

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

Projet Systèmes d'Information

2017-2018

Prototype d'application de quantification de mouvement de la volaille



POLYTECH[®]
TOURS

Tuteur académique

Pascal MAKRIS

Étudiants

Kaixing ZHAO (DI5)

Boyang WANG (DI5)

Zhicong LIU (DI5)

Jing YANG (DI5)

Mengyu HU (DI5)

18 janvier 2018



Liste des intervenants

Nom	Email	Qualité
Kaixing ZHAO	kaixing.zhao@etu.univ-tours.fr	Étudiant DI5
Boyang WANG	boyang.wang@etu.univ-tours.fr	Étudiant DI5
Zhicong LIU	zhicong.liu@etu.univ-tours.fr	Étudiant DI5
Jing YANG	jing.yang@etu.univ-tours.fr	Étudiant DI5
Mengyu HU	mengyu.hu@etu.univ-tours.fr	Étudiant DI5
Pascal MAKRIS	pascal.makris@univ-tours.fr	Tuteur académique, Département Informatique



Avertissement

Ce document a été rédigé par Kaixing ZHAO, Boyang WANG, Zhicong LIU, Jing YANG et Mengyu HU susnommés les auteurs.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par Pascal MAKRIS susnommé le tuteur académique.

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

Les auteurs reconnaissent 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.

Les auteurs attestent que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

Les auteurs attestent ne pas s'appropriier le travail d'autrui et que le document ne contient aucun plagiat.

Les auteurs attestent que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

Les auteurs reconnaissent qu'ils ne peuvent diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

