



# **Système de suivi de mouvement**

**Mémoire**

**Zahra Rezaei**

**Maîtrise en génie électrique - avec mémoire**  
Maître ès sciences (M. Sc.)

Québec, Canada

# **Système de suivi de mouvement**

**Mémoire**

**Zahra Rezaei**

Sous la direction de:

Benoit Gosselin

# Résumé

Le comportement des petits animaux est important pour les chercheurs scientifiques et précliniques; ils veulent connaître les effets des interventions sur leur vie naturelle. Pour les maladies humaines, les rongeurs sont utilisés comme modèles. L'étude du comportement des rongeurs permet d'identifier et de développer de nouveaux médicaments pour les troubles psychiatriques et neurologiques.

La surveillance des animaux peut être traitée et un grand nombre de données traitées peuvent conduire à de meilleurs résultats de recherche dans un temps plus court. Ce mémoire présente le système de suivi du comportement des rongeurs basé sur des techniques de vision numérique.

En vision numérique, la détection d'un sujet consiste à balayer et à rechercher un objet dans une image ou une vidéo (qui n'est qu'une séquence d'images), mais la localisation d'un objet dans des images successives d'une vidéo est appelée suivi.

Pour trouver la position d'un sujet dans une image, nous avons utilisé la détection du sujet et le suivi, car le suivi peut aider lorsque la détection échoue et vice et versa. Avec cette approche, nous pouvons suivre et détecter tout type du sujet (souris, headstage, ou par exemple un ballon). Il n'y a pas de dépendance au type de caméra.

Pour trouver un sujet dans une image, nous utilisons l'algorithme AdaBoost en ligne qui est un algorithme de suivi du sujet et l'algorithme de Canny qui est un algorithme de détection du sujet, puis nous vérifions les résultats. Si l'algorithme Adaboost en ligne n'a pas pu trouver le sujet, nous utilisons l'algorithme Canny pour le trouver. En comparant les résultats de notre approche avec les résultats des algorithmes AdaBoost en ligne et Canny séparément, nous avons constaté que notre approche permet de mieux trouver le sujet dans l'image que lorsque nous utilisons ces deux algorithmes séparément.

Dans ce mémoire, nous décrivons les algorithmes de détection et de suivi du sujet.

# Abstract

Small animal behavior is important for science and preclinical researchers; they want to know the effects of interventions in their natural life.

For human diseases, rodents are used as models; studying rodent behavior is good for identifying and developing new drugs for psychiatric and neurological disorders.

Animal monitoring can be processed and a large number of data can lead to better research result in a shorter time. This thesis introduces the rodents' behavior tracking system based on computer vision techniques.

In computer vision, object detection is scanning and searching for an object in an image or a video (which is just a sequence of images) but locating an object in successive frames of a video is called tracking.

To find the position of an object in an image, we use object detection and object tracking together because tracking can help when detection fails and inversely. With this approach, we can track and detect any objects (mouse, headstage, or a ball). There is no dependency to the camera type.

To find an object in an image we use the online AdaBoost algorithm, which is an object tracking algorithm and the Canny algorithm, which is an object detection algorithm together, then we check the results. If the online Adaboost algorithm could not find the object, we use the Canny algorithm to find the object. By comparing the results of our approach with the results of the online AdaBoost and Canny algorithms separately, we found that our approach can find the object in the image better than when we use these two algorithms separately.

In this thesis, we will describe implemented object detection and tracking algorithms.

# Table des matières

<b>Résumé</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>Table des matières</b>	<b>v</b>
<b>Liste des figures</b>	<b>viii</b>
<b>Liste des tableaux</b>	<b>x</b>
<b>Liste de symboles</b>	<b>xi</b>
<b>Remerciements</b>	<b>xiii</b>
<b>Introduction</b>	<b>1</b>
1.1 Interface de machine à cerveau . . . . .	1
1.2 Motifs stimulant la lumière . . . . .	4
1.3 Comportement des rongeurs . . . . .	4
1.4 Objectifs de ces travaux . . . . .	5
1.5 plan du mémoire . . . . .	5
<b>2 Revue de littérature</b>	<b>7</b>
2.1 Méthodes de l'implémentation physique . . . . .	7
2.1.1 Système OSE sans fil (Fi-Wi) . . . . .	7
2.1.2 Système basé sur une cage intelligente . . . . .	7
2.2 Méthodes de suivi . . . . .	8
2.2.1 Système basé sur Kinect . . . . .	8
2.2.2 Système basé sur Edglet . . . . .	9
2.2.3 Algorithme de k-means . . . . .	9
2.2.4 Seuil de couleur . . . . .	10
2.2.5 Spatiotemporal 3D . . . . .	10
2.2.6 Le toilettage, la rotation et le redressement . . . . .	11
2.2.7 Système automatisé de reconnaissance d'animaux multiples . . . . .	11
2.2.8 Classificateur de fonctionnalités multiples . . . . .	11
2.2.9 Suivi de contour . . . . .	12
2.2.10 Technique de segmentation pour détecteur de bords . . . . .	12
2.2.11 Spectromètre comportemental . . . . .	12
2.2.12 Suivi en temps réel via Boosting en ligne . . . . .	13

2.2.13	Suivi visuel avec l'apprentissage en ligne à plusieurs instances . . . . .	13
2.2.14	Suivi par détection avec kernels . . . . .	13
2.2.15	Détection automatique . . . . .	14
2.2.16	Suivi d'apprentissage de détection . . . . .	14
2.2.17	Apprendre à suivre avec des réseaux de régression profonds . . . . .	14
2.3	État de la caméra . . . . .	15
2.3.1	Problèmes d'éclairage . . . . .	15
2.3.2	Calibrage de la caméra . . . . .	15
2.3.3	Emplacement de la caméra entre les lignes 2D et 3D et les correspon- dances de points . . . . .	15
2.3.4	Détermination de l'emplacement de la caméra à l'aide d'un ensemble de <i>m</i> points de contrôle . . . . .	16
2.3.5	Système de robot en temps réel . . . . .	16
2.4	Les systèmes de suivi commerciaux pour suivre un rongeur . . . . .	16
2.4.1	IntelliCage . . . . .	16
2.4.2	EthoVision XT . . . . .	17
2.4.2.1	Base EthoVision XT . . . . .	18
2.4.2.2	Module d'interaction sociale . . . . .	18
2.4.2.3	Module de reconnaissance du comportement de la souris . . . . .	18
2.4.3	CleverSys . . . . .	19
2.5	Conclusion . . . . .	21
<b>3</b>	<b>Présentation de la solution proposée</b>	<b>22</b>
3.1	Résumé . . . . .	22
3.2	Calibration de la caméra . . . . .	24
3.2.1	Paramètres extrinsèques . . . . .	24
3.2.2	Paramètres intrinsèques . . . . .	24
3.2.3	Distorsion dans la calibration de la caméra . . . . .	24
3.2.3.1	Radial Distortion . . . . .	25
3.3	Définition de détection du sujet et de suivi du sujet . . . . .	26
3.3.1	Suivi du sujet . . . . .	26
3.3.1.1	Suivi en temps réel via Boosting en ligne . . . . .	27
3.3.1.2	AdaBoost en ligne . . . . .	29
3.3.1.3	Caractéristiques . . . . .	30
3.3.1.4	Expériences . . . . .	31
3.3.1.5	de manière adaptative . . . . .	31
3.3.1.6	Robustesse . . . . .	31
3.3.1.7	Généralité . . . . .	32
3.3.2	Détection du sujet . . . . .	33
3.3.2.1	Soustraction d'arrière-plan . . . . .	33
3.3.2.2	Mélange de distributions gaussiennes <i>K</i> . . . . .	34
3.3.2.3	Modèle de mélange gaussien adaptatif . . . . .	35
3.3.2.4	Algorithmes EM en ligne . . . . .	36
3.3.2.5	Détection des ombres et modèle de couleur . . . . .	36
3.3.2.6	Algorithme de Canny . . . . .	37
3.3.3	Trouver le comportement de la souris . . . . .	38
<b>4</b>	<b>Résultats expérimentaux</b>	<b>39</b>

4.1	Installation . . . . .	39
4.2	Calibration de la caméra . . . . .	40
4.3	Suivi du sujet . . . . .	42
4.4	Détection du sujet . . . . .	44
4.5	Évaluation des performances . . . . .	48
4.5.1	Erreur quadratique moyenne . . . . .	50
4.5.2	Le taux de trame . . . . .	51
4.5.3	Analyse de la complexité des algorithmes . . . . .	51
4.5.3.1	Complexité d’algorithme Boosting en ligne . . . . .	51
4.5.3.2	Complexité d’algorithme Canny . . . . .	52
4.5.3.3	Complexité d’algorithme proposée . . . . .	52
	<b>Conclusion</b>	<b>53</b>
	<b>Bibliographie</b>	<b>55</b>

# Liste des figures

1.1	Concept de la plateforme de neurosciences sans fil utilisée pour décrire le détecteur de pics [1]. . . . .	2
1.2	Principaux blocs de construction a) Vue 3D du circuit imprimé flexible rigide plié dans sa position finale b) Headstage pliée avec la batterie installée à l'intérieur d'une enceinte de protection c) Système complet avec le module implantable prêt à être installé sur le crâne de la souris [1]. . . . .	3
1.3	Le headstage [2]. . . . .	4
2.1	Schéma fonctionnel du suivi et du comportement des petits animaux proposés [3]. . .	8
2.2	Projection d'histogramme cible. (a,b) une modification de la répartition des couleurs cible divise la cible en deux. (c,d) La cible et l'arrière-plan partagent des zones de couleur communes qui donnent à la cible une taille apparente plus grande. (e,f) Une modification dans la répartition des couleurs de la cible lui donne une taille apparente plus petite [4]. . . . .	9
2.3	Le nouveau standard pour le phénotypage comportemental et cognitif dans la cage. .	17
2.4	EthoVision XT[5]. . . . .	18
2.5	L'étude de l'interaction sociale dans EthoVision XT [5]. . . . .	19
2.6	Reconnaissance du comportement des rats dans EthoVision XT [5]. . . . .	19
2.7	Clever Sys est un fournisseur de premier plan de systèmes d'analyse comportementale automatique 24 h + [6]. . . . .	20
3.1	La caméra sur le dessus de la cage et un ordinateur portable qui lit la vidéo image par image et traite les images pour déterminer la position et le comportement de la souris.	22
3.2	L'organigramme de la solution proposée. . . . .	23
3.3	Le triangle ABC est similaire au triangle AB'C' car leurs trois angles sont égaux. . .	25
3.4	Les quatre étapes principales de suivi par un classificateur. Étant donné la position initiale de l'objet (a) à l'instant $t$ , le classificateur est évalué dans de nombreuses positions possibles à l'intérieur d'une zone de recherche voisine dans l'image $t+1$ . La carte de confiance obtenue (c) est analysée pour estimer la position la plus probable et le suivi (classificateur) est mis à jour (d) [7]. . . . .	29
3.5	Principe du boosting en ligne pour la sélection des fonctionnalités [7]. . . . .	30
3.6	Objet suivi marqué d'un rectangle jaune [7]. . . . .	31
3.7	Suivi des résultats sur une séquence (ligne 1) contenant une combinaison de changements d'apparence (mouvement, illumination, rotation hors plan, occlusion). Le comportement du traqueur proposé est analysé en considérant la carte de confiance (rangée2) et la valeur de confiance maximale en fonction du temps (rangée3) [7]. . .	32



3.8	Pour montrer la généralité de la méthode proposée, des séquences de quatre objets différents ont été capturées. Les résultats du suivi montrent que même les objets sans texture (voir la ligne 3) peuvent être suivis avec succès. De plus, l’algorithme de suivi peut gérer plusieurs objets initialisés même s’ils ont une apparence similaire (voir la ligne 4) [7]. . . . .	33
3.9	Utilisation l’algorithme MOG pour supprimer l’arrière-plan. . . . .	34
4.1	Système de suivi de mouvement. . . . .	39
4.2	Image d’échiquier utilisé pour calibrer la caméra. . . . .	41
4.3	Une image de sélection incluant le sujet dans la première image. La vidéo est en ligne. . . . .	42
4.4	Utiliser l’algorithme AdaBoost en ligne pour suivre un ballon vert dans une cage. La vidéo est en ligne. . . . .	43
4.5	Utiliser l’algorithme AdaBoost en ligne pour suivre une souris noire dans une cage et la position de la caméra est modifiée. La vidéo est hors ligne. . . . .	43
4.6	Utiliser l’algorithme AdaBoost en ligne pour suivre une souris blanche dans une cage grise. La vidéo est hors ligne[8]. . . . .	43
4.7	a) Le chemin de la souris blanche dans la cage grise. b) Le chemin de la souris noire. c) Le chemin du ballon vert. . . . .	43
4.8	Le ballon vert après la soustraction de l’arrière-plan avec l’algorithme MOG. La vidéo est en ligne. . . . .	45
4.9	La souris blanche dans la cage grise après la soustraction d’arrière-plan avec l’algorithme MOG. La caméra est placée sur le dessus de la cage. La vidéo est hors ligne [8]. . . . .	45
4.10	La souris noire après la soustraction d’arrière-plan avec l’algorithme MOG. La caméra est placée sur le côté de la cage. La vidéo est hors ligne [9]. . . . .	45
4.11	Les bords des sujets avec l’algorithme Canny. La première rangée montre les bords du ballon vert, la vidéo est en ligne. La deuxième rangée montre les bords de la souris noire, la vidéo est hors ligne [9]. La troisième rangée montre les bords de la souris blanche dans la cage grise, la vidéo est hors ligne [8]. . . . .	47
4.12	Les cadres de sélection des sujets en appliquant les algorithmes MOG and Canny. La première rangée montre le cadre de sélection du ballon vert, la vidéo est en ligne. La deuxième rangée montre le cadre de sélection de la souris noire, la vidéo est hors ligne [9]. La troisième rangée montre le cadre de sélection de la souris blanche dans la cage grise, la vidéo est hors ligne [8]. . . . .	47
4.13	Le comportement de la souris et le chemin du souris en exécutant la solution proposée. La vidéo est hors ligne [8] . . . . .	48

