifier les causes de mortalité des agneaux élevés en allaitement artificiel, une des tâches plus importantes est celle d'analyser leur comportement dans les premières années de vie. Les activités spontanées des agneaux (locomotion, alimentation, activité debout/couché, jeu) seront quantifiées à partir de vidéos enregistrées en continu. L'analyse des comportements est donc effectuée en analysant principalement les trajectoires parcourus par les différents agneaux ainsi que les interactions entre eux. Dans la littérature scientifique existent plusieurs méthodes de détection et suivi d'objets en mouvement dans une vidéo afin d'extraire les trajectoires des objets présents dans la vidéo.

2.2. Objectifs

Le travail de ce projet consistera à implémenter un système de suivi d'objets, éventuellement particularisé pour le suivi d'animaux, qui puisse fournir les trajectoires des agneaux et d'autres informations utiles à détecter leur comportements.

En particulier les principaux modules à mettre en place seront les suivants:

- Un module de configuration des caméras (réglage des paramètres des algorithmes, configuration des régions d'intérêts dans la vidéo, etc.);
- Un module de détection et suivi des animaux en mouvement;
- Un module de sauvegarde des résultats.

Le travail comprendra la rédaction de cahier de spécification, rapport technique et utilisateurs.

2.3. Hypothèses

Lorsque nous créons le premier système de détecter et de suivi, nous utilisons une seule caméra pour suivre un seul animal. Si il y a d'autres animaux se déplacent dans la scène, ou il y a des changements de scène, le système peut être désactivé.

Lorsque le premier test du système est réussie, nous allons améliorer le système afin qu'il puisse suivre plusieurs animaux.

2.4. Bases méthodologiques

A) Méthodes de travail

Durant ce projet, je vais essayer de respecter au maximum les théories de **traitement d'image numérique** et étudier l'**Open Source Computer Vision Library**.

OpenCV est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. Nous pouvons trouver de nombreuses fonctions dans le OpenCV, pour

réaliser l'algorithmes de traitement d'image.

Pour traitement d'images, **OpenCV** propose la plupart des opérations classiques en traitement bas niveau des images :

- lecture, écriture et affichage d'une image ;
- calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs ;
- lissage, filtrage ;
- seuillage d'image (méthode d'Otsu, seuillage adaptatif)
- segmentation (composantes connexes, GrabCut) ;
- morphologie mathématique.

Pour traitement vidéos, **OpenCV** s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision des ordinateurs tels que :

- lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra)
- détection de droites, de segment et de cercles par Transformée de Hough
- détection de visages par la méthode de Viola et Jones
- cascade de classifieurs boostés
- détection de mouvement, historique du mouvement
- poursuite d'objets par mean-shift ou Camshift
- détection de points d'intérêts
- estimation de flux optique (Méthode de Lucas–Kanade)
- triangulation de Delaunay
- diagramme de Voronoi
- enveloppe convexe
- ajustement d'une ellipse à un ensemble de points par la méthode des moindres carrés

Pour l'outils de développement, j'ai choisi d'utiliser **Microsoft Visual Studio**, et en langage **C++**.

B) Outil de gestion de versions

Les outils de gestion de versions permettent de garder l'ensemble des fichiers déposés dessus tout en conservant la chronologie de toutes les modifications faites sur chaque fichier. De plus, si jamais il y a une modification faite sur un fichier par deux personnes à la fin, l'outil est capable, sous certaines conditions, de fusionner le code afin de provoquer un nombre de conflit très restreint.

J'ai choisi d'utiliser le **SVN** pour des soucis d'accessibilité et le fait que l'école possède un **Redmine** et un SVN qui fonctionne.

C) Outil de gestion de projet

La gestion de projet est une méthode permettant de faciliter le travail de gestion du projet : autrement dit, cela permet d'automatiser des tâches de sauvegarde, ainsi que de gestion du temps.

Pour des raisons d'accessibilité, j'ai choisi d'utiliser le **Redmine** de Polytech'Tours.

3. Description générale

3.1. Environnement du projet

Il s'agit d'un nouveau projet, on n'a pas de projet **existant**. Donc il n'y a pas de projet précédente pour référence.