Les auteurs autorisent 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

Kaixing ZHAO, Boyang WANG, Zhicong LIU, Jing YANG et Mengyu HU, *Prototype d'application de quantification de mouvement de la volaille:* , Projet Systèmes d'Information, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2017-2018.

```
@mastersthesis{
  author={ZHAO, Kaixing and WANG, Boyang and LIU, Zhicong and YANG, Jing and HU,
    Mengyu},
  title={Prototype d'application de quantification de mouvement de la volaille: },
  type={Projet Systèmes d'Information},
  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	iv
1 Introduction	1
1 Contexte de la réalisation.....	1
2 Objectifs de l'application.....	2
2 Description générale	3
1 Environnement du projet	3
2 Fonctionnement du système existant.....	3
3 Structure générale du système existant	3
3 Analyse de l'existant	5
1 fonctionnalités implémentées	5
1.1 Récupération de la vidéo	5
1.2 Sélection de la zone d'intérêt	5
1.3 Algorithme de traitements d'images	5
1.4 Sélection d'opérateur	5
1.5 Affichage du graphe	6
2 Code incomplet.....	6
2.1 Algorithme flots optiques.....	6

2.2	Détection de l'opérateur	7
2.3	IHM non ergonomique	7
2.4	Manque de commentaires du code.....	7
2.5	Structure du code.....	8
3	Bogues et solutions trouvés	8
4	Les algorithmes utilisés	9
1	Algorithme de Distance.....	9
2	Algorithme de Flot Optique	10
3	Comparaison de deux algorithmes	11
5	Améliorations apportées	13
1	Ergonomie intuitive.....	13
2	Détection automatique de l'opérateur	13
3	Commentaires.....	14
4	Flot optique.....	14
5	Création d'un guide utilisateur.....	15
6	Gestion de projet	16
1	Organisation du groupe	16
2	Outils utilisés.....	16
3	Méthodologie	16
4	Planning.....	16
7	Problèmes rencontrés	18
1	Mauvaise qualité de la version précédente	18
2	Mauvaise qualité des vidéos.....	18
8	Travaux restant à réaliser	19
1	Création d'une sortie des données	19
2	Mettre le curseur de la courbe suit le changement de frame	19
9	Conclusion	20
	Annexes	21
A	Spécifications fonctionnelles	22
1	Diagramme de cas d'utilisation	22
2	Diagramme de classe	23
3	Définition de la fonction "Détection d'opérateur"	23
4	Définition de la fonction "Sélection Zone intérêt"	24

B	Spécifications non fonctionnelles	25
1	Contraintes de développement et conception	25
2	Caractéristiques des utilisateurs	25
3	Contraintes de fonctionnement et d'exploitation	25
3.1	Performances.....	25
3.2	Capacités.....	25
3.3	Contrôlabilité.....	25
3.4	Sécurité	26
C	Technologies utilisées	27
1	Python.....	27
2	OpenCV	27
3	PyQt	27
4	Matplotlib	28

Table des figures

1 Introduction

1	Logo du SYSAAF.....	1
---	---------------------	---

2 Description générale

1	Diagramme de classes réalisé par les étudiants précédent	4
---	--	---

3 Analyse de l'existant

1	Graphe initiale généré	6
2	Fenêtre normal	7
3	Fenêtre zoomé.....	7

4 Les algorithmes utilisés

1	Detection de passants.....	9
2	Résultat d'algorithme de distance.....	11
3	Points de flot.....	11
4	Résultat d'algorithme de flot optique	12

5 Améliorations apportées

1	Nouvelle interface	13
2	la trajectoire des points caractéristiques.....	14
3	Résultat de flot optique	15

6 Gestion de projet

1	Diagramme de Gantt	17
---	--------------------------	----

A Spécifications fonctionnelles

1	Diagramme de cas d'utilisation	22
2	Diagramme de classe	23

1

Introduction

Ce document présente l'amélioration d'un système doté d'une interface graphique permettant de quantifier le mouvement d'une volaille au travers d'une vidéo. Le prototype fourni à notre équipe a été réalisé par plusieurs étudiants au cours d'un précédent projet en 2016. Il s'appuie lui même sur un autre projet de l'école de 2014.

Actuellement, le projet s'inscrit dans le cadre d'un projet SI de 5ème année spécialité informatique. Nous sommes une équipe constituée de cinq personnes : Kaixing, Boyang, Zhicong, Jing et Mengyu.

Ce projet a été proposé par le SYSAAF **Figure 1** (Chapitre 6), le Syndicat des Sélectionneurs Avicoles et Aquacoles Français. Nous sommes encadrés par Pascal Makris, enseignant chercheur au département informatique de l'École Polytechnique de l'Université de Tours.



Figure 1 – Logo du SYSAAF

1 Contexte de la réalisation