# Liste des tableaux

4.1	Les caractéristiques techniques de la webcam Logitech C170. . . . .	40
4.2	Les caractéristiques techniques laptop Dell Inspiron 15R N5110. . . . .	40
4.3	Taux de réussite de notre approche à trouver le sujet, et ce, dans trois cas différents, en comparant les algorithmes AdaBoost en ligne et Canny. . . . .	50

# Liste de symboles

<i>ABR</i>	Automated Behaviour Recognition system
<i>AP</i>	Action Potential
<i>ASIC</i>	Application Specific Integrated Circuit
<i>BMI</i>	Brain Machine Interface
<i>BEL</i>	Boosted Edge Learning
<i>CNN</i>	Convolutional Neural Network
<i>COTS</i>	Commercial Off-The-Shelf
<i>DoG</i>	Difference of Gaussian
<i>DooG</i>	Difference of offset Gaussian
<i>Fi – Wi</i>	Fiberless and Wireless
<i>fps</i>	frames per second
<i>IR</i>	Infrared
<i>KFC</i>	Kernelized Correlation Filters
<i>MIL</i>	Multiple Instance Learning
<i>NEO</i>	Nonlinear Energy Operator
<i>NN</i>	Neural Network
<i>PTE</i>	Power Transmission Efficiency
<i>RGB</i>	Red, Green and Blue
<i>SVM</i>	Support Vector Machine
<i>TLD</i>	Tracking, Learning and Detection
<i>WPT</i>	Wireless Power Transmission
<i>FPS</i>	Frame Per Second
<i>MSE</i>	Mean Squared Error

*Pour mes parents, Mehri Shaker et  
Noroozali Rezaei, pour qui ce  
mémoire représente autant que pour  
moi.*

*À mes frères, Masoud et Mohamad,  
qui ont toujours été une source  
constante d'amour pour moi.*

*À mes professeurs pendant toutes  
ces années d'éducation, qui m'ont  
appris à penser.*

# Remerciements

Tout d'abord, je voudrais remercier le professeur Benoit Gosselin d'avoir guidé mes recherches, de me permettre de découvrir plusieurs sujets et de me fournir les ressources nécessaires pour les mener à bien. Il m'a encouragé à faire de mon mieux et m'a donné l'occasion d'apprendre différents sujets. J'aimerais également remercier le professeur Denis Laurendeau pour ses conseils et son excellent soutien. J'ai eu l'occasion de tirer parti de ses connaissances dans mon projet. J'aimerais également remercier le membre de mon équipe, Esmael Maghsoudloo, pour sa coopération, son aide et son soutien dans le cadre de la recherche. Enfin, je suis profondément reconnaissante envers mon frère et ma famille pour leurs encouragements sans fin. Ce travail n'aurait pu être accompli sans leur soutien dévoué, leur confiance et leur amour tout au long de ma vie.

# Introduction

Dans ce chapitre, une revue de l'application de surveillance du cerveau est fournie. De plus, il y est expliqué en quoi l'optogénétique est utile dans cette application et pourquoi le mouvement des animaux doit être étudié.

## 1.1 Interface de machine à cerveau

Une interface cerveau-machine (BMI) est un dispositif à implantation chronique qui permet d'enregistrer et de stimuler les neurones cérébrales chez les patients souffrant de maladies neurologiques. Une BMI pratique doit être entièrement implantable, fournit plusieurs centaines de canaux d'enregistrement, consomme très peu d'énergie et comporte une taille extrêmement réduite. Une telle interface neurale transmet des informations précieuses sur le fonctionnement du cerveau et joue un rôle clé dans le développement de nouvelles prothèses traitant différentes maladies neurales chroniques.

Une technique de traitement du signal employée dans cette application est la détection de pic, qui peut réduire le débit de données et économiser des ressources dans une BMI. Des composants commerciaux prêts à l'emploi (COTS) pourraient être utilisés dans les BMI miniatures pour étudier les animaux en mouvement [10, 11, 12, 13, 14]. Il existe également d'autres techniques de traitement du signal, telles que la compression de données, qui minimisent aussi le débit de données [12]. Ces techniques sont généralement essentielles pour les systèmes dont le débit de données est limité dans la partie émetteur-récepteur sans fil du BMI. Un tel émetteur-récepteur polyvalent de faible puissance, peut réduire les coûts et accélérer le développement, mais limite considérablement le débit de données maximal.

La détection des pics peut être utilisée dans ces systèmes pour extraire et envoyer la partie pertinente du signal, nous donnant ainsi des facteurs de réduction des données élevés [12, 15, 16]. Avec ces détecteurs de pics, la valeur absolue du seuil positif / négatif [17], l'opérateur d'énergie non linéaire (NEO) [18], le MTEO [19], les processeurs de signaux basés sur ondelettes ou la correspondance de modèles figurent parmi les opérateurs les plus courants.

Ces opérateurs offrent de bonnes performances, mais la plupart d'entre eux ne peuvent pas être implémentés dans des systèmes intégrés de faible puissance et nécessitent une intégration dans des systèmes coûteux de circuits intégrés qui s'avèrent spécifiques à une application (ASIC).

Nous utilisons les performances in vivo d'un détecteur de potentiel d'action numérique (AP) à ressources optimisées et doté d'un seuil adaptatif basé sur une boucle de contrôle sigma-delta adaptée à une implémentation dans des systèmes embarqués à faible consommation. Les résultats in vivo ont été obtenus à partir du cortex moteur d'une souris transgénique se déplaçant librement en utilisant une plate-forme optogénétique miniature, contrainte en ressources.

Le concept de plate-forme de neuroscience sans fil avec optogénétique et enregistrement neuronal pour décrire le détecteur de pic est illustré à la figure 1.1.

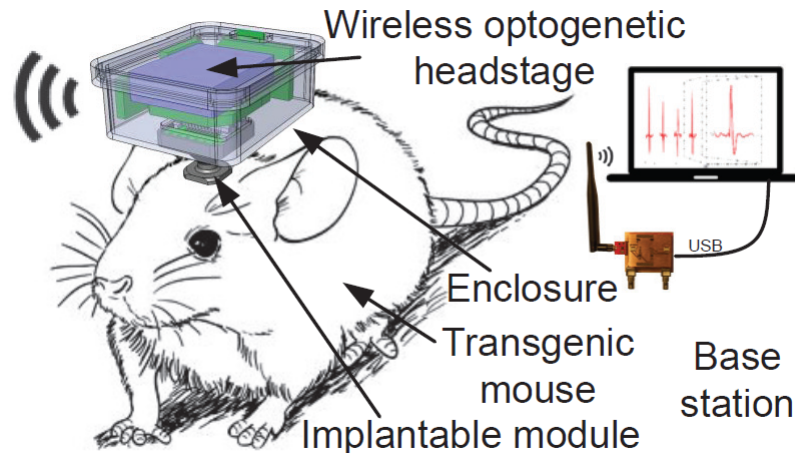


Figure 1.1 – Concept de la plateforme de neurosciences sans fil utilisée pour décrire le détecteur de pics [1].

Ce système est construit à l'aide de composants COTS, dont le schéma fonctionnel est présenté à la figure 1.2. Il est composé de cinq éléments principaux : 1) le module implantable contenant les microélectrodes, les LED et la fibre optique, 2) l'interface d'enregistrement neuronal RHD2132 d'Intan Technologies, USA, 3) un émetteur-récepteur sans fil nRF24L01 de 2,4 GHz de Nordic Semiconductors, en Norvège, comportant un débit de données maximal de 2 Mbits/s (effectif de 1,4 Mbits/s), 4) une unité de gestion de la consommation (UGP) et 5) un MCU MSP430F5328 à faible consommation de Texas Instruments, États-Unis. La famille MSP430 MCU a été utilisée dans de nombreuses applications de surveillance biomédicale [12], [20], ce qui la rend parfaitement adaptée à l'exécution de notre détecteur de pics [1].

L'optogénétique est une nouvelle méthode expérimentale. Cette méthode révolutionne les recherches en neurosciences. Elle peut éclairer le comportement des lésions cérébrales et des maladies en sélectionnant les options activer ou désactiver. L'optogénétique permet d'obtenir une modulation cellulaire rapide, contrôlée et peu invasive de l'activité cellulaire chez les animaux en mouvement, en introduisant dans les neurones des protéines sensibles à la lumière qui gèrent la conductance des ions membranaires [2].

La stimulation optique doit être associée à une lecture quantitative appropriée pour planifier la connec-

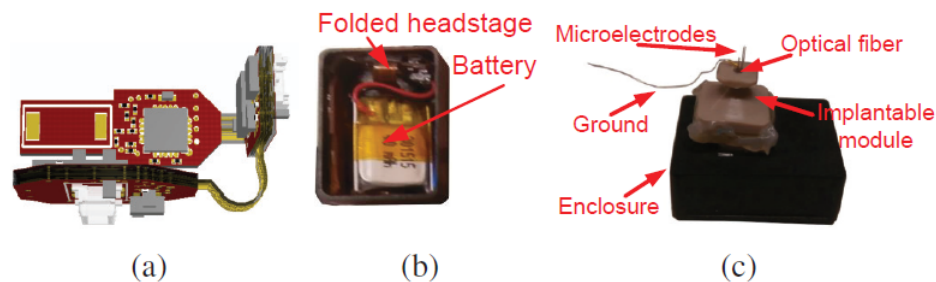


Figure 1.2 – Principaux blocs de construction a) Vue 3D du circuit imprimé flexible rigide plié dans sa position finale b) Headstage pliée avec la batterie installée à l'intérieur d'une enceinte de protection c) Système complet avec le module implantable prêt à être installé sur le crâne de la souris [1].

tivité pratique des collections de neurones demandées.

Le mélange de l'enregistrement électrique et du contrôle optique recèle un fort potentiel pour des applications cliniques telles que l'étiologie et le traitement de plusieurs maladies, comme la maladie de Parkinson, l'épilepsie, la dépression, les troubles obsessionnels compulsifs et la douleur chronique.

De nos jours, les chercheurs peuvent enregistrer et stimuler les capacités neuronales avec différents produits commerciaux. Ces produits comportent deux parties principales : 1) les enregistreurs multicanaux 2) les stimulateurs multicanaux [21],[22]. À l'aide de ces produits commerciaux, les chercheurs travaillent à la conception de nouveaux outils pour l'application neuronale.

Alors que les configurations optogénétiques actuellement disponibles utilisent de la lumière stimulante et de l'électronique d'enregistrement attachés à une source d'alimentation, les petits headstages électroniques sans fil visent à gérer des expériences avec des animaux en mouvement [23]. En effet, aucun appareil sans fil capable d'enregistrer des activités neuronales et de stimuler optiquement les neurones n'est disponible pour le moment [21],[24].

Sur ce modèle, le grand défi est la quantité importante d'énergie consommée par les circuits LED et électroniques stimulants, combinée au besoin d'un dispositif léger, qui devrait être installé sur le dessus de la tête des rongeurs. Nous utilisons une tête de lecture alimentée sans fil qui se concentre autour d'un super-condensateur et d'un système de transfert de puissance résonnant à haut rendement [25],[26]. Avec le microcontrôleur sans fil, les utilisateurs peuvent configurer l'étage pour obtenir les données électrophysiologiques capturées à partir des deux canaux A / D.



## 1.2 Motifs stimulant la lumière

Il existe deux façons de contrôler la LED : les modes constant et périodique.

1- Mode constant : la LED est constamment activée et envoie des messages, de manière à actionner les cellules du cerveau de manière constante.

2- Mode périodique : la LED est périodiquement allumée et éteinte.

L'intensité lumineuse est réglable dans les deux modes. Elle est proportionnelle au courant des LED qui est directement proportionnel à la tension sur la grille du transistor dans le circuit d'attaque LED. La headstage est représentée à la figure 1.3.

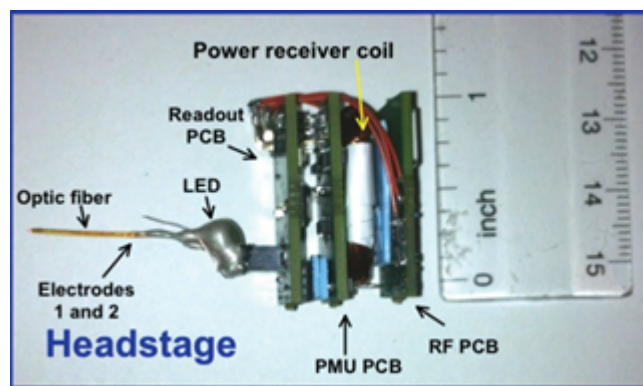


Figure 1.3 – Le headstage [2].

## 1.3 Comportement des rongeurs

Le comportement des rongeurs est important pour promouvoir la recherche en neurosciences et en pharmacologie. Pour les maladies humaines, les rongeurs sont utilisés comme modèles. L'étude du comportement des rongeurs permet d'identifier et de développer de nouveaux médicaments pour les troubles psychiatriques et neurologiques.

Pour découvrir l'effet de la variation génétique sur les changements de comportement, l'homme étiquette principalement ce comportement. L'annotation manuelle est laborieuse, sujette aux erreurs et à l'interprétation individuelle.

La recherche en neurosciences utilise un large éventail de modèles animaux pour améliorer l'analyse des relations structure / fonction dans le cerveau et le système nerveux au sens large. La majorité des recherches biomédicales et précliniques consacrées aux mécanismes de la maladie et au développement précoce des médicaments reposent sur la souris comme substitut de la condition humaine [27].

Habituellement, les rongeurs dans les laboratoires de recherche vivent en petits groupes, mangent, boivent, dorment, font leur toilette et interagissent socialement. De plus, les animaux sont placés dans ces groupes dès le sevrage et sont susceptibles de créer des comportements sociaux de haut niveau [28].

Plusieurs expériences de phénotypage actuelles supposent que l'animal est retiré de son environnement de cage et placé dans un appareil différent. Certains tests, en particulier les comportements de mesure, sont compliqués, biaisés et sous l'influence variable d'un expérimentateur [29].