Nous allons étudier les algorithmes associés. Et puis, Nous voulons essayer d'établir le premier système.

3.2. Caractéristiques des utilisateurs

Les principaux utilisateurs de ce système seront donc les chercheurs de l'INRA.

Les caractéristiques qui affecteront l'« interface utilisateur »:

- connaissance ou non de l'informatique :
Les utilisateurs connaissent l'informatique mais n'est pas un informaticien. Nous allons créer une interface utilisateur simple et facile à utiliser. Il ne nécessite pas beaucoup de connaissances informatiques .
- expérience de l'application : aucune
Comme il s'agit d'un nouveau système, personne n'a l'expérience pertinente. Nous allons fournir aux utilisateurs de la version bêta. Grâce à la rétroaction des expériences des utilisateurs, nous pouvons améliorer les fonctions du système et de l'interface graphique.
- droits d'accès utilisateurs : **CRUD**
Les utilisateurs peuvent créer des fichiers vidéo, importer des fichiers vidéo. Après le traitement du vidéo, le système va créer un fichier .xls pour stocker les coordonnées de l'animal. Ce fichier .xls contient également des informations de l'animal. Les utilisateurs peuvent importer le fichier, lire le fichier, mais n'ont pas l'autorisation de modifier le ce fichier.

3.3. Fonctionnalités et structure générale du système

Il y a aura 2 fonctionnalités distinctes :

1) Détection d'animaux

Sur cette fonctionnalité, nous voulons identifier le animal en mouvement dans la vidéo, enregistrées certaines des caractéristiques de l'animal. Et marquer le animal en mouvement. Utilisateurs peuvent voir ces vidéos dans l'interface que nous avons créé.

Lorsque les utilisateurs démarrez l'enregistrement vidéo, il faut enregistrer une scène initiale, aucun des objets en mouvement, c'est l' image de fond. L'image de fond sera utilisé dans le traitement d'image après.

Dans le premier système, nous pouvons surveiller uniquement un seul animal.

2) Suivi d'animaux dans les vidéos

Sur cette fonctionnalité, nous allons suivre l'animal qui a détecté à l'étape précédente.

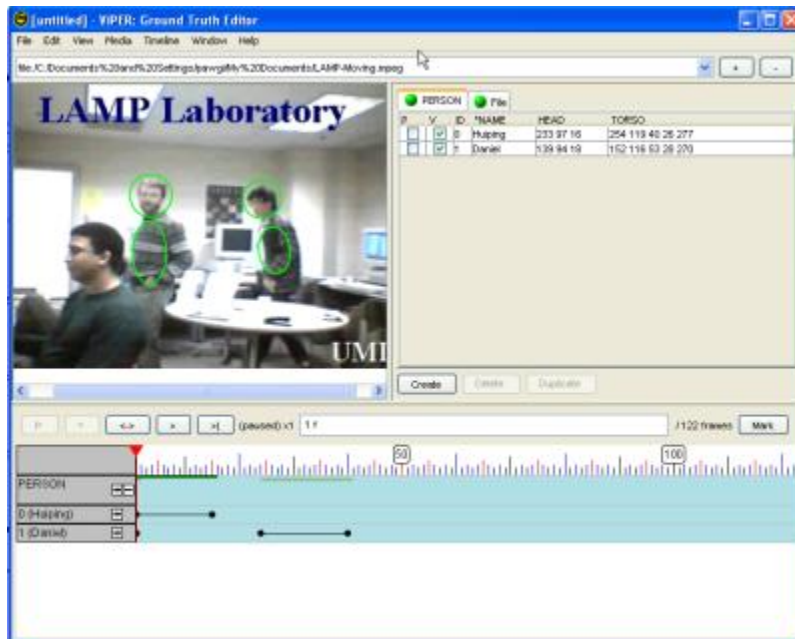
Enregistrer le tracé du mouvement de l'animal dans la scène, et affichées sur l'interface utilisateur.