Les éleveurs de volailles cherchent à minimiser le stress de leurs animaux lors de l'envoi à l'abattoir afin de maximiser la qualité de la viande. Pour ce faire, ils cherchent à sélectionner les espèces les moins sujettes à des situations de stress afin de les faire se reproduire et ainsi créer des lignées plus calmes.

Le procédé mis en place actuellement se fait par analyse sanguine (ou bien par simple observation). Cette technique nécessite l'intervention d'un expert et d'un laboratoire la rendant longue et coûteuse.

2 Objectifs de l'application

Le coeur de ce projet est le développement d'une application qui sera réalisée en Python et qui sera amenée à être utilisée pour l'automatisation de quantification du mouvement de la volaille lors de la mise sur chaîne à l'abattoir.

Avant d'être abattu, la volaille est transportée la tête en bas sur une chaîne, et accrochée par les pattes. La méthode de quantification de mouvement mis en place consiste à traiter des vidéos prises au moment de cette mise sur chaîne. En effet, c'est à cet instant que la volaille est la plus stressée par conséquent nous cherchons à déterminer celles qui réagissent le moins à cet état de stress. Soit la volaille à tendance à être inactive et à ne pas bouger, soit au contraire cette dernière est agitée et se débat en déployant ses ailes afin d'essayer de s'échapper.

2

Description générale

1 Environnement du projet

Le prototype reçu et à modifier était une version incomplète du projet. Les étudiants précédents ont réalisé une application mais cette application ne satisfait pas toutes nos demandes et il y existe encore des bogues à corriger.

La version fournie n'était pas bien commentée et comportait des portions de code mises en commentaire sans savoir si ces portions étaient fonctionnelles ou non. Il y avait également des codes non commenté mais jamais appelé.

En plus, l'application existante n'est pas ergonomique, les interfaces ne conviennent pas aux différents systèmes.

Avec l'aide de notre encadrant, nous avons décidé de refaire l'interface et corriger les algorithmes non fonctionnel, en plus rajouter de nouvelles fonctionnalités.

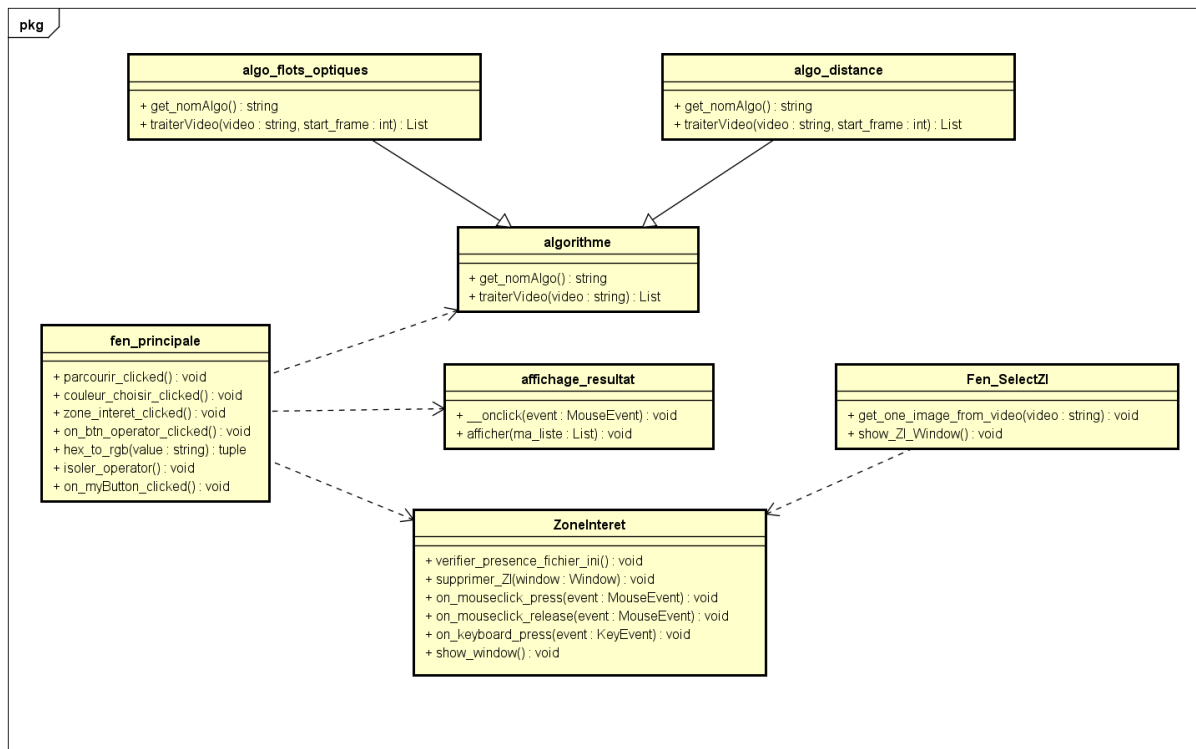
2 Fonctionnement du système existant

L'application fonctionne de la manière suivante. Tout d'abord, il faut lancer l'application pour démarrer le programme puis choisir une zone d'intérêt. Ensuite l'onglet "Algorithmes" permet de choisir entre un calcul de mouvement par distance ou par flot optique. Maintenant il n'y a que l'algorithme par distance qui fonctionne. A cette étape il nous manque la vidéo à traiter, dans l'onglet fichier le bouton ouvrir y remédie. Puis l'utilisateur choisit une couleur d'opérateur mais celui-ci ne fonctionne pas. Une fois le traitement lancé par le bouton "Lancer le traitement", une fenêtre apparaît contenant le résultat sous formes d'un graphique.

3 Structure générale du système existant

Les étudiants précédents ont tenté de créer une application respectant les principes du MVC (Modèle Vue Contrôleur). Pour cela, ils ont créé des classes ne s'occupant que d'afficher les éléments de l'interface graphique du logiciel et d'appeler d'autres classes traitant les informations importantes (comme appliquer les différents algorithmes sur les vidéos récupérées).

Figure 1



powered by Astah

Figure 1 – Diagramme de classes réalisé par les étudiants précédent

3

Analyse de l'existant

1 fonctionnalités implémentées

Dans cette partie, nous parlerons les fonctionnalités déjà implémentés dans cette application et leurs fonctionnements.

1.1 Récupération de la vidéo

Un bouton permettait d'ouvrir une fenêtre de sélection de fichier. Une vidéo pouvait être choisie pour futur traitement.

1.2 Sélection de la zone d'intérêt

Un bouton permettait d'ouvrir une fenêtre de sélection de fichier, l'utilisateur peut choisir une vidéo pour déterminer une zone d'intérêt.

1.3 Algorithme de traitements d'images

Deux algorithmes ont été implémentés afin de pouvoir détecter les mouvements dans les vidéos. L'algorithme par distance entre deux images ainsi que l'algorithme par flots optiques. Dans l'ancien projet, il n'y a que l'algorithme par distance qui fonctionne bien et l'autre sera réalisé et amélioré dans notre projet.

1.4 Sélection d'opérateur