Les différents environnements et la proximité de l'expérimentateur pendant le test peuvent affecter le résultat phénotypique, même si la capture des données peut être contrôlée. En effet, les souris sont des animaux sociaux dans la nature ; le logement est nécessaire pour des mesures d'activité à plus long terme, et il a été déterminé que le fait de retirer la souris de ses partenaires de cage et de la placer dans une cage différente modifiait son comportement, son bien-être général et son métabolisme. Par exemple, l'isolement social peut affecter la progression de la maladie dans de nombreux modèles de neurodégénérescence dans le cas de la souris [27].

## **1.4 Objectifs de ces travaux**

Comme nous l'avons expliqué auparavant le comportement des rongeurs est important pour promouvoir la recherche en neurosciences et en pharmacologie. Pour les maladies humaines, les rongeurs sont utilisés comme modèles.

Le but de ce travail est d'observer la souris, la suivre et découvrir ses comportements. Dans ce projet, nous présentons un système très simple, sans aucun outil complexe, pour réduire le stress de la souris afin d'obtenir le meilleur résultat sur notre test.

Nous utilisons une simple webcam qui peut être installée sur le dessus ou sur le côté de la cage pour observer la souris, la suivre dans la cage et essayons de trouver ses comportements comme marcher ou le redressement.

Dans le chapitre suivant, nous verrons que Les plus grands problèmes de suivis d'un objet sont l'éclairage, la couleur de l'objet ou les distanciations de la caméra. Puisque nous avons utilisé ces deux algorithmes ensemble, nous diminuons la probabilité de ces problèmes. Nous expliquerons notre approche en détail au chapitre 3.

## **1.5 plan du mémoire**

Dans ce mémoire, nous allons passer en revue quelques travaux récents sur des méthodes d'implémentation physique, méthodes de suivi, état de la caméra et les systèmes de suivi commerciaux pour

suivre un rongeur. Nous présentons ensuite la solution proposée en détails. En fin, nous montrons les résultats expérimentaux et évaluons notre approche.

Nous allons utiliser deux algorithmes différents pour trouver un objet dans les séquences d'images. On utilise l'algorithme AdaBoost en ligne pour suivre du sujet et l'algorithme Canny pour détecter du sujet. AdaBoost en ligne commence à suivre le sujet, si elle peut trouver le sujet, la position du sujet est gardée. Si les algorithmes Adaboost en ligne et Canny trouvent la même zone, nous estimons le comportement de la souris.

Si le sujet n'est pas trouvé par l'algorithme AdaBoost en ligne, nous utilisons l'algorithme de Canny pour trouver le sujet, alors la position obtenue ainsi que l'algorithme Canny est gardé et le comportement de la souris est estimé.

## Chapitre 2

# Revue de littérature

Pour trouver la base neuronale du comportement, il est nécessaire d'enregistrer le mouvement chez des animaux en liberté. Dans cette section, nous examinerons des approches récentes sur l'implémentation liée à la reconnaissance du comportement des rongeurs.

### 2.1 Méthodes de l'implémentation physique

#### 2.1.1 Système OSE sans fil (Fi-Wi)

Gagnon-Turcotte et al. [10, 30, 31] ont présenté un headstage figure 2.1 optogénétique sans fil capable de simuler des neurones optiques et d'enregistrer des données électrophysiologiques. Ce système utilise une petite batterie lithium-ion, pour une meilleure flexibilité. Il est construit en utilisant des matériaux commerciaux de faible coût et des composants prêts à l'emploi, avec un temps de développement réduit et un coût moindre. La stimulation lumineuse a recours à un cycle de fonctionnement et à des schémas de stimulation personnalisables de fréquence variable. La puissance optique référencée par la LED est transmise aux neurones sensibles à la lumière ciblées à l'aide de fibres optiques implantables. La headstage utilise une conception de carte de circuit imprimé pliable rigide-flexible. Ils ont utilisé une caméra de suivi du comportement à propos de laquelle aucune information n'était disponible à ce moment.

#### 2.1.2 Système basé sur une cage intelligente

Maghsoudloo et al. [32, 33] présentent un système comportant un système de la cage à transmission de puissance sans fil (WPT) qui utilise un lien résonant à 4-bobines, un système de suivi de mouvement, une headstage neuronale multicanaux et une station de base. La cage WPT se sert d'un hybrid parallel power transmitter bobines array et des segmented multi-bobines resonators permettant d'obtenir un rendement de transmission élevé (PTE) et une puissance élevée sur des distances pouvant atteindre

20 cm. Le multi-bobines power receiver (RX) utilise une RX bobines d'un diamètre de 1,0 cm et une bobines de résonateur d'un diamètre de 1,5 cm. L'efficacité moyenne de transfert de puissance de la cage WPT est de 29,4%, à une distance nominale de 7 cm, pour une fréquence porteuse de 13,56 MHz. Il dispose d'une PTE maximale et minimale de 50% et 12% le long de l'axe Z, et peut délivrer une puissance constante de 74 mW pour alimenter la headstage neurale miniature. La headstage neurale contient 1 canal de stimulation optique et 4 canaux d'enregistrement. Ils ont montré que la cage WPT hybride peut alimenter correctement la headstage en continu sans interruption, pendant que le système de suivi des mouvements permet de suivre l'activité de l'animal en temps réel pour générer une évaluation comportementale et physiologique simultanée.

## 2.2 Méthodes de suivi

### 2.2.1 Système basé sur Kinect

Wang et al. [3], de leur côté, ont travaillé sur un système capable de reconnaître et d'enregistrer cinq comportements différents sur de longues périodes : rotation, redressement, toilettage, marche, se reposer. En ayant recours à l'image de profondeur IR, la reconnaissance du comportement est possible. Cependant, le comportement reconnaissable ne comprenait que le repos, la marche et le soulèvement, et la précision de la classification était faible en raison du niveau de bruit élevé dans l'image de profondeur. La hauteur optimale pour MS-Kinect, définie par Microsoft Company, est de 110 cm.

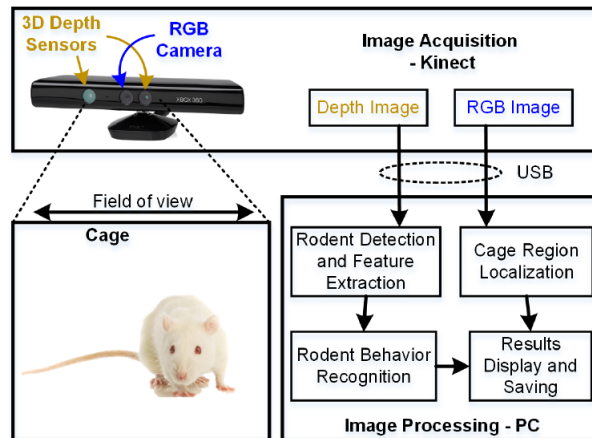


Figure 2.1 – Schéma fonctionnel du suivi et du comportement des petits animaux proposés [3].

### 2.2.2 Système basé sur Edglet

Le comportement des petits animaux, comme le rat, constitue un élément important pour les chercheurs scientifiques et précliniques. Ces derniers veulent connaître les effets des interventions sur leur vie naturelle. La surveillance des animaux peut être traitée et un grand nombre de données traitées peut conduire à de meilleurs résultats de recherche dans un temps plus court.

De nombreuses pratiques de recherche sur le comportement animal dans les systèmes numériques ont été améliorées pour augmenter la précision et réduire les erreurs causées par la notation manuelle.

Farah et al. [4] ont extrait des rongeurs d'une scène en utilisant un système de vision numérique pour suivre approximativement la cible qu'ils ont fusionnée avec trois entités, puis ils ont ajusté les limites du dispositif de suivi afin d'extraire l'animal. Ils ont utilisé une technique de segmentation pour extraire les limites de la cible par un edge arrière-plan en ligne, une soustraction et des impulsions créées par Edglets. Edglets sont des morceaux de bords séparés. L'arrière-plan en ligne est un cadre continuellement mis à jour, constitué de la collection des edglets d'arrière-plan. La projection d'un histogramme cible est affichée dans la figure 2.2.

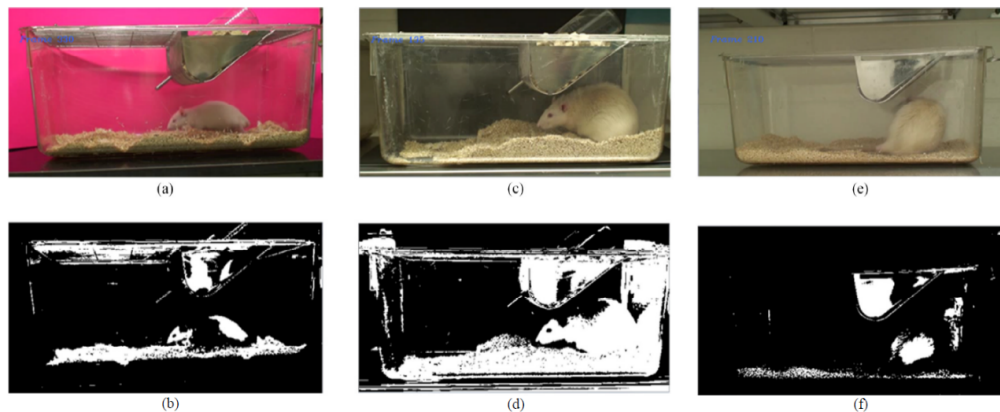


Figure 2.2 – Projection d’histogramme cible. (a,b) une modification de la répartition des couleurs cible divise la cible en deux. (c,d) La cible et l’arrière-plan partagent des zones de couleur communes qui donnent à la cible une taille apparente plus grande. (e,f) Une modification dans la répartition des couleurs de la cible lui donne une taille apparente plus petite [4].

### 2.2.3 Algorithme de k-means

Pistori et al. [34] et Goncalves et al. [35] ont utilisé l’algorithme k-means pour suivre plusieurs rats dans une cage. Pour ce faire, ils ont eu recours à un filtre à particules combiné. Ils ont calculé le paramètre du centre du rat et de l’ellipse limite avec un algorithme qui extrait les boules. Étant donné que leur algorithme utilise un seuillage de couleur simple pour la segmentation, ils ne peuvent utiliser que des souris blanches et un arrière-plan sombre. Ils ont introduit un moyen de combiner différents

modèles d'observation dans un cadre de filtre à particules. Lorsque les objets suivis se chevauchent sans enfreindre une pénalité d'exécution importante sur l'ensemble du système de suivi, ils améliorent la précision du filtre à particules. L'approche a été testée dans deux situations réelles liées au comportement animal : le suivi des larves et des souris. Ils ont montré qu'un modèle d'observation basé sur la combinaison entre analyse de composants connectés et k-mean, choisi dynamiquement, conduira à une correction des taux de classification par rapport au modèle basé uniquement sur l'analyse des composants connectés qui s'avéreront presque aussi élevés que celui des k-mean, le tout sans grande pénalité dans le temps de traitement.

#### **2.2.4 Seuil de couleur**

Ishii et al. [36] ont suivi une souris blanche sur un arrière-plan noir. Leur algorithme utilise également un seuillage de couleur simple pour la segmentation, et leur algorithme pose le même problème. Open field test s'avère l'une des méthodes de recherche les plus répandues pour analyser le comportement des rats. Ceux-ci sont libérés dans le carré ou le cercle Open field, plat et entouré de murs, et leurs paramètres de comportement tels que la distance de déplacement, la position, le nombre de redressement et le toilettage sont mesurés. L'enregistrement et l'observation sont des tâches difficiles dans les expériences et comprennent peut-être des erreurs humaines et de la subjectivité. Par conséquent, ils ont utilisé un dispositif expérimental autonome qui mesure la position des rats, leur nombre et leur toilettage en open-field. En outre, ils ont eu recours à une caméra de suivi active qui prend des images de rat à haute résolution en open-field. Pour éviter les traces manquantes du rat, un modèle de prédiction du mouvement du rat est réalisé et intégré au contrôle de cette caméra.

Nie et al. [37, 38] ont suivi une souris noire dans une cage transparente. Leur algorithme utilise d'ailleurs la même segmentation. Ils ont développé un système d'analyse de mouvement en temps réel voué à détecter un comportement de grattage de souris de laboratoire dans des expériences à long terme, ce qui permet d'obtenir une quantification de comportement automatisée pour le développement de nouveaux médicaments. Doté d'un système de vision à haute vitesse spécialement conçu, le système détecte les souris de laboratoire qui se grattent selon une méthode non invasive. Il calcule la différence image par image à une cadence de 240 images par seconde.

#### **2.2.5 Spatiotemporal 3D**

Dollar et al. [39] et Belongie et al. [40] n'ont pas utilisé la méthode de la segmentation pour trouver des comportements spécifiés chez l'homme et la souris, mais plutôt des fonctions de gradient spatio-temporel 3D.

L'utilisation de telles fonctionnalités rend le problème plus facile à gérer tout en offrant une robustesse accrue à la pose et à la variation du bruit. Dollar et al. [39] ont développé une extension de ces idées au cas spatio-temporel. Ils ont montré que les homologues 3D directs utilisant couramment des

détecteurs de points d'intérêt 2D sont inadéquats et ils ont proposé une alternative. Un algorithme de reconnaissance basé sur des données fenêtrées spatio-temporelles par ancrage de ces points d'intérêt a ainsi été conçu. Les résultats de la reconnaissance sur différents ensembles de données, y compris le comportement des humains et des rongeurs, sont présentés.

Belongie et al. [40], pour leur part, ont utilisé le Smart Vivarium, un système de surveillance automatisée continue du comportement animal. Ce système constituera un outil précieux pour les chercheurs médicaux, car il permettra d'utiliser moins d'animaux et de se servir de ces derniers de manière optimale. Ils ont d'ailleurs décrit leurs progrès dans deux domaines du projet Smart Vivarium à UC San Diego, notamment le suivi de souris multiples et la reconnaissance du comportement dans l'environnement de la cage à domicile.