3.4. Contraintes de développement, d'exploitation et de maintenance

3.4.1. Contraintes de développement

Préciser les contraintes liées aux :

- matériels : **Camera**
Nous avons déjà une caméra pour tester.
- langages de programmation imposés ou adoptés : **C++**
Dans la nouvelle version de OpenCV, C++ est plus pratique et plus efficace.
- logiciels de base à utiliser pour le développement : Eventuellement pour la vérité terrain on utilisera **ViPER**



- environnements nécessaires : simulateurs, outils logiciels : **Microsoft Visual Studio**
Visual Studio est un ensemble complet d'outils de développement. Mais c'est pas gratuit.
- bibliothèques de programmes imposées : **openCV**
OpenCV (pour Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. Nous pouvons trouver de nombreux fonction dans le OpenCV, pour réaliser l'algorithmes de traitement d'image.
- délais de réalisation
Fin d'Avril

3.4.2. Contraintes d'exploitation

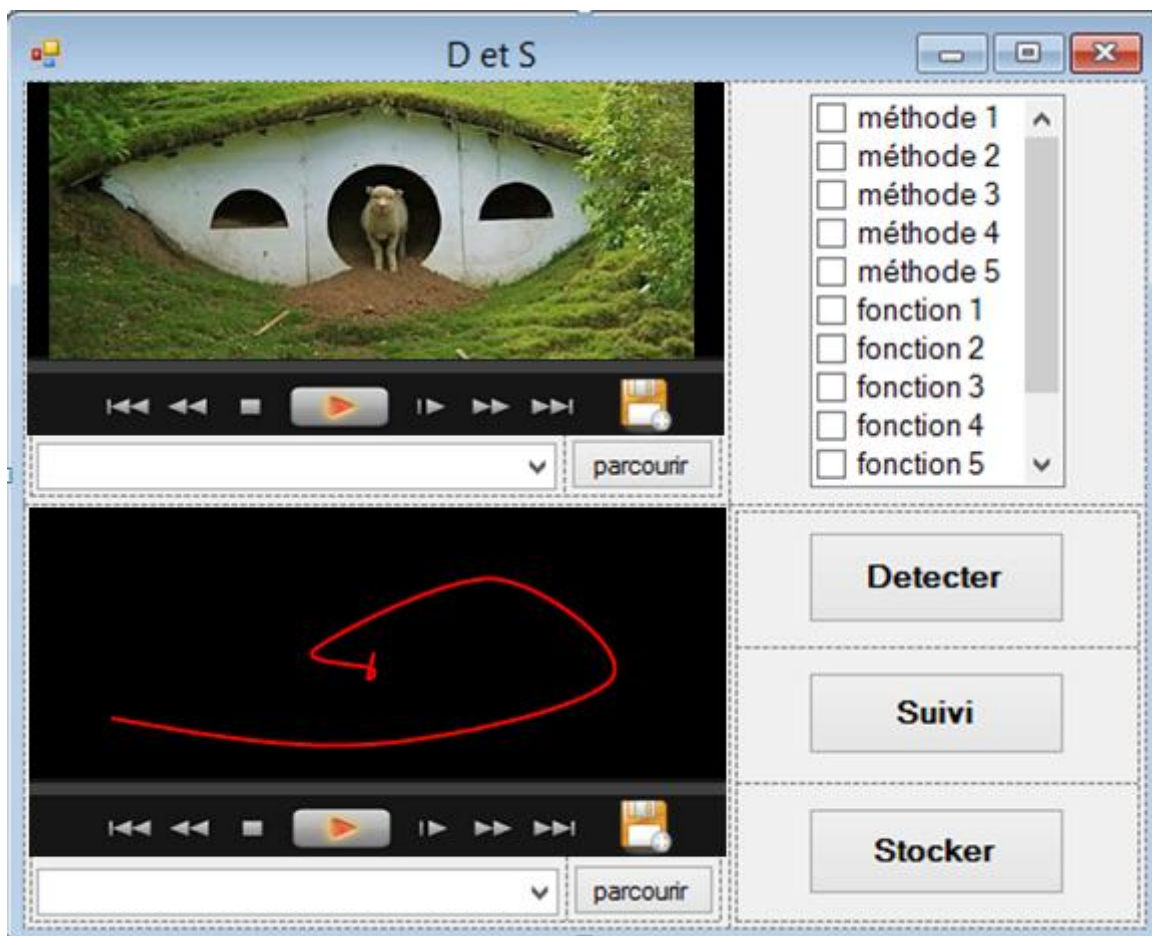
Il s'agit d'un nouveau projet, nous avons besoin de trouver un algorithme approprié, il n'ya pas de système complet. Maintenant, nous ne pouvons pas prédire des contraintes d'exploitation.