Dans le projet précédent, les étudiants voulaient réduire l'influence de l'opérateur donc élever la confiance sur le résultat, ils ont implémenté une sélection de couleur d'opérateur pour cet objectif, mais cette fonctionnalité ne fonctionne pas, elle est juste une partie de l'interface.

1.5 Affichage du graphe

Un graphique correspondant aux résultats renvoyés par les deux algorithmes précédents était généré grâce à la librairie matplotlib, librairie de référence en python. Il était cependant dénué d'informations ou d'interactivité. **Figure 1**

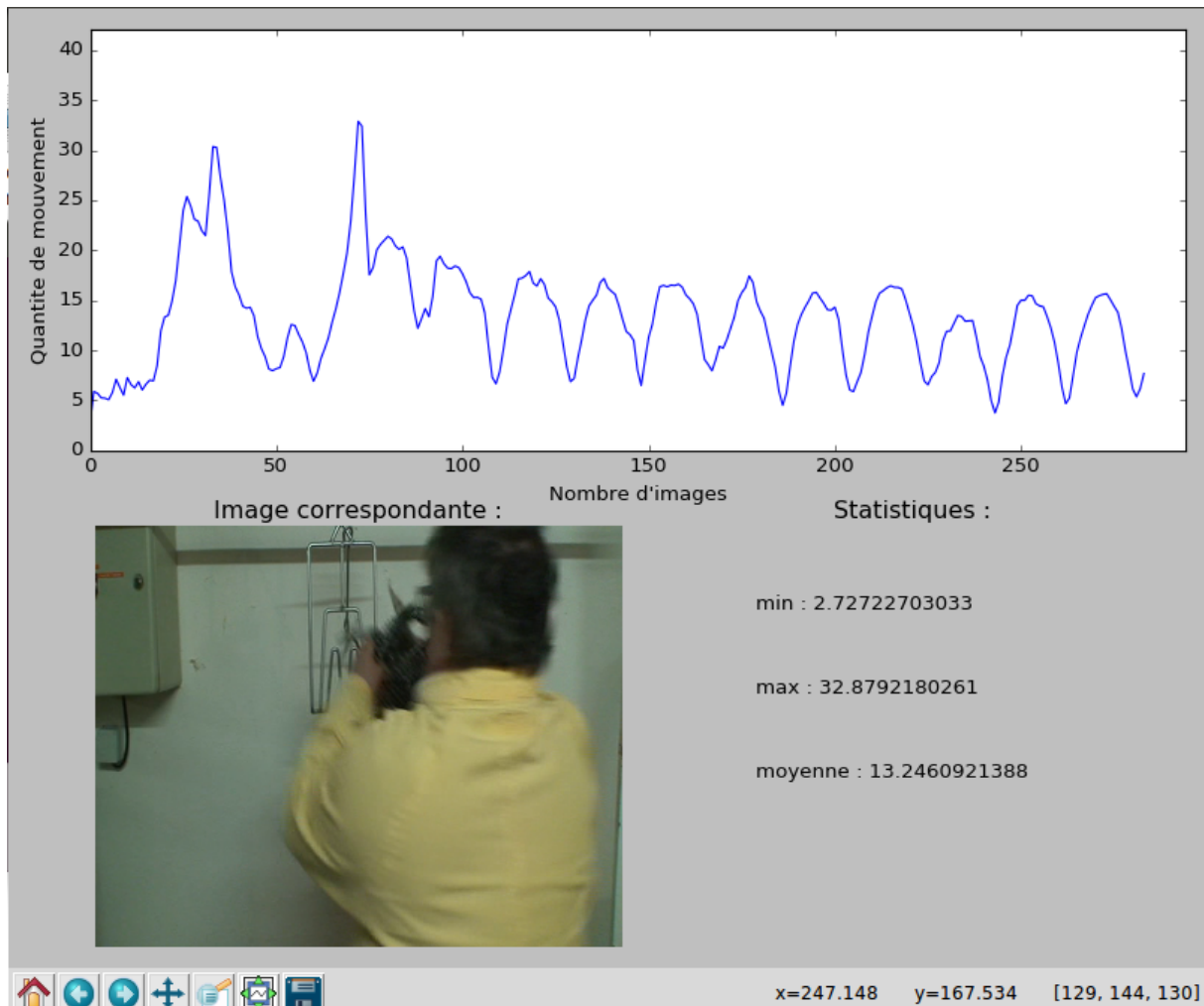


Figure 1 – Graphe initiale généré

2 Code incomplet

Après la lecture des anciens rapports et du code, nous avons pu trouver cinq axes d'amélioration de cette application. Les fonctions suivantes sont incomplètes ou n'ont pas fonctionné.

2.1 Algorithme flots optiques

L'algorithme flots optiques est pour la détection du mouvement. Nous avons testé cette algorithme, mais elle n'a pas fonctionné. Le résultat obtenu par cette algorithmes n'est pas correct.

2.2 Détection de l'opérateur

La fonction de détection de l'opérateur n'a pas bien fonctionné.

2.3 IHM non ergonomique

Dans cette application, il faut d'abord choisir une vidéo et une zone d'intérêt. Pourtant, c'est difficile de trouver la fonction de "ZoneIntérêt". Après le test, il s'est avéré que le fenêtre devraient être zoomé pour utiliser la fonction de "ZoneIntérêt" dans le menu.

Le fenêtre normal est comme [Figure 2](#). Le fenêtre zoomé est comme [Figure 3](#).

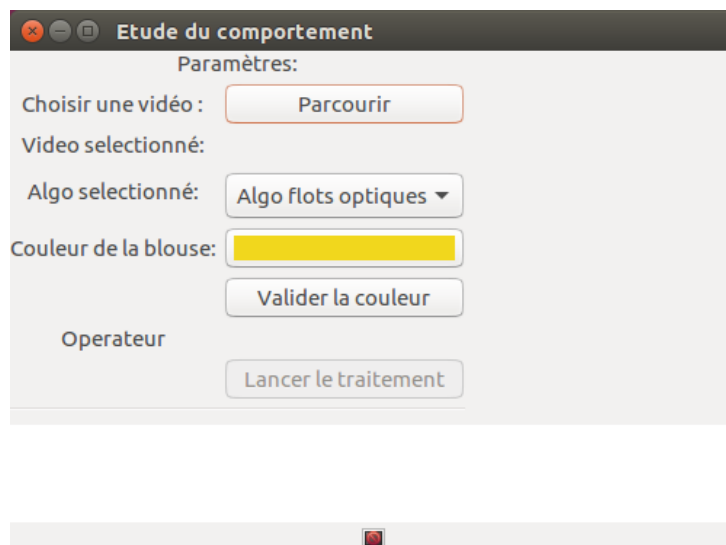


Figure 2 – Fenêtre normale

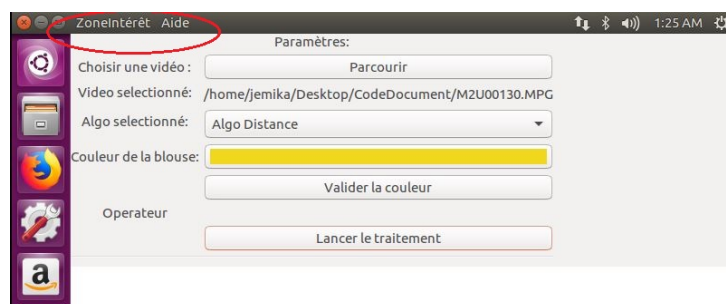


Figure 3 – Fenêtre zoomé

Ensuite, la fonction de choisir une vidéo et la fonction de choisir une zone d'intérêt sont redonant.

Hors cela, pour détecter le mouvement, il est nécessaire de exécuter étape par étape. Mais dans l'application existée, l'interface est mauvaise et elle n'a pas défini les étapes.

2.4 Manque de commentaires du code

Pour le code de source, il y a peu de commentaires. Et il n'y a pas de document pour la reprise du code. Pour cela, nous avons dépensé beaucoup de temps pour comprendre le code.

2.5 Structure du code

Le modèle MVC n'est pas utilisé. Le code est tout mélangé donc il est en désordre.

3 Bogues et solutions trouvés

4

Les algorithmes utilisés

Pour notre projet, nous avons réalisé deux algorithmes : l'algorithme de Distance et l'algorithme de Flot Optique. Ce chapitre a le but de présenter des mécanismes de ces deux algorithmes.

1 Algorithme de Distance

C'est un algorithme très simple mais utile dans le domaine de Computer Vision. La distance entre deux images, en fait, est la réalisation de calcul de soustraction. Cet algorithme permet d'identifier le mouvement de la poule entre plusieurs images successives. Par exemple, si la poule ne bouge pas d'une image à l'autre, la différence sera vide. Mais si on trouve une différence, alors la poule a bougé et plus la poule fait de grand mouvement et plus on détecte une différence importante.

Il existe beaucoup d'applications de cet algorithme quand on a besoin de traiter des images ou vidéos. Par exemple, la soustraction d'arrière-plan est une étape majeure du prétraitement dans de nombreuses applications basées sur la vision. Par exemple, considérez les cas comme un compteur de visiteurs où une caméra statique prend le nombre de visiteurs entrant ou sortant de la pièce, ou une caméra de circulation extrayant des informations sur les véhicules etc. Dans tous ces cas, on doit d'abord extraire les personnes ou les véhicules. Techniquement, on doit extraire le premier plan mobile de l'arrière-plan statique.



Figure 1 – *Detection de passants*

L'image ci-dessus nous donne un exemple de Background Subtraction, on peut détecter les

personnes dans la rue en faisant la soustraction de la rue sans passant avec celle avec les passants.

2 Algorithme de Flot Optique

Le terme de flot optique a été inventé par le psychologue James Jerome GIBSON dans une étude sur la vision humaine. En 1980, HORN et SCHUNCK ont proposé une méthode d'estimation du flot optique basée sur la régularisation. La mesure du flot optique est une étape de traitement de l'image dite de bas niveau. Il existe déjà de nombreuses applications comme l'analyse de mouvements de fluides en physique expérimentale, la compression de séquences d'images vidéo par compensation de mouvement, ou son utilisation pour des phases de traitement des images de plus haute niveau, comme la reconstruction de scènes tridimensionnelles.