### **2.2.6 Le toilettage, la rotation et le redressement**

Shi et al. [41],[42] ont proposé un cadre pour la reconnaissance du comportement du rat, comme le toilettage, la rotation et le redressement, afin de contrôler un rat robotisé en fonction du résultat de la reconnaissance d'interaction avec de vrais rats. La fonction principale de ce système repose sur l'analyse hors ligne des comportements des rats et de la reconnaissance en temps réel. L'algorithme de traitement de l'image, Labeling and Contour Finding, a été utilisé pour extraire les paramètres de la fonction du rat (zone du corps, longueur du corps, angle de rotation, rayon du corps et ellipticité). Ces paramètres sont combinés en tant que vecteurs d'entité d'entrée du SVM (Support Vector Machine) et du système d'entraînement NN (Neural Network) respectivement. Des expériences préliminaires ont d'ailleurs montré que les actions de rotation, de redressement et de toilettage pouvaient être reconnues avec un taux extrêmement élevé à la fois par SVM et par NN.

### **2.2.7 Système automatisé de reconnaissance d'animaux multiples**

Dam et al. [43] ont mis au point un système Automated Behaviour Recognition system (ABR) qui identifie les comportements des rats tels que « manger », « boire », « renifler », « arrière non soutenu », « sauter », « repos », « mur arrière », « marcher » et « se contracter ». Le système ABR ne nécessite aucune formation sur site, les seuls intrants nécessaires étant les tailles de l'animal et de la cage. ABR utilise une vue de caméra aérienne plus pratique dans les situations de laboratoire et facilite les tests à haut débit. Il est disponible sur le marché.

### **2.2.8 Classificateur de fonctionnalités multiples**

Dans [44], Dollar et al. ont extrait les arêtes d'une cible à l'aide de plusieurs classificateurs d'entités. Les traits étaient les gradients, les ondelettes Harr et la différence d'histogrammes mesurée sur des images filtrées après utilisation d'une différence de décalage gaussien (DooG) ou d'une différence de



gaussien (DoG). Ils ont proposé un algorithme d'apprentissage supervisé pour une détection de limite d'objet et de bord appelée BEL (Boosted Edge Learning). Une décision de point d'arête est prise indépendamment à chaque emplacement de l'image. une très grande partie est utilisée fournissant un contexte significatif pour chaque décision. L'algorithme sélectionne et combine un grand nombre de traits à travers différentes échelles pour apprendre un modèle discriminant en appliquant une version étendue de l'algorithme de classification Probabilistic Boosting Tree. Il n'y a pas de paramètres à ajuster et le cadre d'apprentissage est adaptatif.

### **2.2.9 Suivi de contour**

Branson et Belongie [45] ont suivi plusieurs souris dans un filtre à particules utilisé dans une cage, dont un traqueur à plusieurs boules et un traqueur de contour. Cette méthode dépend des contours des animaux pour mettre en place le traqueur, donc elle doit détecter le bord avec des traits spécifiques. Ils ont présenté un algorithme de filtrage des particules permettant de suivre de manière robuste les contours de plusieurs objets déformables à travers des occlusions sévères. L'algorithme combine un traqueur de contour avec un traqueur à plusieurs boules afin de garder le nombre requis d'échantillons. Le traqueur multi-boules recherche un espace plus petit et plus simple et utilise un modèle multi-cible naturel. Le suivi de contour fournit des résultats plus précis et repose sur des indices disponibles pendant les occlusions.

### **2.2.10 Technique de segmentation pour détecteur de bords**

Martin et al. [46] ont utilisé une technique de segmentation avec 12 000 images pour apprendre le détecteur de bord. Le détecteur a besoin d'au moins cinq minutes pour traiter une image. Ils ont donc formulé des traits qui répondent aux changements caractéristiques de couleur, de texture et de luminosité associés aux limites naturelles. Ils ont programmé un classificateur travaillant des images étiquetées par l'homme en tant que vérité fondamentale afin de fusionner ces informations de manière optimale. La sortie du classificateur fournit une probabilité postérieure aux limites pour chaque orientation et emplacement d'image. Les chercheurs ont effectué des courbes de rappel de précision en déterminant que le détecteur résultant surpassait de manière significative les approches existantes. Les résultats sont une combinaison de signaux qui peut être réalisée de manière adéquate avec un modèle linéaire simple. Un traitement correct et explicite de la texture est nécessaire pour détecter les limites des images naturelles.

### **2.2.11 Spectromètre comportemental**

Brodkin et al. [47] ont utilisé le spectromètre comportemental pour mesurer le comportement des souris, qui vise à identifier des modèles de souris et à fournir une description détaillée de leur com-

portement. Il nécessite des capteurs, comme une rangée de photocellules et un accéléromètre sous le plancher instrumenté, ce qui augmente le coût final. Ils ont exécuté un système automatisé en combinant l'analyse des vibrations et la vidéo, qui présente la description la plus détaillée du comportement de la souris disponible. Le spectromètre comportemental a permis une évaluation rapide des anomalies comportementales dans le modèle BTBR de l'autisme, le modèle irritant de la douleur inflammatoire et le modèle de contrainte du stress.

### **2.2.12 Suivi en temps réel via Boosting en ligne**

Ce traqueur fonctionne avec une version en ligne d'AdaBoost. Ce classificateur doit être formé au moment de l'exécution avec des exemples positifs (objet) et négatifs (arrière-plan) de l'objet. Le cadre de sélection initial fourni par l'utilisateur est pris comme exemple positif de l'objet et de nombreux correctifs d'image en dehors du cadre de sélection sont utilisés comme arrière-plan. En ayant un nouveau cadre, le classificateur est appliqué à chaque pixel au voisinage de l'emplacement précédent, et le score du classificateur est enregistré. Le nouvel emplacement de l'objet est celui où le score est maximal. Par conséquent, il y a désormais un autre exemple positif du classificateur. Au fur et à mesure que plus de cadres entrent, le classificateur est mis à jour avec ces données supplémentaires[7].

### **2.2.13 Suivi visuel avec l'apprentissage en ligne à plusieurs instances**

MIL Tracker est une idée similaire au traqueur Boosting. Toutefois, MIL traqueur regarde dans une petite région autour de l'emplacement actuel pour générer plusieurs exemples positifs potentiels au lieu de ne considérer que l'emplacement actuel de l'objet comme exemple positif. Dans MIL, il n'existe pas d'exemples précis, négatifs ou positifs, mais des « sacs » négatifs et positifs. Dans le sac positif, la collection d'images ne comporte pas seulement des exemples positifs. Une seule image dans le sac positif devrait être un exemple positif ! Lorsque des échantillons du voisinage de l'emplacement actuel sont placés dans le sac positif, si l'emplacement actuel de l'objet suivi n'est pas précis, il y a de fortes chances que ce sac contienne au moins une image dans laquelle l'objet est bien centré [48].

### **2.2.14 Suivi par détection avec kernels**

KFC signifie Kernelized Correlation Filters. Ce traqueur se base sur les idées MIL et Boosting. Ce suivi se base sur le fait que les multiples échantillons positifs utilisés dans le suivi MIL ont des régions qui se chevauchent. Ces données qui se chevauchent donnent lieu à certaines caractéristiques mathématiques utilisées par ce suivi pour rendre le suivi plus rapide et précis en même temps.

Utilisant la théorie bien établie des matrices circulantes, ils fournissent un lien vers l'analyse de Fourier qui ouvre la possibilité d'un apprentissage et d'une détection extrêmement rapides avec la transformation rapide de Fourier. Cela peut être fait dans le double espace des machines à noyau aussi

rapidement qu'avec les classificateurs linéaires. Ils développent des solutions analytiques pour la formation et la détection avec plusieurs types de kernels, y compris les populaires gaussiens et polynomiaux [49].

### **2.2.15 Détection automatique**

Ce suivi détecte l'objet dans le sens aller et retour dans le temps et mesure les différences entre ces deux trajectoires. En minimisant cette erreur avant-arrière, les défaillances de suivi détectées et les trajectoires sélectionnées dans les séquences vidéo seront fiables. Le traqueur est adapté aux mouvements très fluides et prévisibles lorsque l'objet est visible tout au long de la séquence. C'est tout à fait exact pour ce type de problèmes (les auteurs ont notamment montré qu'ils surpassaient la norme MIL) [50].

### **2.2.16 Suivi d'apprentissage de détection**

Le suivi suit l'objet d'une image à l'autre. Le détecteur localise toutes les apparences observées jusqu'à présent et corrige le suivi si nécessaire. Le détecteur d'apprentissage estime les erreurs et les met à jour pour éviter ces erreurs à l'avenir [51].

### **2.2.17 Apprendre à suivre avec des réseaux de régression profonds**

GOTURN est une sorte de traqueur basé sur les réseaux de neurones convolutionnels (CNN). Tout en tirant parti des avantages des suivis de CNN, GOTURN est beaucoup plus rapide grâce à une formation hors ligne sans réglage en ligne. GOTURN Tracker résout le problème du suivi d'une cible unique : à partir du libellé d'un cadre de sélection pour un objet situé dans la première image de la vidéo, nous suivons cet objet dans le reste de la vidéo. Cependant, il est assez robuste pour les changements de point de vue, les changements d'éclairage et les déformations. Les entrées de GOTURN sont deux correctifs RGB représentant les correctifs de cible et de recherche redimensionnés à  $227 \times 227$ . Les sorties de GOTURN sont les coordonnées prévues du cadre de sélection, par rapport au système de coordonnées du patch de recherche, au format coordonnées à deux points  $(X1, Y1)$  et  $(X2, Y2)$ .  $(X1, Y1)$  est le point en haut à gauche et  $(X2, Y2)$  est le point en bas à droite du cadre de sélection [52].

## **2.3 État de la caméra**

### **2.3.1 Problèmes d'éclairage**

Dans le suivi des animaux, il existe certains problèmes, comme l'environnement ou les paramètres : par exemple, les cages sont généralement empilées sur des étagères. Les corps d'animaux sont déformables et l'éclairage dans les laboratoires biomédicaux n'est pas personnalisé pour le traitement du système de vision numérique. Ces conditions limitent la position de la caméra. Pour résoudre le problème d'éclairage, le chercheur a décidé d'utiliser des capteurs infrarouges. Microsoft Kinect® est développé avec des caméras de profondeur rouges, vertes, bleues (RGB) et infrarouges (IR), ce qui constitue une bonne option à utiliser largement pour le suivi des humains et des animaux [53, 54, 55]. Le système basé sur Kinect ne dépend pas des conditions d'éclairage [56]. Un algorithme de traitement d'image basé sur Microsoft Kinect peut suivre la position de l'animal en temps réel et analyser son comportement dans des données d'image 2D et de profondeur [57].

### **2.3.2 Calibrage de la caméra**

Zhang et al. [58] ont introduit une nouvelle technique pour simplement calibrer une caméra. Il suffit que la caméra reconnaisse un motif plan montré à plusieurs (i.e. deux) orientations différentes. La caméra et le motif planaire peuvent être facilement déplacés. Le mouvement n'a pas besoin d'être connu. La distorsion radiale de la lentille est modélisée. Cette procédure, suivie d'un affinement non linéaire, basé sur le critère du maximum de vraisemblance, inclut une solution analytique. La simulation informatique et les données réelles ont été utilisées pour tester cette technique et de très bons résultats ont été obtenus.

### **2.3.3 Emplacement de la caméra entre les lignes 2D et 3D et les correspondances de points**

L'un des problèmes les plus importants dans le traitement des images est de trouver l'emplacement de la caméra, d'une image à l'autre.

Ce problème sera plus important lors de l'estimation de l'emplacement à trois dimensions à partir duquel une image a été prise par un ensemble de repères reconnus apparaissant dans l'image.

Liu et al. [59] ont mis au point une nouvelle méthode de détermination de la position de la caméra, de la correspondance entre lignes 2D et 3D. La matrice de rotation et le calcul du vecteur de translation de la caméra sont séparables. Dans cette méthode, la matrice de rotation peut être trouvée par un algorithme linéaire utilisant huit correspondances de lignes ou plus, ou par un algorithme non linéaire utilisant trois lignes ou plus. Les correspondances ponctuelles donnent les correspondances de ligne.

Par la suite, en résolvant certaines équations linéaires basées sur deux ou plusieurs correspondances ponctuelles ou trois ou plusieurs correspondances de lignes, l'approche linéaire nécessite six correspondances de points 2D à 2D ou huit correspondances de lignes 2D à 3D, tandis que l'approche non linéaire nécessite trois correspondances de lignes ou de points 2D à 3D. Si cette méthode utilise plus que le nombre minimum requis de correspondances, de bons résultats seront obtenus en présence de bruit.

### **2.3.4 Détermination de l'emplacement de la caméra à l'aide d'un ensemble de $m$ points de contrôle**

Fischler et Bolles [60] ont proposé la détermination de la position de la caméra : « Étant donné un ensemble de  $m$  points de contrôle dont les coordonnées tridimensionnelles sont connues dans un cadre de coordonnées et une image dans laquelle un sous-ensemble des points de contrôle  $m$  est visible déterminer l'emplacement (par rapport au système de coordonnées des points de contrôle) à partir duquel l'image a été obtenue. » Ils ont résolu ce problème en calculant les longueurs du rayon depuis le point focal de la caméra jusqu'aux points de contrôle du plan image. Ils ont également confirmé des résultats sur plusieurs solutions pour différents points de contrôle. Enfin, ils ont compris qu'en utilisant trois points de contrôle, il existe jusqu'à quatre solutions. Pour trouver une solution unique, six points de contrôle sont nécessaires. Keller et Tewinkel [61] ont trouvé une solution à ce problème, basée sur les techniques des moindres carrés non linéaires.

### **2.3.5 Système de robot en temps réel**

Lenz et Tsai [62] ont développé un algorithme en utilisant un système de robot en temps réel plus précis et plus simple que les méthodes de calibration disponibles. Ils disposent d'un seul joint rotatif d'un bras de robot se déplaçant pour chaque mouvement. Dans cette méthode, le calcul sera plus facile car le problème multidimensionnel se transforme en un ensemble de problèmes unidimensionnels.

## **2.4 Les systèmes de suivi commerciaux pour suivre un rongeur**

### **2.4.1 IntelliCage**

TSE Systems développe et vend des solutions de test pour l'évaluation comportementale et cognitive de petits rongeurs de laboratoire.