4. Description des interfaces externes du logiciel

4.1. Interfaces matériel/logiciel



4.2. Interfaces homme/machine



En haut de l'interface, une grande fenêtre est pour 'play' le vidéo obtenu par le webcam ou le vidéo enregistré dans votre ordinateur.

En dessous de la fenêtre, trois boutons de gauche à droite représentent « démarrer », « arrêter », « stocker ».

La fenêtre du tracé est dans le coin de gauche, il affiche les images du tracé du mouvement.

Il y a une liste des méthodes et des fonctions, vous pouvez choisir la méthode de détection et de suivi ou la fonction de traiter chaque image de la vidéo.

À droite de la fenêtre de la trace, il y a trois boutons : « détecter », « suivi » et « stocker ».

« détecter » est pour commencer la détection et afficher dans la première fenêtre.

« suivi » est pour commencer le suivi de l'animal et créer une table des coordonnées de l'animal.

« stocker » est pour enregistrer le tracé, transformer la table des coordonnées sous la forme de l'Excel.

5. Architecture générale du système

L'application sera réalisée en C++ et OpenCV comme précisé précédemment.

Nous allons maintenant énumérer les différents composants nécessaires au fonctionnement de l'application :

Composant Image :

Nous pouvons enregistrer une vidéo en utilisant d'une caméra vidéo existante. Le format de la vidéo, la taille de la vidéo, le taux d'image de la vidéo dépend de la performance de la caméra vidéo.

Composant Animaux :

Définir une zone, l'animal se déplacer dans cette zone. Il peut y avoir un seul animal. Une caméra vidéo pour enregistrer l'activité de l'animal dans cette zone.

Composant Détection de l'animal :

Detecter l'animal dans la vidéo et marquer l'animal sur les images de la vidéo.

Composant Suivi de l'objet :

Dessiner la tracé de l'animal dans la vidéo. Et puis creer d'une table, le tableau contient les coordonnées du animal et de temps. Enregistrer le coordonne environ toutes les 0,5 secondes.

Composant Affichage :

Creer une interface utilisateur. Comme nous l'avons vu en 4.2.

6. Description des fonctionnalités

6.1. Définition de la fonction Détection

6.1.1. Identification de la fonction Détection

- nom de la fonction : Détection d'animaux
- Détection d'animaux : Sur cette fonctionnalité, nous voulons identifier le animal en mouvement dans la vidéo, enregistrer certaines des caractéristiques de l'animal et marquer le animal sur la image de la vidéo. Utilisateurs peuvent trouver le bouton « detecter » dans l'interface, et voir ces vidéos dans l'interface que nous avons créé.

6.1.2. Description de la fonction Détection d'animaux

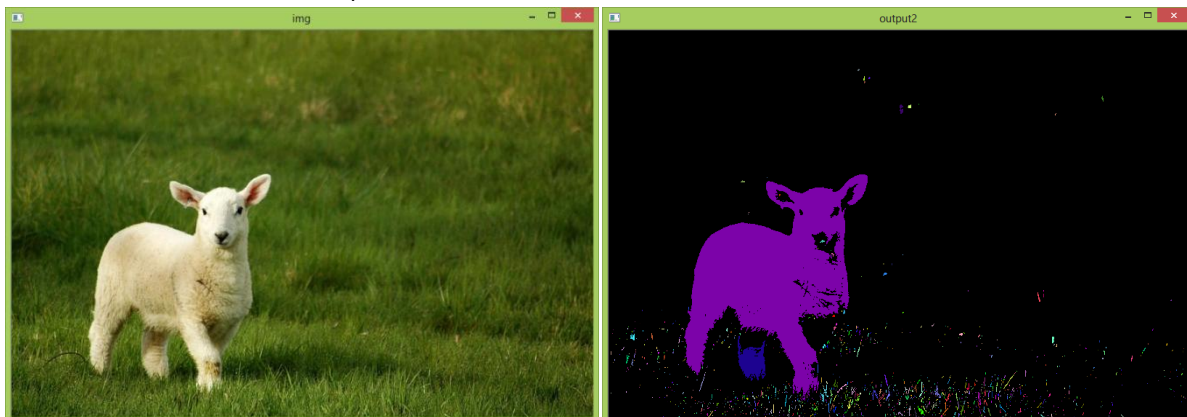
Cette fonctionnalité est basée sur la théorie des Connected Components.