Concernant le mécanisme de cet algorithme, on sait qu'une séquence d'images est une fonction réelle $I(t, x_1, x_2)$ de trois variables t , x_1 et x_2 que nous supposons pour l'instant continues. Nous utiliserons les notations agrégées \mathbf{x} pour (x_1, x_2) et $\mathbf{x}(t)$ pour $(x_1(t), x_2(t))$. Le modèle mathématique standard utilisé pour trouver des équations qui définissent le flot optique est basé sur un hypothèse d'illumination constante :

Un point réel $[X_1(t) X_2(t) X_2(t)]$ de la scène est projeté sur le plan image de la caméra au point $(x_1(t), x_2(t))$ au temps t .

Le flot optique au temps t et au point $\mathbf{x}(t)$ est défini comme la vitesse du point image :

$$\mathbf{v} = (v_1, v_2) = \left(\frac{dx_1}{dt}, \frac{dx_2}{dt} \right)$$

L'hypothèse d'illumination constante consiste à dire que la luminosité apparente du point $(x_1(t), x_2(t))$ au temps t ne dépend pas du temps t , ce qui s'écrit :

$$I(t; \mathbf{x}(t)) = I_0$$

Le flot optique est donc contraint par l'équation suivante :

$$\frac{\partial I}{\partial t} + \nabla I \cdot \frac{d\mathbf{x}}{dt} = 0$$

ou

$$\mathbf{v} \cdot \nabla I + \frac{\partial I}{\partial t} = 0$$

Selon les mécanisme ci-dessus, on peut savoir le flot optique est l'ensemble des vecteurs de déplacement d'un objet dans une vidéo. Pour chaque image, des vecteurs sont générés en rapport à un mouvement particulier. Le flot optique fonctionne selon l'hypothèse que l'intensité d'un pixel ne change pas sur toute la vidéo. De même, on suppose dans cet algorithme, que les pixels voisins de ce dernier subissent le même mouvement.

Pendant l'utilisation d'algorithme flot optique, il ne devra y avoir aucune fluctuation dans les ombres ou la luminosité dans les vidéos à traiter. La caméra devra être placée sur trépied pour éviter tout mouvement de caméra car si le référentiel est en déplacement, l'ensemble des points de l'image seront en mouvement.

3 Comparaison de deux algorithmes

Dans cette partie, nous allons donner des comparaisons de ces deux algorithmes en utilisant le logiciel amélioré. Cette année, nous avons aussi fait des vidéos grâce aux outils que M. Pascal Makris nous offre. En fait, pour que nos résultats puissent être plus objectifs, nous avons réalisé des vidéos d'une vingtaine de secondes d'un poids suspendu à un ressort. Nous voudrions constater la vibration harmonique simple et aussi voudrions obtenir les courbes qui peuvent décrire ce mouvement.

Nous pouvons regarder l'image ci-dessous qui s'est produit par l'algorithme de distance, on peut voir cet algorithme peut bien simuler le mouvement.

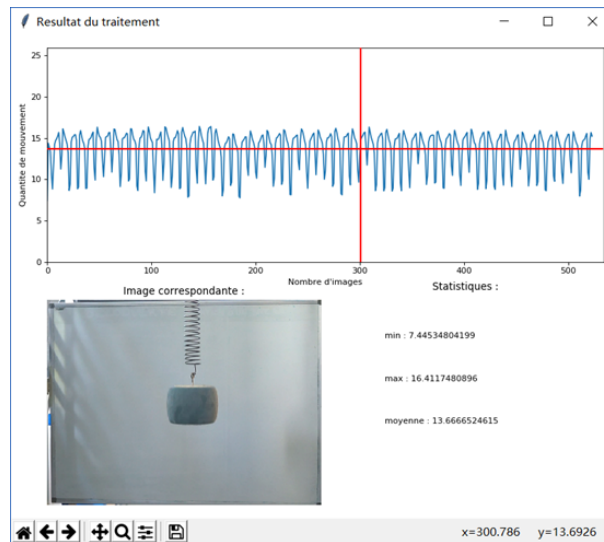


Figure 2 – Résultat d'algorithme de distance

Concernant l'algorithme flot optique, quand nous avons utilisé la même vidéo, nous pouvons obtenir le résultat comme ci-dessous :

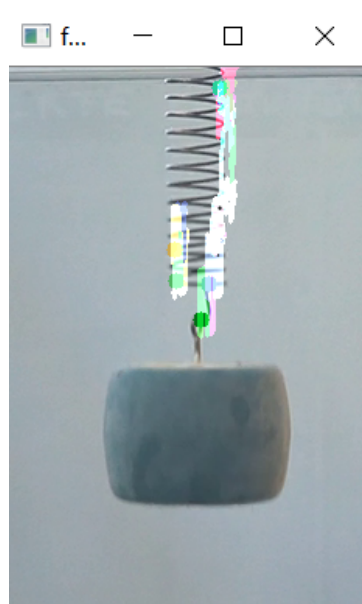


Figure 3 – Points de flot

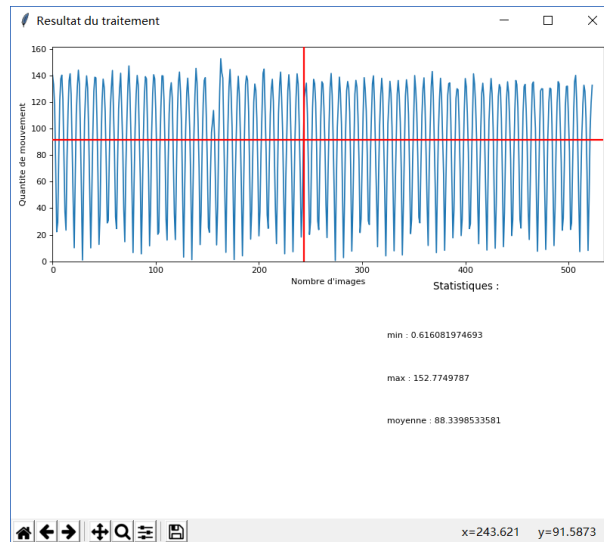


Figure 4 – *Résultat d'algorithme de flot optique*

Ces tests permettent de mettre en évidence la sensibilité au bruit des deux algorithmes, et également leurs limitations. L'utilisation de flot optique nécessite une vidéo de meilleure qualité ou l'objet ne doit jamais sortir du cadre.

5

Améliorations apportées

1 Ergonomie intuitive

Considérant les interfaces graphiques de la version précédente, bien que le programme nous offre la fonctionnalité de manipulation par les interfaces graphiques, il n'est pas encore assez facile et claire pour les utilisateurs. Cependant, nous avons repensé les interfaces. En fait, nous concevons les interfaces graphiques en fonction de la séquence des opérations. La version actuelle peut nous donner plus d'informations (par exemple on peut voir la zone d'intérêt sélectionnée après avoir fini la sélection sur la vidéo) et elle peut aussi aider les clients de mieux comprendre le processus de traitement.

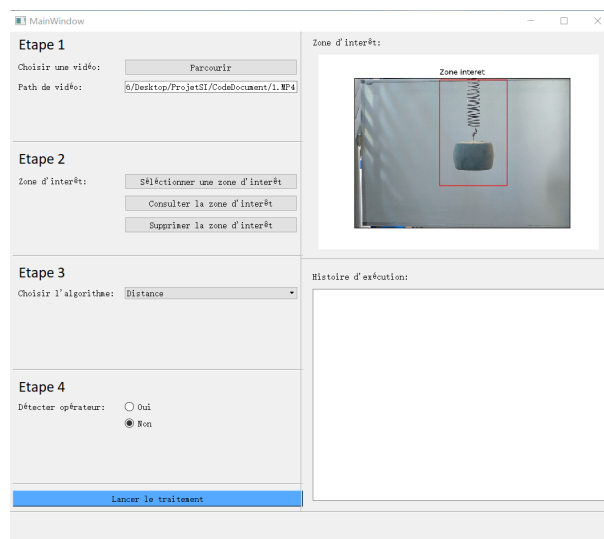