IntelliCage [63] vous permet d'évaluer le comportement et les performances cognitives de 16 souris ou rats individuels vivant dans un environnement social. Cette installation expérimentale favorise un comportement social naturel dans un contexte de cage biologiquement pertinente. De cette manière,

IntelliCage minimise le besoin de manipulation et d'intervention humaine, augmentant ainsi la validité de la tâche, la reproductibilité des données et assurant un niveau élevé de bien-être animal. IntelliCage est présenté à la figure 2.3.

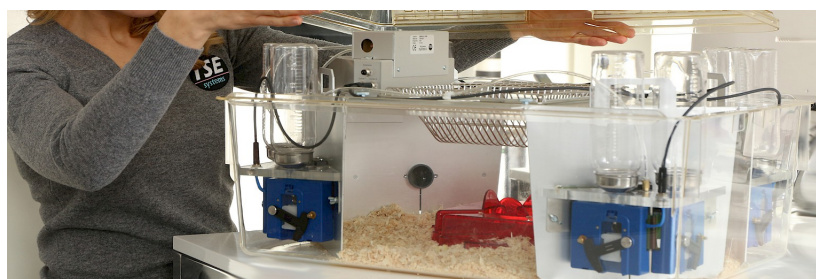


Figure 2.3 – Le nouveau standard pour le phénotypage comportemental et cognitif dans la cage.

Les animaux étiquetés par transpondeur RFID sont reconnus individuellement dans les coins de conditionnement entièrement automatisés qui évaluent le comportement spontané, l'apprentissage opérant / associatif et la mémoire. Les quatre coins sont équipés de capteurs (enregistrement continu des événements comportementaux sur 24h) et d'acteurs (permettant au système de faire un retour sur le comportement des animaux d'une manière prédéfinie). Associé au progiciel utilisant une interface utilisateur programmable, ce principe unique de capteur / acteur permet la mise en forme directe du comportement d'animaux hébergés en groupe dans un système tout-en-un en fonction de divers intérêts de recherche. Les paradigmes conçus visuellement sont entièrement automatisés, ce qui permet de tester et de surveiller efficacement et en continu les grandes cohortes d'animaux expérimentaux.

Il a été démontré qu'IntelliCage permettait le transfert de paradigmes comportementaux validés dans une configuration automatisée, en complément des procédures de test comportementaux habituelles.

#### **2.4.2 EthoVision XT**

Noldus possède un logiciel nommé EthoVision XT. Il faut un modèle qui doit être défini pour chaque expérience. Le modèle peut aider à montrer les espèces de l'animal, les dimensions et les bords de la cage. Il faut l'ajuster pour régler manuellement le gabarit et la cage. L'entreprise fournit les cages spéciales. Pour réduire la dépendance de l'éclairage et les problèmes de couleur de la fourrure, la caméra et la caméra infrarouge sont jointes [64].

EthoVision XT peut être utilisé dans de nombreux domaines liés aux neurosciences, tels que la pharmacologie de sécurité, la toxicologie, la découverte de médicaments, la psychopharmacologie, la biologie moléculaire, la génétique et les neurosciences comportementales. En outre, il peut être utilisé dans les études de bien-être animal. EthoVision XT est affiché dans la figure 2.4.

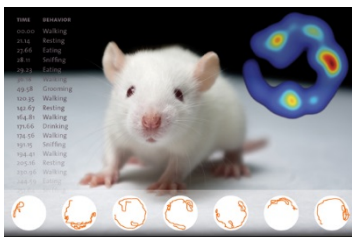


Figure 2.4 – EthoVision XT[5].

### 2.4.2.1 Base EthoVision XT

La version de base d’EthoVision XT permet de suivre et d’analyser les mouvements, l’activité et le comportement d’un animal dans une arène à la fois. Il vous guide à travers toutes les phases de votre expérience.

### 2.4.2.2 Module d’interaction sociale

Les tests d’interaction sociale chez le rat et la souris sont un paradigme couramment utilisé dans la recherche sur l’anxiété, l’agressivité, la schizophrénie et d’autres troubles psychiatriques.

Lorsque le module d’interaction sociale est utilisé en association avec le module de points multiples, les interactions nez-nez et les interactions nez-rongeurs peuvent également être analysées.

EthoVision XT offre une mesure supplémentaire de contact avec le paramètre de contact corporel. Cette méthode examine le contact entre le corps des animaux et leurs points corporels. EthoVision XT stocke ces informations concernant l’analyse des contours du corps qui se touchent pendant le suivi vidéo. L’étude de l’interaction sociale dans EthoVision XT est illustrée à la figure 2.5.

### 2.4.2.3 Module de reconnaissance du comportement de la souris

Traditionnellement, les observateurs humains sont formés pour reconnaître les comportements des rats. Cette démarche prend du temps, constitue un sujet de problèmes de répétabilité et est par nature subjective. Le module de reconnaissance du comportement du rat de VisionVision XT est inlassable. Indépendamment de la note, de la configuration ou de la durée de l’essai, les comportements suivants sont notés en permanence : manger, boire, l’élevage sans soutien, l’élevage avec soutien, le toilettage, le reniflement, la marche, le repos, les sauts et les contractions [64]. La reconnaissance du comportement des rats dans EthoVision XT est illustrée à la figure 2.6.

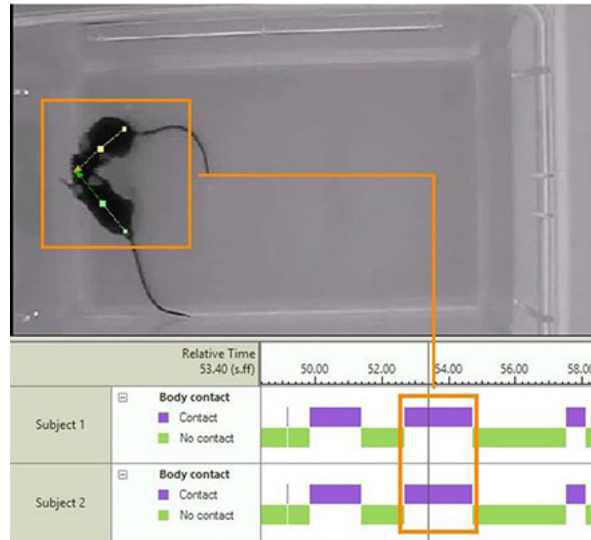


Figure 2.5 – L'étude de l'interaction sociale dans EthoVision XT [5].

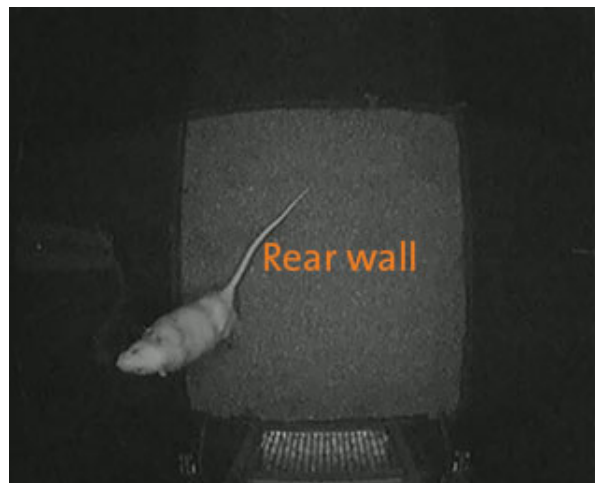


Figure 2.6 – Reconnaissance du comportement des rats dans EthoVision XT [5].

### 2.4.3 CleverSys

CleverSys [6], [65] a développé une technique pour extraire le rongeur en premier plan par une soustraction d'arrière-plan à base de couleurs. Pour suivre le rongeur, cette méthode trouve le centre de masse du rongeur. En outre, il nécessite un système d'éclairage adapté et un fond blanc [66].

Cette technologie de reconnaissance du comportement basée sur la vidéo présente des outils complets pour la recherche préclinique, y compris tout le matériel nécessaire pour analyser divers modèles de comportement.

Ce logiciel fonctionne dans plusieurs vues : il est efficace pour trouver le meilleur angle afin de saisir les informations sur les parties du corps les plus pertinentes pour le comportement qui vous intéresse.



Les systèmes sont classés en quatre groupes distincts : vue de dessus, vue latérale, vue ventrale et vue double ou stéréo. Chaque angle d'observation est spécifique à la catégorie de comportements à analyser [66]. Vue de dessus : Les systèmes basés sur la vue de dessus sont très utiles dans les tests de locomotion, d'opération, d'apprentissage et de mémoire et de comportement social. Bien que la vue de dessus ne soit pas aussi avancée que la vue latérale, la capacité à identifier le centre de masse et à reconnaître avec précision la tête et la queue de l'animal s'avère inégalée [66].

Vue latérale : Les systèmes de vue latérale peuvent segmenter l'animal en différentes parties du corps et fournir d'excellentes informations sur l'animal et son comportement sans contrainte [66].

Vue ventrale : Cette méthode est particulièrement utilisée pour analyser les données de mouvements. Que la marche soit nécessaire sur un tapis roulant ou une association libre dans la piste ou la boîte de marche libre, les systèmes de vue ventrale constituent un outil d'analyse du comportement de premier niveau [66].

Vue double : Il s'agit de la collecte d'informations à partir de la vue de dessus et de côté. Cela permet de reconnaître plusieurs animaux dans une arène avec des informations sur les parties du corps combinant à la fois les comportements familiaux et sociaux [66]. Clever Sys est illustré à la figure 2.7.



Figure 2.7 – Clever Sys est un fournisseur de premier plan de systèmes d'analyse comportementale automatique 24 h + [6].

Clever Sys a adopté plusieurs approches, en fonction du périphérique.

Une approche simple consiste à envoyer un signal de synchronisation commun qui est un signal de référence. Les deux données de comportement sont extraites de Clever Sys. Les produits et les données physiologiques de Clever Sys peuvent se référer à ce signal de synchronisation commun pour rechercher une corrélation et une synchronisation. C'est une approche peu coûteuse et fiable. Il s'agit du mode de fonctionnement « intégration libre » dans l'intégration et la synchronisation des données de comportement et de physiologie.

Clever Sys a également évolué vers le haut. Il permet de synchroniser les comportements de l'animal de laboratoire et ses données de physiologie réelle dans un « mode d'intégration serré ». Ce mode est

généralement passé en suivant deux approches différentes.

Lorsque les fournisseurs d'instruments de physiologie ont fourni une interface de programmation d'application (API), Clever Sys a alors disposé du logiciel avec l'API pour accéder aux données de physiologie du fournisseur, à l'intégration et au synchroniseur avec les données de comportement. L'intégration transparente avec Pinnacle Technologies, qui fournit les API, est un exemple.

Lorsque les fournisseurs utilisent des physiothérapeutes ou des utilisateurs, ils enregistrent leur sortie de signal électrique, leur logiciel et leur matériel, ils sont mémorisés, ils sont synchronisés et ils se synchronisent avec les données de comportement. Les instruments de physiologie de Triangle Biosystems en sont un exemple [6].

## **2.5 Conclusion**

Dans ce chapitre, nous avons examiné les systèmes de suivi récents et le nouveau système de suivi commercial comme IntelliCage, EthoVision XT, CleverSys. Dans le chapitre suivant, nous avons proposé notre approche en détails.

Nous allons utiliser deux algorithmes différents ensemble, Adaboost en ligne et Canny. Nous savons que les outils commerciaux existants sont trop coûteux, nous allons donc expliquer un système qui ne coûte pas cher et qui n'a pas besoin d'exigences compliquées.

## Chapitre 3

# Présentation de la solution proposée

### 3.1 Résumé

Ce mémoire présente une méthode fiable pour suivre une souris et découvrir son comportement dans une cage spéciale du laboratoire.

Le système comprend une base réglable, une caméra standard, une cage et un ordinateur portable.

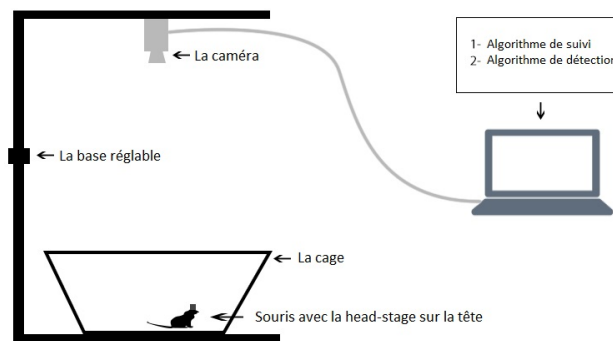


Figure 3.1 – La caméra sur le dessus de la cage et un ordinateur portable qui lit la vidéo image par image et traite les images pour déterminer la position et le comportement de la souris.

Nous présentons une méthode précise qui n'est pas dispendieuse, car elle n'a pas besoin de matériel complexe, la cage, la caméra ou tout matériel spécial et elle fonctionne assez rapidement.

Nous pensons que la meilleure façon de suivre une souris et de connaître son comportement est d'utiliser des algorithmes de détection du sujet et de suivi du sujet ensemble, car le suivi peut aider lorsque la détection ne peut pas trouver le sujet et vice-versa.

On utilise l'algorithme AdaBoost en ligne pour suivre du sujet et l'algorithme Canny pour détecter du sujet.

Nous commençons à suivre le sujet avec AdaBoost en ligne, et si cet algorithme peut trouver le sujet,

nous gardons la position du sujet pour dessiner le chemin du suivi. Si les algorithmes Adaboost en ligne et Canny trouvent la même zone, nous estimons le comportement de la souris.

Si l’algorithme AdaBoost en ligne ne peut pas trouver le sujet, nous utilisons l’algorithme de Canny pour trouver le sujet, puis nous gardons la position obtenue ainsi que l’algorithme Canny pour dessiner le chemin du suivi et nous estimons le comportement de la souris.

Nous expliquerons tous les algorithmes de manière détaillée dans les sections suivantes.

La figure 3.2 montre l’organigramme de la solution proposée.