Dans l'API de OpenCV, on peut trouver le fonction suivant :

`int connectedComponents(InputArray image, OutputArray labels, int connectivity=8, int ltype=CV_32S)`

- **image** – the image to be labeled
- **labels** – destination labeled image
- **connectivity** – 8 or 4 for 8-way or 4-way connectivity respectively
- **ltype** – output image label type. Currently CV_32S and CV_16U are supported.

Cette fonction est utile pour trouver domaine connecté.



Avant utilisation, nous avons besoin de traiter d'abord l'image originale, pour se conformer à le format de paramètre du fonction. Comme il peut y avoir beaucoup de domaine connecté dans une image. Donc, nous voulons filtrer et supprimer les parties que on ne besoin pas.

Lorsque les utilisateurs démarrez l'enregistrement vidéo, il faut enregistrer une scène initiale, aucun des objets en mouvement, qui servira image de fond. L'image de fond sera utilisé pour filtrer et supprimer les parties que on ne besoin pas.

Dans le premier système, nous pouvons surveiller uniquement un seul animal.

6.2. Définition de la fonction Suivi

6.2.1. Identification de la fonction Suivi

- nom de la fonction : Suivi d'animaux
- Suivi d'animaux : Sur cette fonctionnalité, nous allons suivre l'animal qui a détecté à l'étape précédente. Enregistrer le tracé du mouvement de l'animal dans la scène, et affichées sur l'interface utilisateur.

6.2.2. Description de la fonction Suivi d'animaux

Cette fonctionnalité est basée sur la théorie des Connected Components, Find Contours, Feature Detection.

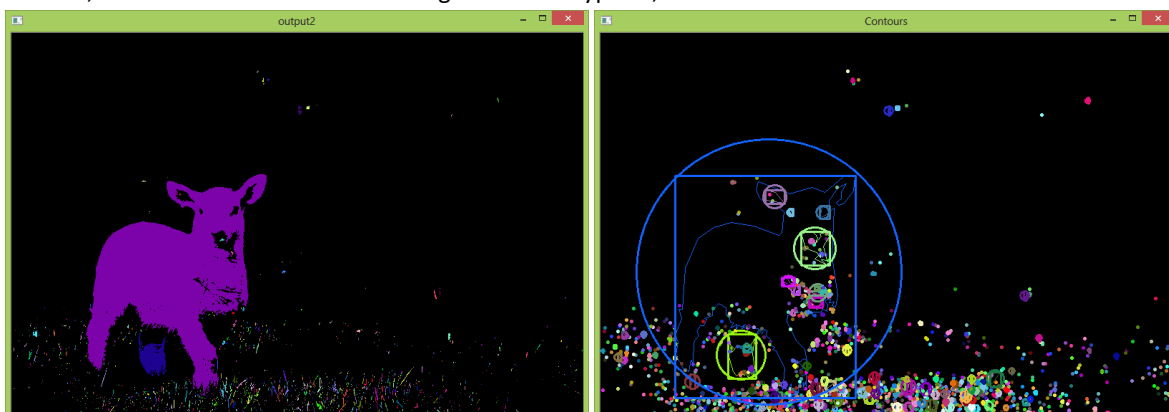
Dans l'API de OpenCV, on peut trouver le fonction suivant :

```
void findContours(InputOutputArray image, OutputArrayOfArrays contours, int mode, int method, Point offset=Point())
```

```
void Canny(InputArray image, OutputArray edges, double threshold1, double threshold2, int apertureSize=3, boolL2gradient=false )
```

Référence Intro : <http://docs.opencv.org/master/modules/refman.html>

Nous allons dessiner le contour de domaine connecté, par le proportion de la longueur et la largeur du contour, d'autres informations de la image comme keypoint, réaliser suivi de l'animal.