Figure 1 – Nouvelle interface

2 Détection automatique de l'opérateur

Afin d'éliminer les présences d'opérateur qui peuvent fausser l'analyse de vidéo, nous devons ajouter la fonctionnalité de détection de l'opérateur dans notre programme. Il peut nous aider

de connaître à partir de quelle image on peut commencer le traitement. Comme nous avons déjà dit dans la partie précédente, il existe déjà une classe qui peut détecter la couleur de l'opérateur, mais après avoir une analyse précise, on pense que c'est mieux que l'on propose une nouvelle idée de détection (Car la réalisation précédente n'est pas performante).

Notre idée repose sur le calcul de la distribution des valeurs de gris, elle peut détecter les images qui contiennent l'opérateur automatiquement et en même temps essayer de les enlever. A cause de grand nombre de calculs, cette opération prend un peu de temps en général. Pour plus d'informations, on l'explique dans l'annexe.

3 Commentaires

Parce qu'il y a presque aucun commentaire dans les codes sources originale, pour faciliter la compréhension des codes pour les clients, nous avons ajouté des commentaires pour tous les fichiers. Bien que les clients n'aient pas une telle demande, nous pensons que l'ajout du commentaire approprié est vraiment nécessaire pour un projet informatique. Cela nous permet de modifier les codes plus facilement et de plus il peut réduire aussi considérablement le temps nécessaire pour la maintenance.

4 Flot optique

Nous avons réécrit les codes de l'algorithme Flot Optique pour le faire fonctionner correctement. Dans la nouvelle version de la méthode Flot Optique, nous avons amélioré la méthode de calcul de la distance et aussi modifié les mesures de résultat pour que l'affichage est plus propre. Nous avons également ajouté la trajectoire des points caractéristiques pour que on puisse bien montrer le fonctionnement de l'algorithme Flot Optique.

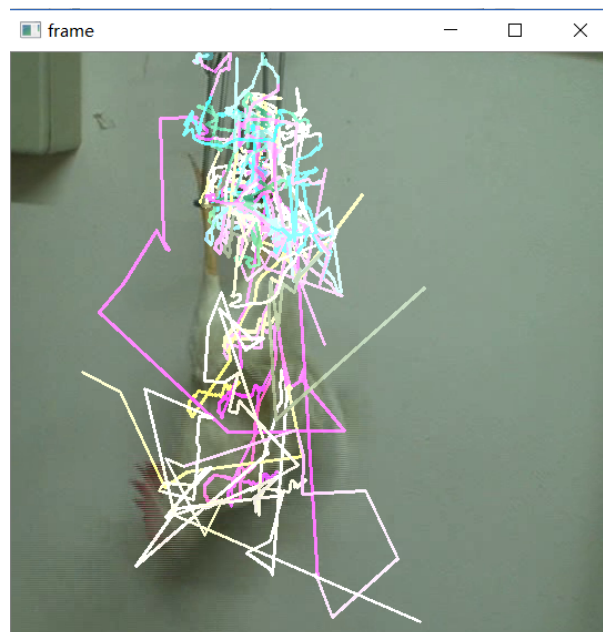


Figure 2 – la trajectoire des points caractéristiques

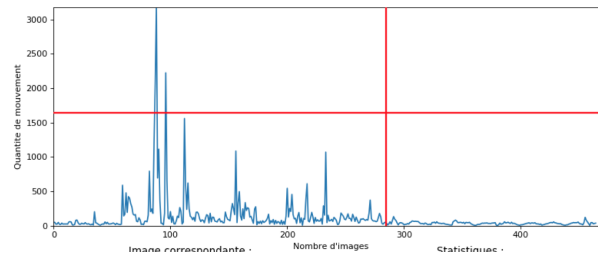


Figure 3 – Résultat de flot optique

5 Création d'un guide utilisateur

Pour que notre client puisse utiliser notre produit plus simplement, on a créé un guide utilisateur qui indique les différentes étapes d'utilisation et de plus la configuration et l'environnement nécessaire pour notre projet.

6

Gestion de projet

1 Organisation du groupe

2 Outils utilisés

Afin de conduire et gérer ce projet, l'outil GanttProject est utilisé pour réaliser la planification des tâches et le suivi de celle-ci.

Le projet sera suivi grâce à Trello qui nous permet de gérer l'avancement du projet (Documentation, Rendez-vous...).

Pour la partage des documents nous avons utilisé Google Drive. Github est utilisé pour la gestion de version.

Durant ce projet, l'ensemble de la documentation est réalisé sous Latex.

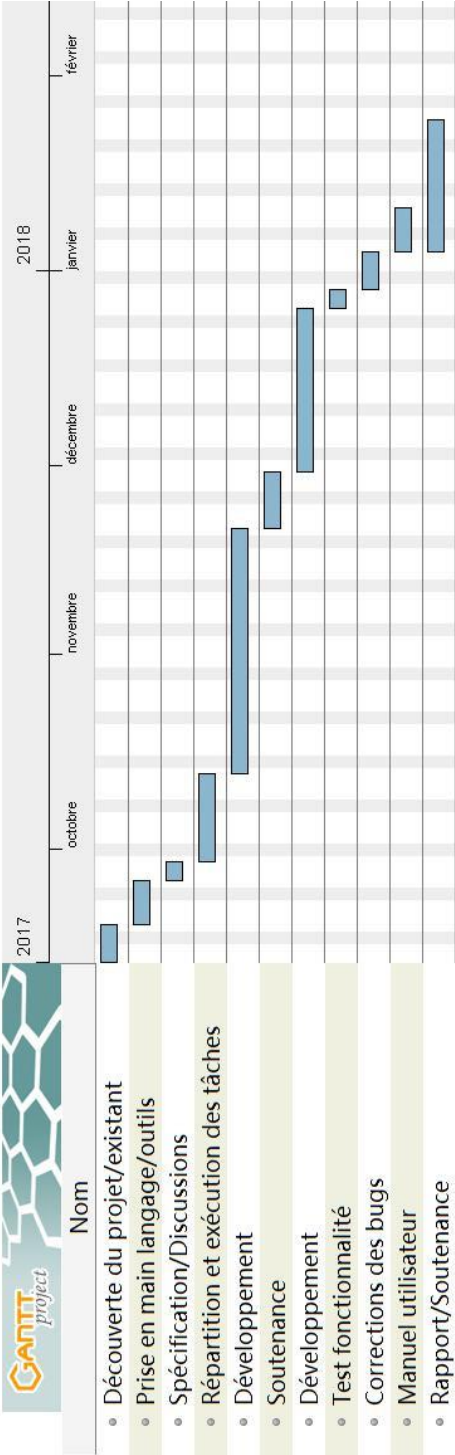
3 Méthodologie

La méthodologie pour gérer le projet a été la suivante :

- Prendre en main le projet, le langage, le contexte, les rapports et les outils utilisés au début du projet.
- Discuter avec l'équipe et l'encadrant de la suite à donner au projet — Modéliser les besoins du client (diagramme de cas d'utilisation, séquences, classes) — Répartir et planifier les tâches — Communiquer avec notre encadrant et notre client sur l'état du projet — Réaliser les tâches. En cas de problèmes, faire tourner les membres de l'équipe dessus. Proposer une solution alternative si la tâche ne peut pas être réalisée comme prévue. — Tester les scénarios d'utilisation du logiciel

4 Planning

Le diagramme de Gantt est comme [Figure 1](#)



7

Problèmes rencontrés

1 Mauvaise qualité de la version précédente

Notre projet repose sur une idée d'amélioration d'un projet existant, la qualité de l'existant peut déterminer considérablement notre planning. En effet, à cause de mauvaise qualité du projet précédent, on a utilisé beaucoup de temps pour le bien comprendre et de plus pour corriger les bugs dans ce projet. Il est évident que le fait d'avoir reçu une application non terminée à modifier a forcément ralenti le travail de l'équipe et en même temps, un logiciel n'avait jamais été montré au client était difficile d'anticiper les modifications à venir.