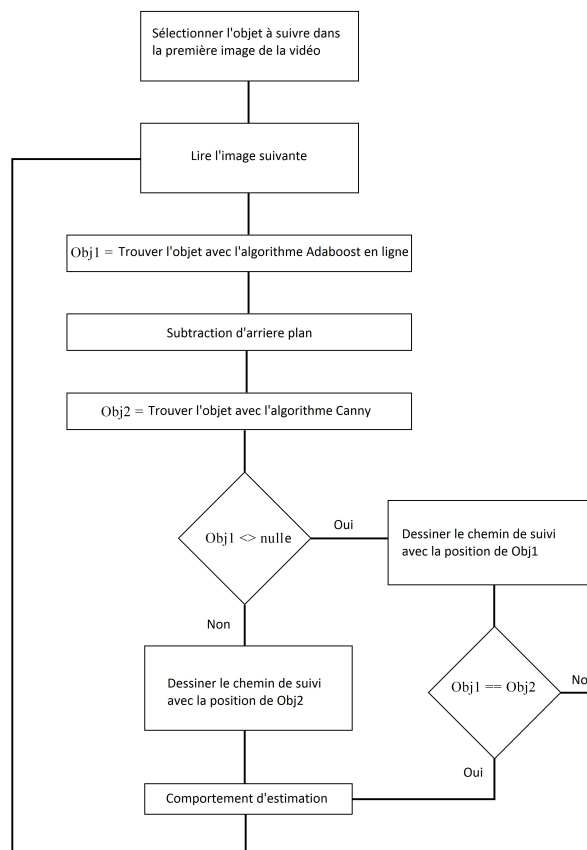


Figure 3.2 – L’organigramme de la solution proposée.

La figure 3.1 montre la caméra sur le dessus de la cage et un ordinateur portable qui lit la vidéo image par image et traite les images pour déterminer la position et le comportement de la souris.

## 3.2 Calibration de la caméra

Depuis que nous avons une caméra de modèle sténopé avec distorsion radiale, nous devons la calibrer avant de l'utiliser.

La calibration de la caméra, introduite sous le nom de résection de la caméra, détermine les paramètres d'un objectif et du capteur d'image d'une image ou d'une caméra vidéo. Nous utiliserons ces paramètres pour trouver la position du sujet en unités internationales.

Les paramètres de l'appareil photo incluent les coefficients extrinsèques, intrinsèques et de distorsion. Après avoir calibré une caméra, pour estimer la précision des paramètres calculés, nous pouvons tracer le modèle de calibration et les emplacements correspondants de la caméra, déterminer les erreurs de projection, calculer les erreurs de paramètre, utiliser le calibre de caméra pour afficher la calibration de la caméra et évaluer la précision des paramètres estimés.

Les paramètres extrinsèques décrivent une transformation rigide d'un système de coordonnées 3D du monde en système de coordonnées de la caméra 3D. Les paramètres intrinsèques décrivent une transformation projective des coordonnées de la caméra 3-D en coordonnées de l'image 2-D.

### 3.2.1 Paramètres extrinsèques

Les paramètres extrinsèques comprennent une rotation,  $R$ , et une translation,  $t$ . L'origine du système de coordonnées de la caméra est à son centre optique et ses axes  $x$  et  $y$  représentent le plan de l'image.

### 3.2.2 Paramètres intrinsèques

Les paramètres intrinsèques consistent en la distance focale  $(f_x, f_y)$ , le centre optique, appelé point principal  $(c_x, c_y)$ , et le coefficient de biais. Il est exprimé sous forme de matrice 3x3 :

$$CameraMatrix = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (3.1)$$

### 3.2.3 Distorsion dans la calibration de la caméra

La matrice de la camera ne prend pas en compte la distorsion de l'objectif, parce qu'un appareil photo à sténopé idéal n'a pas d'objectif. Pour représenter avec précision une caméra réelle, le modèle de caméra inclut la distorsion radiale et tangentielle de l'objectif.

### 3.2.3.1 Radial Distortion

La distorsion radiale se produit lorsque les rayons lumineux se plient plus près des bords d'une lentille qu'au centre optique.

Distorsion radiale modélisent ce type de distorsion. Les points déformés sont notés comme  $(x_{distorted}, y_{distorted})$ .

$$\begin{aligned} x_{distorted} &= x(1 + k_1 * r^2 + k_2 * r^4) \\ y_{distorted} &= y(1 + k_1 * r^2 + k_2 * r^4) \end{aligned} \quad (3.2)$$

$x, y$  : Emplacements de pixels non déformés.  $x$  et  $y$  sont en coordonnées d'image normalisées. Les coordonnées d'image normalisées sont calculées à partir des coordonnées de pixels en les traduisant vers le centre optique et en les divisant par la distance focale en pixels.

$k_1, k_2$  : Coefficients de distorsion radiale de la lentille.

$r^2 : x^2 + y^2$

Pour trouver la position du sujet en mm, nous utilisons la règle en forme de triangle similaire.

Les triangles sont équiangulaires si les angles d'un triangle sont égaux à ceux d'un autre triangle. Ces triangles ont la même forme avec des tailles différentes. Les triangles équiangulaires sont nommés triangles similaires.

Par exemple, sur la figure 3.3, le triangle ABC est similaire au triangle AB'C' car leurs trois angles sont égaux.

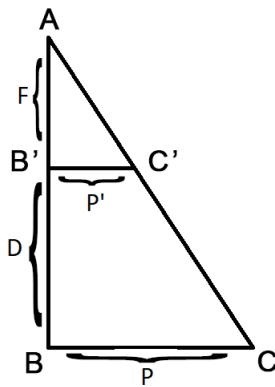


Figure 3.3 – Le triangle ABC est similaire au triangle AB'C' car leurs trois angles sont égaux.

$F$  est la distance focale en mm,  $D$  est la distance de la caméra au sujet,  $P$  est la position du sujet en mm et  $P'$  est la position du sujet en pixels. On obtient  $P$  comme cela est exposé ci-dessous :

$$f/P' = (F + D)/P \quad (3.3)$$

$f$  est la distance focale en pixels obtenue avec la calibration de la caméra par 3.1.

### 3.3 Définition de détection du sujet et de suivi du sujet

Dans les techniques de vision numérique, la détection du sujet consiste à balayer et à rechercher un sujet dans une image ou une vidéo (qui n'est qu'une séquence d'images), mais la localisation du sujet dans des images successives d'une vidéo est appelée suivi. Dans ce projet, on a implémenté la détection du sujet et le suivi du sujet, car le suivi peut aider lorsque la détection échoue et vice versa. Avec cette approche, nous pouvons suivre et détecter tout type de sujet (souris, headstage, ou par exemple un ballon). Il n'y a pas de dépendance au type de camera.

#### 3.3.1 Suivi du sujet

Dans le suivi, le but est de rechercher un objet dans l'image en cours, en fonction de l'objet suivi avec succès dans toutes les images précédentes. Puisque l'objet est suivi jusqu'à l'image en cours, son déplacement est connu. De plus, l'objet apparaît dans chacune des images précédentes. Il est donc possible de créer un modèle d'apparence qui encode l'objet. Ce modèle d'apparence peut être appliqué pour rechercher dans un petit voisinage de l'emplacement prédit par le modèle de mouvement afin de prédire plus précisément l'emplacement de l'objet. Le modèle de mouvement prédit l'emplacement approximatif de l'objet. Le modèle d'apparence affine cette estimation pour fournir une estimation plus précise basée sur l'apparence. Cependant, l'apparence de l'objet peut changer. Pour résoudre ce problème, dans de nombreux traqueurs modernes, ce modèle d'apparence est un classificateur qui est entraîné en ligne.

Le classificateur classe une région d'image rectangulaire en objet ou en arrière-plan. Le classificateur utilise un patch d'image en entrée et renvoie une marque comprise entre 0 et 1 pour indiquer la probabilité que la région d'image contienne l'objet. Lorsqu'il est certain que le correctif d'image est d'arrière-plan, la marque est égale à 0 et lorsqu'il est certain que le correctif est l'objet, il est égal à 1.

En apprentissage automatique, le mot « en ligne » fait référence à des algorithmes formés au moment de l'exécution. Un classificateur hors ligne peut nécessiter des milliers d'exemples pour former un classificateur, mais un classificateur en ligne est formé en utilisant très peu d'exemples au moment de l'exécution.

Les algorithmes de suivi sont rapides; il est indéniable que, grâce au suivi d'un objet détecté dans la trame précédente, les informations sur l'apparence de l'objet seront connues. L'emplacement dans le cadre précédent, la direction et la vitesse de son mouvement seront également connus. Ainsi, dans le cadre suivant, l'outil de suivi utilise toutes ces informations pour prédire l'emplacement de l'objet dans l'image suivante et effectue une petite recherche autour de l'emplacement attendu de l'objet pour

localiser l'objet avec précision.

Dans ce projet, nous utilisons une version en ligne d'AdaBoost. Ce classificateur doit être formé à l'exécution à l'aide d'exemples positifs (objet) et négatifs (arrière-plan) de l'objet. Le cadre de délimitation initial fourni par l'utilisateur est pris comme exemple positif de l'objet, et de nombreux correctifs d'image en dehors du cadre de sélection sont utilisés, tels que l'arrière-plan. En ayant une nouvelle trame, le classificateur est exécuté sur chaque pixel au voisinage de l'emplacement précédent et la marque du classificateur est enregistré. L'objet du nouvel emplacement est celui où la marque est le plus grand. Par conséquent, il y a désormais un autre exemple positif du classificateur. Au fur et à mesure que d'autres trames entrent, le classificateur est mis à jour avec ces données supplémentaires.

### **3.3.1.1 Suivi en temps réel via Boosting en ligne**

Le suivi d'objets robuste et efficace dans des environnements complexes est essentiel pour une variété d'applications, notamment l'interaction homme-machine, la vidéosurveillance ou la conduite autonome. Par conséquent, un grand défi consiste à concevoir des méthodes de suivi visuel robustes capables de faire face aux variations inévitables et se produisant dans des scènes naturelles telles que les changements de forme, les changements d'éclairage, la réflexion de l'objet ou les modifications de point de vue. De plus, le suivi de l'échec ou du succès peut également dépendre de la séparabilité d'un objet par rapport à son arrière-plan. Si l'objet est très distinctif, un simple traqueur peut déjà satisfaire aux exigences. Néanmoins, avoir des objets similaires à l'arrière-plan nécessite des fonctionnalités plus avancées.

Par conséquent, les suiveurs doivent pouvoir gérer toutes les variations possibles de changement d'apparence de l'objet cible et de l'autre ; ils peuvent gérer l'encombrement de l'arrière-plan de manière fiable [7].

Plusieurs approches ont été développées pour répondre à ces deux exigences principales en matière de suivi. Pour faire face aux variations d'apparence de l'objet cible pendant le suivi, l'adaptabilité améliore les méthodes de suivi existantes afin de pouvoir s'adapter progressivement aux modifications de l'environnement de suivi spécifique. Les méthodes ou les représentations adaptatives obtiennent une invariance par rapport aux différentes variations. Plusieurs algorithmes classiques ont été modifiés pour pouvoir ajuster l'algorithme de suivi à l'environnement de suivi. La méthode classique de suivi des sous-espaces de Black et al. [67] a été améliorée par une mise à jour incrémentielle des sous-espaces dans [68, 69].

Dans [68], il s'agit de montrer le problème général d'adaptation comme un problème d'adaptation de sous-espace, où les variations d'apparence visuelle sur une courte échelle de temps sont représentées par un sous-espace linéaire. En revanche, [70] recommande une sélection en ligne de caractéristiques locales de type Haar pour manipuler les variations d'apparence possibles.



En ce qui concerne le problème de l'adaptation en ligne, de nombreuses approches ont proposé d'utiliser des informations sur l'arrière-plan pour accroître la robustesse du suivi [70, 71, 72, 73, 74]. Collins et Liu [74] ont mis l'accent sur l'importance de l'apparence de l'arrière-plan. Ils supposent que l'espace qui distingue le mieux l'arrière-plan et l'objet s'avère le meilleur espace à utiliser pour le suivi. L'idée de considérer le problème de suivi comme un problème de classification entre l'arrière-plan et l'objet a conduit à des travaux supplémentaires [71, 73, 75] appliquant des classificateurs bien connus au problème de suivi.

Grabner et al. [7] ont réalisé un suivi en utilisant une machine à vecteurs de support qui apprend hors ligne à distinguer l'arrière-plan de l'objet. Encore une fois, le suivi est expliqué comme un problème de classification binaire, où un ensemble de classificateurs faibles est combiné à un classificateur puissant capturant l'AdaBoost pour sélectionner des caractéristiques discriminantes pour le suivi. Pour atteindre la robustesse contre les variations d'apparence, de nouveaux classificateurs faibles peuvent être ajoutés à l'ensemble. Néanmoins, cela se fait dans un mode de traitement par lots utilisant une amplification hors ligne par lots après la collecte de nouveaux exemples de formation. De plus, les fonctionnalités sont assez limitées. Ils ont présenté une approche de suivi des objets en temps réel basée sur une nouvelle version en ligne de l'algorithme AdaBoost.

L'algorithme effectue la mise à jour en ligne de l'ensemble des caractéristiques de l'objet cible pendant le suivi, ce qui lui permet de faire face aux changements d'apparence de l'objet. De plus, le classificateur formé en ligne utilise le contexte environnant comme des exemples négatifs dans la mise à jour et devient très robuste par rapport au problème de la dérive [76]. Cette mise à jour négative permet à l'algorithme de choisir les caractéristiques les plus discriminantes entre l'arrière-plan et l'objet puisque la méthode peut traiter à la fois les variations d'apparence de l'objet et les différents arrière-plans.

L'algorithme, qui utilise uniquement des informations sur les valeurs de gris, est capable de fonctionner en temps réel car la formation se fait simplement en mettant à jour le modèle avec les exemples négatifs et positifs de la trame actuelle, la tâche de classification et pour assurer la robustesse en mettant à jour en permanence le classifieur actuel de l'objet cible. Le récupérateur suppose que l'objet cible a été détecté. Cette zone d'image est supposée être un échantillon d'images positives pour le traqueur. Dans le même temps, des exemples négatifs sont extraits en prenant des zones de la même taille que la fenêtre cible de l'arrière-plan environnant. Ces échantillons permettent d'effectuer plusieurs itérations de l'algorithme de boosting en ligne pour obtenir un premier modèle déjà stable. Ces itérations ne sont nécessaires que pour l'initialisation du traqueur.