Selon les coordonnées de l'animaux sur l'écran, on peut dessiner la trajectoire.

7. Conditions de fonctionnement

La caméra accrochée à cinq mètres de haut. Zone de surveillance est une bergerie de 40m²⁵(5*8) en intérieur. Il y a seulement une petite fenêtre dans la maison, donc ils ont besoin de plus de lumière artificielle. Et notre première version du système qui ne supporte que la fonction de detection et de suivi d'un seul animal.

7.1. Performances

Les performances dépendent du matériel. Une caméra HD peut enregistrer une vidéo clair, pas de vidéo tremblotante. Plus de la résolution pour la vidéo, plus de la demande pour le matériel du système. Une vidéo avec haute résolution est propice au traitement d'image , elle a une très bonne qualité de chaque image de la vidéo. Nous nous appuierons sur la situation réelle, ajuster les paramètres du système afin d'obtenir de meilleures performances.

7.2. Capacités

Nous avons une seule caméra.

Le système n'agit que sur la même scène.

Dans la scène ne peut avoir qu'un seul animal.

Les données traitées sont stockées dans une table simple. Ça ne prend pas beaucoup de mémoire.

Pour enregistrer le tracé, on transforme la table des coordonnées sous la forme de l'Excel. La taille du fichier est très petit, ne considèrent pas le problème de l'espace de stockage.

Parce qu'il y a une limite du nombre de lignes d'Excel, donc sur les capacités de stockage, nous allons faire le traitement approprié.

7.3. Sécurité

Ce système est utilisé pour la recherche biologique, les données utilisées et les données obtenues sont ouvertes, il n'y a pas de problème de sécurité.

PLAN DE DEVELOPPEMENT

8. Découpage du projet en tâches

8.1. Tâche apprentissage

8.1.1. Description de la tâche

Apprenez des théories sur le traitement d'image et certaines méthodes de traitement d'image.

8.1.2. Estimation de charge

4 semaines - 4 heures par semaine.

8.2. Tâche Configuration de l'environnement

8.2.1. Description de la tâche

Choisir un environnement de développement. Configuration environnement de OpenCV.
(C++,VS,openCV)

8.2.2. Estimation de charge

1 semaine - 4 heures par semaine.

8.3. Tâche Mise en Oeuvre

8.3.1. Description de la tâche

Ecrire les procédures de test. Pratiquer les méthodes de traitement d'image, pratiquer les méthodes on a appris dans le tâche apprentissage.

8.3.2. Estimation de charge

8 semaines - 4 heures par semaine.

8.4. Tâche CDS

8.4.1. Description de la tâche

Ecrire le Cahier des specifications.

8.4.2. Estimation de charge

8 semaines - 4 heures par semaine.

8.5. Tâche Detection

8.5.1. Description de la tâche

Réaliser les fonctions d'identifier un même animal dans plusieurs photos.

8.5.2. Estimation de charge

2 semaines - 16 heures par semaine.

8.6. Tâche Configuration de l'équipement de caméra

8.6.1. Description de la tâche

Configurez le équipement de caméra.

Analyser les informations de chaque trame de la vidéo.

8.6.2. Estimation de charge

1 semaine -8 heures par semaine.

8.7. Tâche Detection avancé

8.7.1. Description de la tâche

Réaliser les fonctions de l'identification des animaux dans la vidéo.

8.7.2. Estimation de charge

2 semaines - 16 heures par semaine.

8.8. Tâche Suivi

8.8.1. Description de la tâche

Réaliser les fonctions de suivi des animaux dans la vidéo.

8.8.2. Estimation de charge

3 semaines - 16 heures par semaine.