2 Mauvaise qualité des vidéos

La mauvaise qualité des vidéos peuvent empêcher le fonctionnement d'algorithme Flot Optique. En fait, pour trouver cette raison, on a pris beaucoup de temps sur les codes et à la fin on a pensé que les exceptions sont causées par la mauvaise qualité des vidéos. Pour y remédier, on a fait plusieurs types de vidéos à l'aide de M. Pascal Makris et fait des tests en interne.

8

Travaux restant à réaliser

1 Création d'une sortie des données

Afin de pouvoir sauvegarder les données d'analyse une sortie CSV a été demandée. Chaque traitement d'une vidéo ajoute une ligne supplémentaire au fichier CSV se trouvant dans le dossier "resultats" du projet. Chaque ligne comporte le nom de la vidéo, la date et l'heure du traitement, l'algorithme utilisé et les valeurs importantes obtenues. De plus, Chaque traitement occasionne l'enregistrement du graphique sous forme d'image dans le même dossier.

2 Mettre le curseur de la courbe suit le changement de frame

Pour visualiser plus clairement le mouvement de volaille montré par la courbe obtenue comme le résultat de notre algorithme, il fallait de faire le curseur de la courbe suivre l'avancement de frame.

9

Conclusion

Ce projet était une très bonne occasion pour toute l'équipe de maîtriser les connaissances dans le domaine de Computer Vision, il peut aussi nous aider de montrer en compétence sur différents aspects : Technique, Gestion de projet, Communication avec les clients, etc. Il aussi nous permet de pratiquer le langage Python et l'utilisation de Qt.

De plus, ce projet concernait un réel besoin en lien avec des partenaires industriels. Grâce à ce projet, on a bien compris le réel processus d'un projet(y compris le dépôt de sujet, la définition des besoins, le planning, le développement et la livraison). Les résultats que nous avons obtenu sont satisfaisants et il peut être délivré pour avoir une évolution dans l'environnement industriel. Grâce aux les documents que nous offrons, ce projet peut aussi être repris facilement par d'autres étudiants dans la futur.

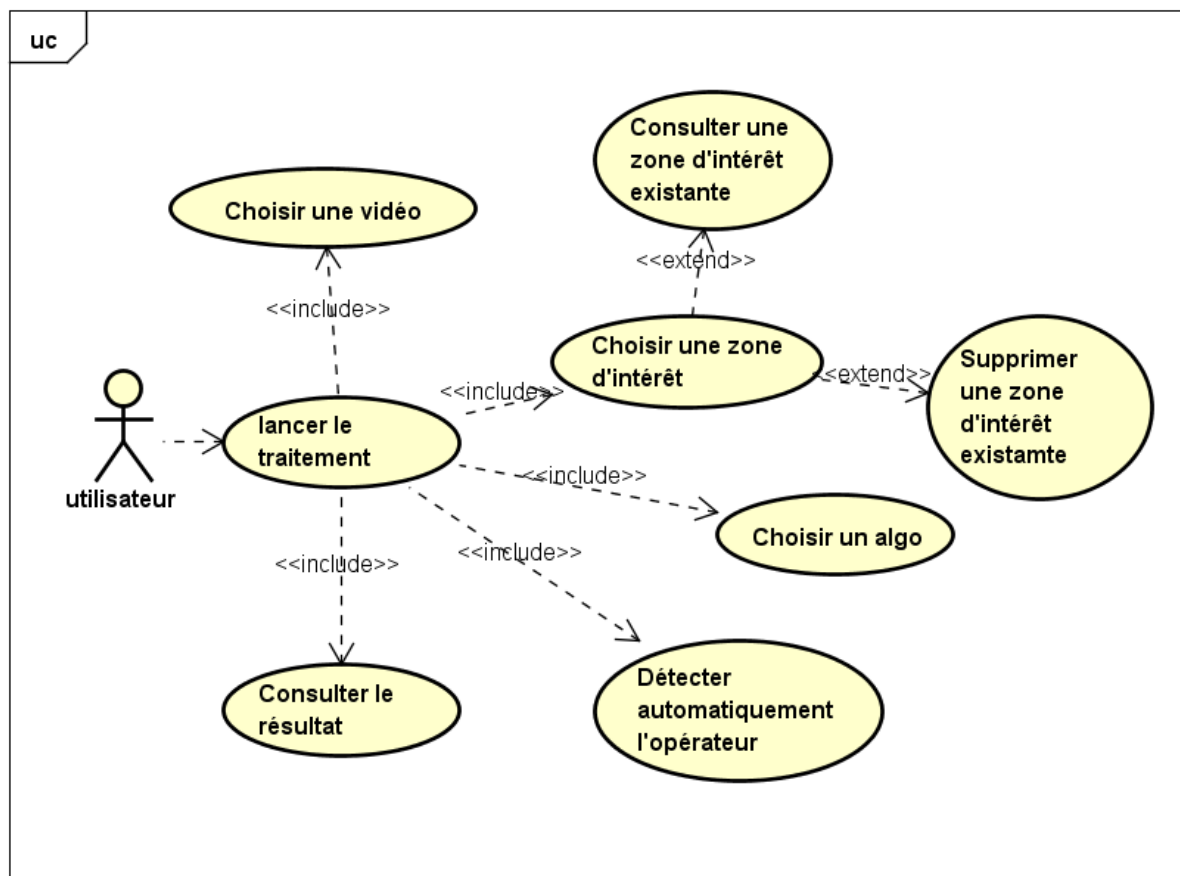
Annexes

A

Spécifications fonctionnelles

1 Diagramme de cas d'utilisation

Voici notre diagramme de cas d'utilisation. **Figure 1**



powered by Astah

Figure 1 – Diagramme de cas d'utilisation

2 Diagramme de classe

Voici notre diagramme de classes. **Figure 2**

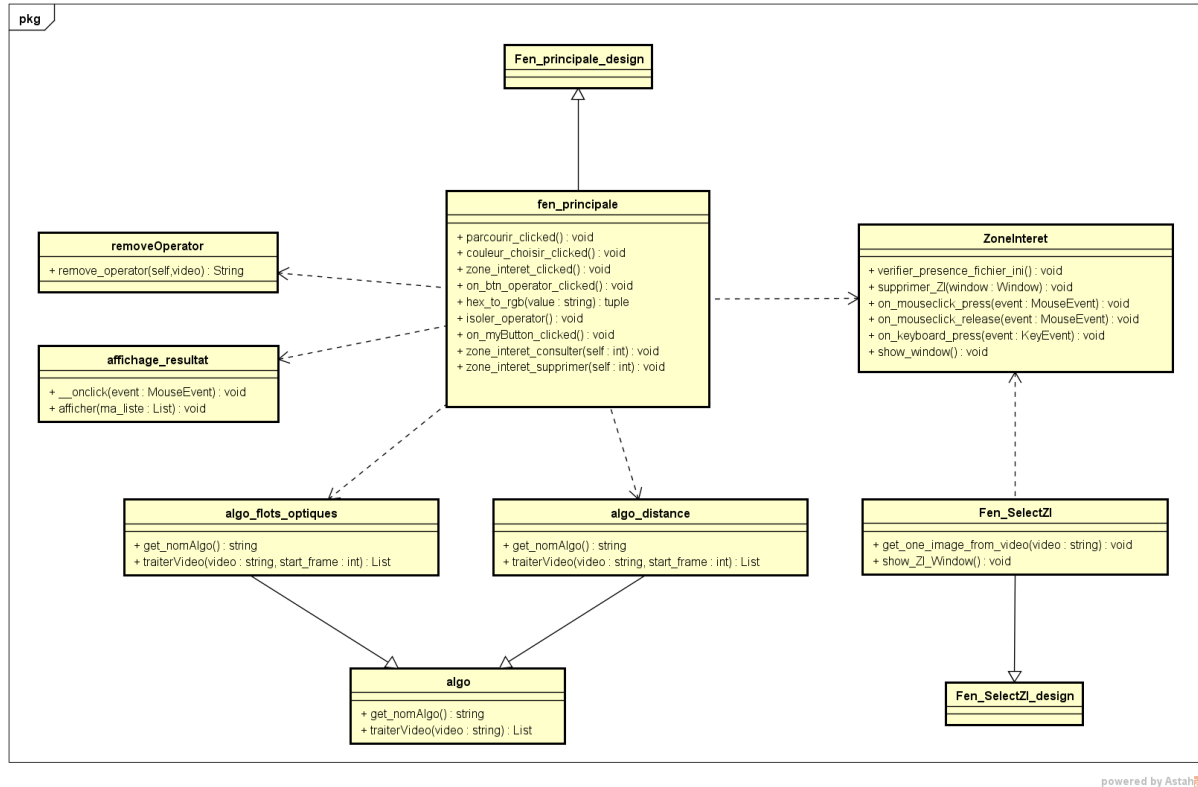


Figure 2 – Diagramme de classe

3 Définition de la fonction "Détection d'opérateur"

Objectif :

Détecter automatiquement l'opérateur en fonction de le taux de changement au niveau de gris, et générer une nouvelle vidéo qui enlève l'opérateur.

Objet utilisé :

OpenCV

Description fonctionnelle :

Après que l'utilisateur a sélectionné la vidéo, l'utilisateur peut choisir de supprimer l'opérateur ou non. Après que l'utilisateur a choisi d'enlever l'opérateur, le programme appellera l'algorithme interne pour traiter la vidéo. Après cela, le système générera automatiquement un nouveau fichier vidéo(video sans opérateur). Tous les algorithmes suivants seront basés sur la vidéo nouvellement générée.

Description technique :

Dans cette fonction, notre idée est de déterminer l'existence de l'opérateur en calculant le taux de changement de la valeur de gris de l'image. Le taux de changement standard comprend deux parties. La première partie est le taux de changement calculé à partir de deux images dans la vidéo (dans notre fonction c'est des images à les 1/2 et 3/4 de la vidéo). La deuxième partie consiste à un bon seuil défini. Nous comparons des deux taux de changement, le plus petit sera utilisé comme taux de changement standard. Après chaque image dans la vidéo sont comparées avec l'image au milieu et calcule le taux de changement de chaque image. Les images qui ont le taux de changement supérieure que le taux de changement standard sont considérés comme "contient un opérateur". Et les autres images vont être ajoutées dans la nouvelle vidéo.

4 Définition de la fonction "Sélection Zone intérêt"

Objectif :

Parce que dans la plupart des zones de l'image nous ne sommes pas intéressés, nous avons donc conçu cette méthode. En spécifiant de la zone intérêt afin que nous puissions nous concentrer sur la zone de traitement.