L'étape de suivi est basée sur la méthode classique de suivi de modèle [77]. Le classificateur actuel est évalué dans une région d'intérêt et obtient une valeur de confiance pour chaque position. La carte de confiance est analysée, puis la fenêtre cible est déplacée vers le nouvel emplacement du maximum. Une procédure de décalage moyen [78] peut être appliquée pour une détection maximale. L'utilisation d'un modèle de mouvement pour l'objet cible permettrait la réduction de la fenêtre de recherche. Une

fois que les objets ont été reconnus, le classificateur doit être mis à jour pour s'adapter aux modifications possibles de l'apparence de l'objet cible et devenir discriminant par rapport à un arrière-plan différent. La zone cible actuelle est utilisée comme mise à jour positive du classificateur tandis que les zones voisines représentent à nouveau les échantillons négatifs. Au fur et à mesure que de nouvelles images apparaissent, la procédure est répétée et le classificateur est capable de s'adapter aux changements d'apparence possibles et devient robuste en cas de fouillis d'arrière-plan. Le classificateur se concentre sur l'objet cible actuel tout en essayant de distinguer la cible de son voisin. Les quatre étapes principales de suivi par un classificateur sont affichées dans la figure 3.4.

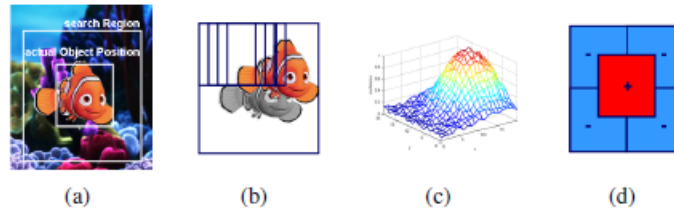


Figure 3.4 – Les quatre étapes principales de suivi par un classificateur. Étant donné la position initiale de l'objet (a) à l'instant  $t$ , le classificateur est évalué dans de nombreuses positions possibles à l'intérieur d'une zone de recherche voisine dans l'image  $t+1$ . La carte de confiance obtenue (c) est analysée pour estimer la position la plus probable et le suivi (classificateur) est mis à jour (d) [7].

### 3.3.1.2 AdaBoost en ligne

L'algorithme d'amplification en ligne permet de générer des classificateurs qui peuvent être mis à jour efficacement en appliquant des échantillons de manière incrémentale. Pour mieux comprendre cette méthode, les termes suivants sont définis.

**Classificateur faible :** un classificateur faible doit seulement être légèrement meilleur que l'estimation aléatoire (c'est-à-dire que pour un problème de décision binaire, le taux d'erreur doit être inférieur à 50%).

L'hypothèse  $h^{weak}$  générée par un classificateur faible correspond à une caractéristique et est obtenue en appliquant un algorithme d'apprentissage défini.

**Sélecteur :** ayant un ensemble de  $M$  classificateurs faibles avec l'hypothèse  $h^{weak} = h_1^{weak}, \dots, h_M^{weak}$  faible, un sélecteur en sélectionne un exactement.

$$h^{sel}(x) = h_M^{weak}(x) \quad (3.4)$$

où  $M$  est déterminé selon un critère d'optimisation. L'erreur estimée  $e_i$  de chaque classificateur faible  $h_i^{weak} \in h^{weak}$  est utilisé et  $M = argmin(e_i)$

**Classificateur fort :** Étant donné un ensemble de  $N$  classificateurs faibles, un classificateur fort est calculé par une combinaison linéaire de sélecteurs. De plus, la valeur  $conf(0)$  peut être interprétée comme une mesure de confiance du classificateur fort.

$$hStrong(x) = sign(conf(x))$$

$$conf(x) = \sum_{n=1}^N \alpha_n \cdot h_n^{sel}(x) \quad (3.5)$$

$\alpha_n$  est la meilleure erreur d'estimation des erreurs  $e_{n,1}, \dots, e_{n,m}$  du classifieur faible.

L'idée principale du boosting en ligne est l'introduction des *sélecteurs*. Ils sont initialisés de manière aléatoire et chacun d'entre eux contient une piscine de fonctionnalités distinctes des classificateurs faibles. Lorsqu'un nouvel échantillon d'apprentissage apparaît, chaque sélecteur de classificateurs faibles est mis à jour. Le meilleur classificateur faible qui présente l'erreur la plus faible est sélectionné par le sélecteur, où l'erreur de classificateur faible est estimée à partir des échantillons vus récemment. La complexité est définie par le nombre de sélecteurs. Le principe du boosting en ligne est affiché dans la figure 3.5.

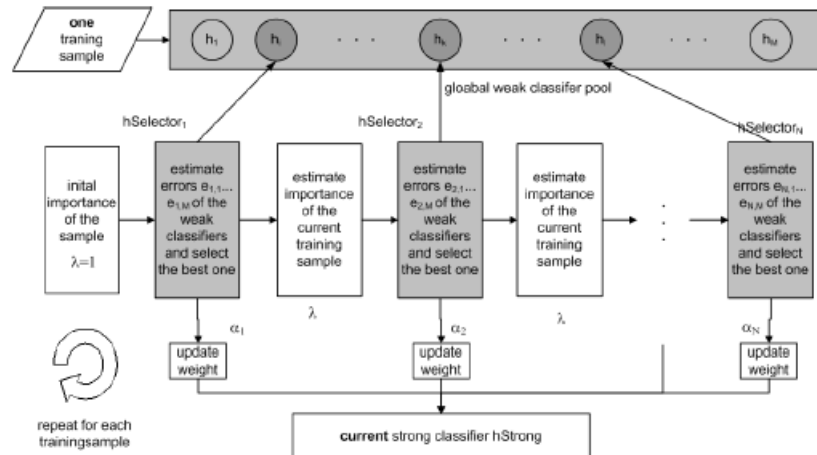


Figure 3.5 – Principe du boosting en ligne pour la sélection des fonctionnalités [7].

### 3.3.1.3 Caractéristiques

Trois différents types de caractéristiques sont utilisés pour générer des hypothèses faibles, soit les caractéristiques de Haar, telles que dans Viola et Jones [79], les histogrammes d'orientation [80, 81, 82] et une version simple des schémas binaires locaux (LBP)[83]. Notez que tous les calculs de types d'entités peuvent être effectués très efficacement en utilisant des images intégrales et des histogrammes intégraux comme les structures de données [81]. Bien que le suivi soit toujours en temps réel, cela permet de faire des correspondances exhaustives. La distribution de probabilité pour les échantillons

négatifs et les échantillons positifs est modélisée pour obtenir une hypothèse à partir de ces caractéristiques. Par une technique de filtrage de Kalman, une estimation de densité de probabilité est effectuée. Pour les ondelettes classiques de type Haar, un simple seuil et un critère de décision bayésien sont utilisés comme algorithmes d'apprentissage. Pour les types d'entités basés sur l'histogramme (LBP et histogrammes d'orientation), l'apprentissage le plus proche est utilisé.

### 3.3.1.4 Expériences

La section de test est séparée en deux parties. Premièrement, nous effectuons des expériences montrant trois propriétés spécifiques de notre méthode de suivi et, deuxièmement, nous présentons des résultats sur des séquences publiques disponibles pour les comparer à d'autres méthodes de suivi. En identifiant manuellement l'objet cible dans la première image, chaque tâche de suivi a été initialisée. Le suivi a été appliqué à des centaines de séquences d'images constituées. La vitesse dépend de la taille de la zone de recherche, définie le traqueur en augmentant la région cible d'un tiers dans chaque direction.

### 3.3.1.5 de manière adaptative

Pour expliquer le traqueur de manière adaptative et sa capacité à sélectionner les meilleures fonctionnalités en fonction de l'arrière-plan, une scène est traitée où l'objet cible est un petit pièce texturé, voir la figure 3.6. L'objectif du scénario est de montrer comment la méthode de sélection de fonctionnalités en ligne proposée peut adapter son modèle au problème de suivi actuel. Étant donné que cette approche semble convenir à des caractéristiques permettant de mieux distinguer l'objet de l'arrière-plan, l'arrière-plan passe d'homogène à texturé.

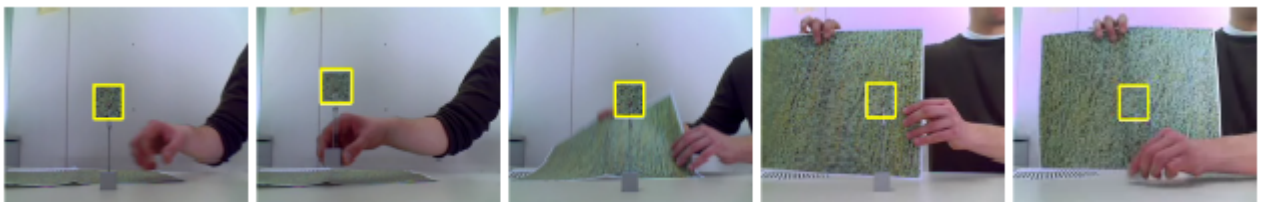


Figure 3.6 – Objet suivi marqué d'un rectangle jaune [7].

### 3.3.1.6 Robustesse

Un traqueur fiable est un traqueur capable de gérer différents changements d'apparence (occlusions, changements d'éclairage, mouvements, rotations hors-plan) qui se produisent dans des scènes naturelles.

La figure 3.7 montre le comportement de la méthode proposée dans le cas de telles interférences de l'objet cible. La séquence montre un verre qui est initialement bloqué par un papier. Par la suite, il est déplacé derrière lui avec des modifications d'éclairage supplémentaires et, enfin, des changements de point de vue de l'objet cible. L'augmentation continue de la valeur de confiance maximale dans la phase initiale (voir la ligne 3, figure 3.7) signifie que le traqueur s'adapte à l'objet cible en ce qui concerne son arrière-plan. Néanmoins, si l'apparence de l'objet cible change ou que l'arrière-plan de la cible devient différent, le traqueur doit mettre à jour ses caractéristiques, ce qui se traduit par des oscillations de confiance maximale (voir la rangée 3, figure 3.7) et une carte de confiance aplatie. Le mouvement de la cible de suivi est représenté par un pic décalé dans la carte de confiance. Le traqueur peut gérer tous les types de variations d'apparence des objets cibles et peut toujours trouver les meilleures caractéristiques discriminantes pour distinguer l'objet cible de l'arrière-plan environnant. Par conséquent, l'adaptation est entièrement liée à la robustesse du traqueur.

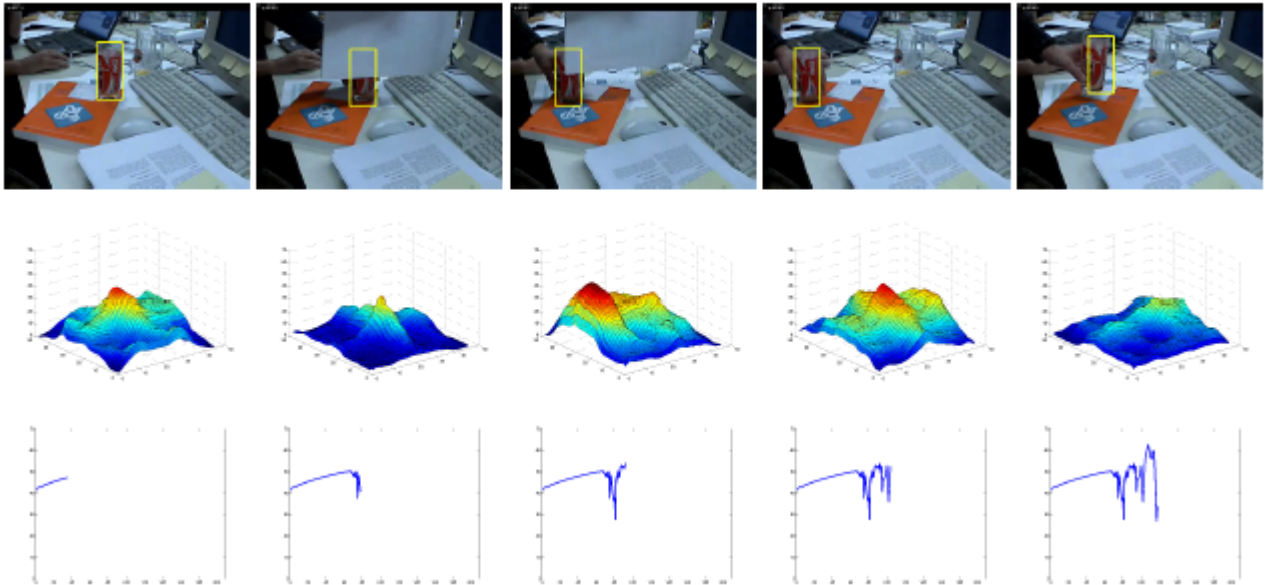


Figure 3.7 – Suivi des résultats sur une séquence (ligne 1) contenant une combinaison de changements d'apparence (mouvement, illumination, rotation hors plan, occlusion). Le comportement du traqueur proposé est analysé en considérant la carte de confiance (rangée2) et la valeur de confiance maximale en fonction du temps (rangée3) [7].

### 3.3.1.7 Généralité

Pour décrire la généralité, quatre séquences différentes avec des objets cibles différents sont utilisées (voir figure 3.8). La première séquence de la ligne 1 montre le suivi d'une petite webcam dans une scène encombrée. Cependant, l'objet cible contient peu de texture et les modifications de pose sont suivies de manière robuste. La deuxième ligne de la figure 3.8 montre les résultats du suivi du visage. De plus, même les variations de la tête ne confondent pas le traqueur, ce qui montre que le traqueur s'adapte aux nouvelles apparences de l'objet cible. La ligne 3 montre que seule une petite texture



de l'objet est suffisante pour le suivi. La ligne 4 montre le comportement de plusieurs objets cibles similaires. Même si les objets se chevauchent de manière significative, les traqueurs ne sont pas confus en montrant que les classificateurs ont appris à distinguer l'objet spécifique de son arrière-plan.