8.9. Tâche Trace

8.9.1. Description de la tâche

Dessiner la trajectoire. Enregistrer les coordonnées du animal dans un table.

Pour engistrer le tracé, on transforme la table des coordonnées sous la forme de l'Excel.

8.9.2. Estimation de charge

2 semaine - 20 heures par semaine.

8.10. Tâche IHM

8.10.1. Description de la tâche

Creer une interface utilisateur. Comme nous l'avons vu en 4.2.

8.10.2. Estimation de charge

2 semaine - 20 heures par semaine.

8.11. Tâche Tester

8.11.1. Description de la tâche

错误！未找到引用源。

Plan de développement

Tester notre système.

8.11.2. Estimation de charge

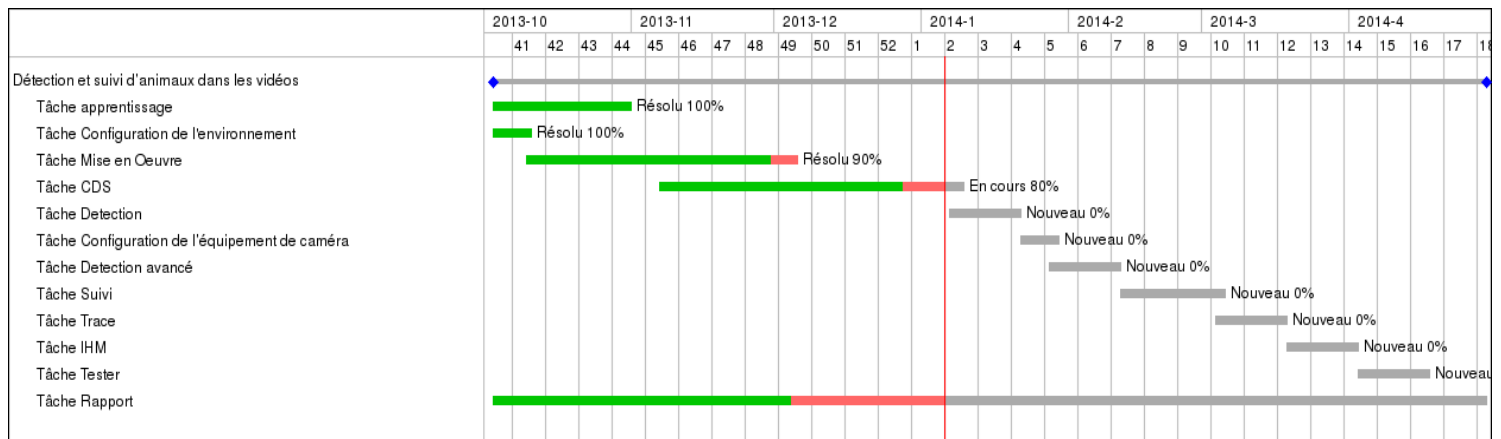
2 semaine - 20 heures par semaine.

8.12. Tâche Rapport

8.12.1. Description de la tâche

Rédiger des rapports. Tout au long du projet.

9. Planning(Gantt)



GLOSSAIRE

OVIN2A :

l'INRA : institut national de la recherche agronomique

SVN: Subversion (en abrégé svn) est un logiciel de gestion de versions, distribué sous licence Apache et BSD. Il a été conçu pour remplacer CVS.

Redmine: Redmine est une application web libre de gestion de projets presque complète en mode web, développée en Ruby sur la base du framework Ruby on Rails. La gestion des tests devra être faite avec un autre outil.

CRUD: Create, Read, Update, Delete.

VIPER: The Video Performance Evaluation Resource.

BIBLIOGRAPHIE

OpenCV: <http://opencv.org/>

SVN: http://fr.wikipedia.org/wiki/Apache_Subversion
<http://subversion.apache.org/>

Redmine: <http://www.redmine.org/>

<http://redmine.polytech.univ-tours.fr>

Les fonctions de OpenCV : <http://docs.opencv.org/master/modules/refman.html>

INDEX

Cette partie indique les pages où sont traités et mentionnés les sujets et les termes les plus importants du document.

Aucune entrée d'index n'a été trouvée.