Objet utilisé :

OpenCV

Description fonctionnelle :

Une fois que l'utilisateur a sélectionné la vidéo à traiter, une zone intérêt a besoin d'être indiquée. La zone sélectionnée peut être enregistrée localement et peut être appelée directement dans la vidéo suivante.

Description technique :

L'idée principale de la méthode est de sélectionner la zone avec la souris, et on va obtenir la zone à traiter par les coordonnées des quatre points dans la zone sélectionnée. Nous traitons uniquement les images dans la zone.

B

Spécifications non fonctionnelles

1 Contraintes de développement et conception

Le développement de cette application sera continuée avec le langage Python et les interfaces seront réalisées via PyQt. De plus, la librairie OpenCV sera utilisée pour traiter les vidéos et Matplotlib pour l'affichage graphique.

2 Caractéristiques des utilisateurs

Il n'y a qu'un seul type d'utilisateur identifié dans l'application - les clients qui vont utiliser cette logiciel dans l'abattoir. Aucune autre demande n'a été évoquée pour cette partie.

3 Contraintes de fonctionnement et d'exploitation

L'interface doit être remaniée pour être plus simple et ergonomique.

3.1 Performances

Le traitement des vidéos devra donc être le plus court possible.

3.2 Capacités

Les objets traités en entrée étant lourds (fichier vidéos), il est nécessaire d'avoir assez de mémoires attribuées.

3.3 Contrôlabilité

Des messages suivant les différentes opérations et actions de l'utilisateur doivent indiquer l'état du logiciel.

3.4 Sécurité

Aucune consigne de sécurité n'a été demandée, que se soit au niveau de la gestion des utilisateurs ou du chiffage des données.

C

Technologies utilisées

Cette section reprend les résumés des anciens rapport et une partie de la nouvelle technologie qu'on a utilisée.

1 Python

D'après wikipedia : Python est un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par "ramasse-miettes" et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl. Le langage Python est placé sous une licence libre et fonctionne sur la plupart des plates-formes informatiques, des supercalculateurs aux ordinateurs centraux⁴, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et aussi avec Java ou encore .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

2 OpenCV

OpenCV (pour "Open Computer Vision") est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques. Cette bibliothèque 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 traitement informatisé de l'image.

3 PyQt

PyQt est un module libre qui permet de lier le langage Python avec la bibliothèque Qt distribué sous deux licences : une commerciale et la GNU GPL. Il permet ainsi de créer des interfaces graphiques en Python. Une extension de QtDesigner (utilitaire graphique de création d'interfaces Qt) permet de générer le code Python d'interfaces graphiques.

4 Matplotlib

Matplotlib est probablement l'un des modules Python les plus utilisés pour la représentation de graphiques en 2D. Il fournit aussi bien un moyen rapide de visualiser des données grâce au langage Python, que des illustrations de grande qualité dans divers formats. C'est avec matplotlib que nous avons créé nos graphiques en 2D quantifiant le mouvement de la volaille.

Prototype d'application de quantification de mouvement de la volaille

Résumé

Ce projet donne une bonne occasion aux étudiants pour approfondir la compréhension de traitement d'images et de vidéos. Il aussi pratique la compétence de programmation Python pour les étudiants. Grâce à ce projet, on peut automatiser la détection de stress de poules et de mettre la surveillance de qualité plus intelligente.

Mots-clés

Traitement d'images, Traitement de vidéos, OpenCV

Abstract

This project provides a good opportunity for students to deepen their understanding of image and video processing. He also practices the Python programming skill for students. With this project, we can automate chicken stress detection and make quality monitoring smarter.

Keywords

Image processing, Video processing, OpenCV

Tuteur académique

Pascal MAKRIS

Étudiants

Kaixing ZHAO (DI5)

Boyang WANG (DI5)

Zhicong LIU (DI5)

Jing YANG (DI5)

Mengyu HU (DI5)