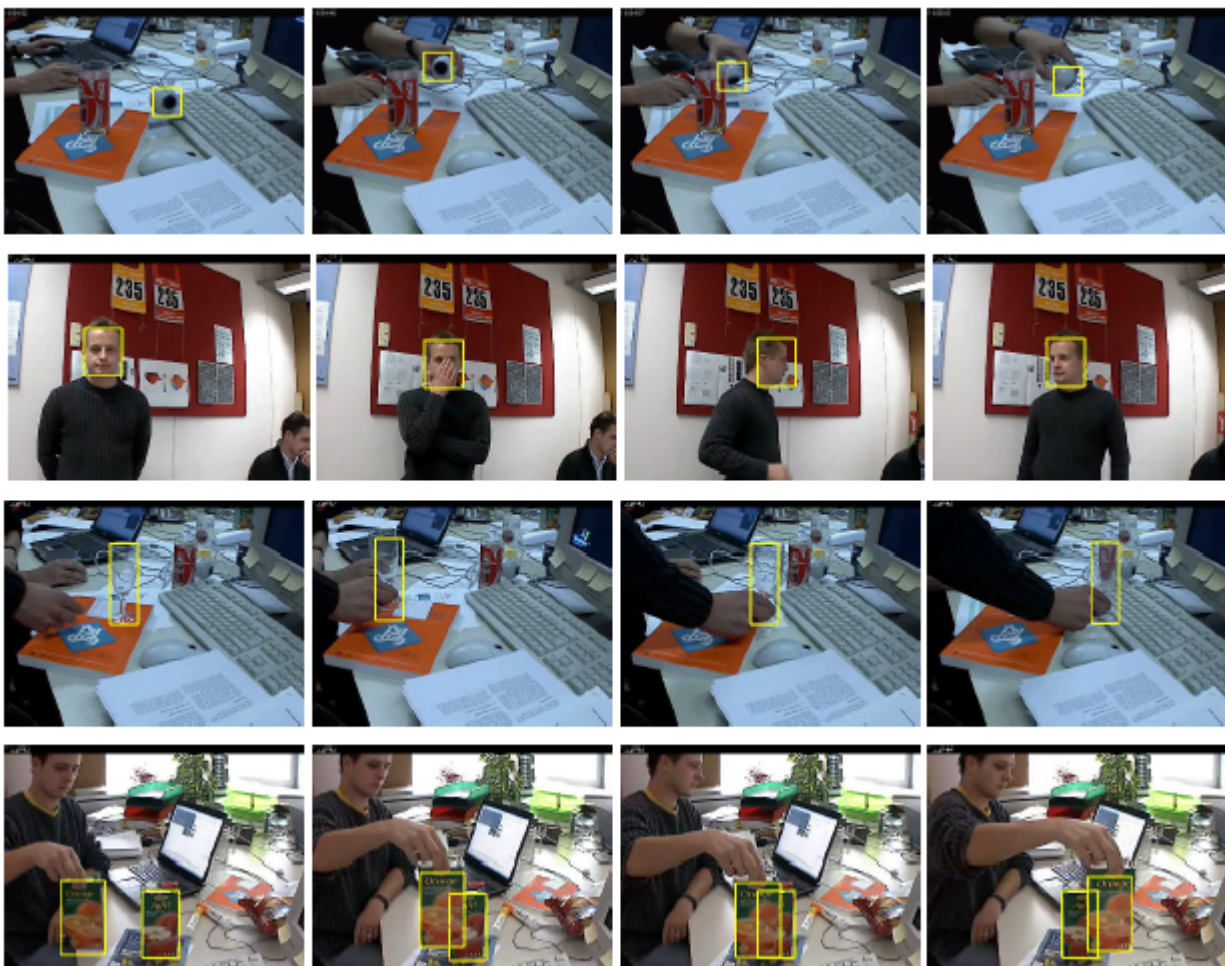


Figure 3.8 – Pour montrer la généralité de la méthode proposée, des séquences de quatre objets différents ont été capturées. Les résultats du suivi montrent que même les objets sans texture (voir la ligne 3) peuvent être suivis avec succès. De plus, l'algorithme de suivi peut gérer plusieurs objets initialisés même s'ils ont une apparence similaire (voir la ligne 4) [7].

### 3.3.2 Détection du sujet

#### 3.3.2.1 Soustraction d'arrière-plan

La segmentation en temps réel des régions en mouvement dans les séquences d'images est la première étape de nombreux systèmes de vision, y compris la surveillance visuelle automatisée, l'interface homme-machine et les télécommunications à très faible bande passante. Une méthode habituelle est

la soustraction d'arrière-plan. De nombreux modèles de base ont été introduits pour traiter différents problèmes.

L'une des solutions efficaces à ces problèmes consiste à utiliser un modèle d'arrière-plan multicolore par pixel proposé par Grimson et al. [84, 85, 86]. Néanmoins, la méthode génère un apprentissage lent au début, en particulier dans les environnements occupés. Un schéma de détection d'ombre est utilisé ; celui-ci repose sur un espace de couleur connu par ordinateur utilisant le modèle d'arrière-plan [87]. Dans ce projet, nous avons nommé cet algorithme MOG. Le résultat de cet algorithme est montré à la figure 3.9.



Figure 3.9 – Utilisation l'algorithme MOG pour supprimer l'arrière-plan.

### 3.3.2.2 Mélange de distributions gaussiennes $K$

$K$  est un nombre compris entre 3 et 5. Grimson et Stauffer [85, 86] introduisent une méthode pour modéliser chaque pixel d'arrière-plan. Les différentes gaussiennes représentent des couleurs différentes. Les paramètres de poids de mélange décrivent les proportions de temps pendant lesquelles ces couleurs restent dans la scène. Contrairement à ce que montrent Friedman et al. [85], les composants d'arrière-plan sont déterminés en supposant que l'arrière-plan contient  $B$  couleurs probables et plus élevées. Les couleurs d'arrière-plan probables sont celles qui restent plus longues et plus statiques. Les objets statiques à une seule couleur visent à former des grappes serrées dans l'espace de couleur quand ils se déplacent sous forme de grappes élargies en raison des différentes surfaces réfléchissantes à travers le mouvement. La mesure de ceci a été appelée *fitness* dans leurs articles. Pour adapter les modifications du modèle en illumination et les exécuter en temps réel, ils ont appliqué un schéma de mise à jour basé sur une mise à jour sélective. Dans l'ordre de *fitness*, chaque nouvelle valeur de

pixel est vérifiée par rapport aux composants de modèle existants. Pour ce faire, ils mettront à jour le premier composant du modèle correspondant. S'il n'y a pas de correspondance, une autre composante gaussienne sera ajoutée avec la moyenne à ce point, une grande matrice de covariance et une petite valeur de paramètre de pondération.

### 3.3.2.3 Modèle de mélange gaussien adaptatif

Un mélange de  $K$  distributions gaussiennes modélise chaque pixel de la scène. La probabilité qu'un certain pixel ait une valeur de  $x_N$  au moment  $N$  peut être décrite comme suit :

$$P(x_N) = \sum_{j=1}^K w_j \eta(x_N; \theta_j) \quad (3.6)$$

Où  $w_k$  est le paramètre de poids de la  $K^e$  composante gaussienne et  $\eta(x; \theta_K)$  est la distribution normale de la  $K^e$  composante décrite par :

$$\eta(x; \theta_K) = \eta(x; \mu_K; \Sigma_K) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_K|^{\frac{1}{2}}} e^{-\frac{1}{2} (x - \mu_K)^T \Sigma_K^{-1} (x - \mu_K)} \quad (3.7)$$

Où  $\mu_K$  est la moyenne et  $\Sigma_K = \sigma_K^2 I$  est la covariance pour les distributions normales multivariées de la  $K^e$  composante. Les  $K$  distributions sont ordonnées en fonction de la valeur du fitness  $\frac{w_K}{\sigma_K}$  et les  $B$  premières distributions sont utilisées comme modèle de l'arrière-plan de la scène où  $B$  est estimé comme suit :

$$B = \underset{b}{\operatorname{argmin}} (\sum_{j=1}^b w_j > T) \quad (3.8)$$

Où  $T$  est le seuil qui est la fraction minimale du modèle d'arrière-plan. C'est la probabilité minimale préalable que le fond est dans la scène. La soustraction d'arrière-plan est présentée en considérant un pixel de premier plan (chaque pixel dépassant de plus de 2,5 écarts-types par rapport aux  $B$  distributions). Les équations de mise à jour suivantes mettent à jour le premier composant gaussien correspondant à la valeur de test [85] :

$$\begin{aligned} \hat{W}_K^{N+1} &= (1 - \alpha) \hat{W}_K^N + \alpha \hat{P}(\omega_K | x_{N+1}) \\ \hat{\mu}_K^{N+1} &= (1 - \alpha) \hat{\mu}_K^N + \rho x_{N+1} \\ \hat{\Sigma}_K^{N+1} &= (1 - \alpha) \hat{\Sigma}_K^N + \rho (x_{N+1} - \hat{\mu}_K^{N+1})(x_{N+1} - \hat{\mu}_K^{N+1})^T \\ \rho &= \alpha \eta(x_{N+1}; \hat{\mu}_K^N, \hat{\Sigma}_K^N) \\ \hat{P}(\omega_K | x_{N+1}) &= \begin{cases} 1 & \text{si } \omega_K \text{ est le premier composant gaussien} \\ 0, & \text{autrement} \end{cases} \end{aligned} \quad (3.9)$$



### 3.3.2.4 Algorithmes EM en ligne

Nous commençons notre estimation du modèle de mélange gaussien par des équations de mise à jour des statistiques suffisantes attendues, puis nous passons à la version de fenêtre  $L$ -récente lorsque les premiers  $L$  échantillons sont préparés. Les équations de mise à jour des statistiques suffisantes attendues fournissent d'abord une bonne estimation avant que tous les  $L$  échantillons puissent être obtenus. Cette estimation initiale augmente la précision de l'estimation et la performance du traqueur, permettant ainsi une convergence rapide sur un modèle d'arrière-plan stable. Les équations de mise à jour de la fenêtre  $L$ -récent donnent la priorité aux données récentes. Le traqueur peut donc s'adapter aux changements de l'environnement.

Les algorithmes EM en ligne avec les statistiques suffisantes attendues sont affichés dans l'équation.3.10, tandis que la version de fenêtre  $L$ -récente est affichée dans l'équation.3.11 [85].

$$\begin{aligned}\hat{W}_k^{N+1} &= \hat{W}_k^N + \frac{1}{N+1} (\hat{P}(\omega_k | x_{N+1}) - \hat{W}_k^N) \\ \hat{\mu}_k^{N+1} &= \hat{\mu}_k^N \frac{\hat{P}(\omega_k | x_{N+1})}{\sum_{i=1}^{N+1} \hat{P}(\omega_k | x_i)} (x_{N+1} - \hat{\mu}_k^N) \\ \hat{\Sigma}_k^{N+1} &= \hat{\Sigma}_k^N + \frac{\hat{P}(\omega_k | x_{N+1})}{\sum_{i=1}^{N+1} \hat{P}(\omega_k | x_i)} ((x_{N+1} - \hat{\mu}_k^N)(x_{N+1} - \hat{\mu}_k^N)^T - \hat{\Sigma}_k^N)\end{aligned}\tag{3.10}$$

$$\begin{aligned}\hat{W}_k^{N+1} &= \hat{W}_k^N + \frac{1}{L} (\hat{P}(\omega_k | x_{N+1}) - \hat{W}_k^N) \\ \hat{\mu}_k^{N+1} &= \hat{\mu}_k^N + \frac{1}{L} \left( \frac{\hat{P}(\omega_k | x_{N+1}) x_{N+1}}{\hat{W}_k^{N+1}} - \hat{\mu}_k^N \right) \\ \hat{\Sigma}_k^{N+1} &= \hat{\Sigma}_k^N + \frac{1}{L} \frac{\hat{P}(\omega_k | x_{N+1}) (x_{N+1} - \hat{\mu}_k^N)(x_{N+1} - \hat{\mu}_k^N)^T}{\hat{W}_k^{N+1}} - \hat{\Sigma}_k^N\end{aligned}\tag{3.11}$$

### 3.3.2.5 Détection des ombres et modèle de couleur

le traqueur de Grimson et al. ne peut pas identifier les ombres en mouvement des objets les projetant. La raison en est qu'il n'existe aucune heuristique pour qualifier les composants gaussiens d'ombres mouvantes. Une solution consiste à utiliser une représentation chromatique de l'espace colorimétrique qui diminue la sensibilité. Comme de nombreux espaces colorimétriques peuvent séparer les composants chromatiques et d'éclairage, le maintien du modèle achromatique, quelle que soit la luminosité, peut créer un modèle instable, en particulier pour les objets très lumineux et les objets lumineux. Cette conversion nécessite également des ressources informatiques, en particulier pour les grandes images. L'idée de préserver les composantes d'intensité et d'économiser des coûts de calcul nous ramène à l'espace RGB. Pour pouvoir identifier les ombres en mouvement, nous devons envisager un modèle de couleur capable de séparer les composants chromatiques et les composants de luminosité.

Cela est effectué en comparant un pixel non-arrière-plan avec les composants d'arrière-plan actuels. Si la différence entre les composantes de luminosité et les couleurs est comprise dans certains seuils, le pixel sera considéré comme une ombre. KaewTraKulPong et al. [88] utilisent un modèle de couleur computationnel utile qui s'avère similaire à celui proposé par Horprasert et al. [89] pour satisfaire à ces exigences. Il consiste en un vecteur de position à la moyenne RGB du fond de pixels,  $E$ , une ligne de chromaticité attendue,  $\|E\|$ , une distorsion chromatique,  $d$ , et un seuil de luminosité,  $\tau$ . Pour une valeur de pixel reconnue,  $I$ , une distorsion de luminosité,  $a$ , et une distorsion de couleur,  $c$ , à partir du modèle d'arrière-plan peuvent être calculées comme suit :

$$\begin{aligned} a &= \underset{z}{\operatorname{argmin}}(I - zE)^2 \\ c &= \|I - aE\| \end{aligned} \tag{3.12}$$

Avec l'hypothèse de la distribution gaussienne sphérique dans chaque composante du mélange, l'écart type de la  $K^e$  composante  $\sigma_K$  peut être égal à  $d$ . en utilisant le produit scalaire, le calcul de  $a$  et  $c$  est trivial. Un échantillon non observé en arrière-plan est considéré comme une ombre en mouvement si  $a$  est, dans notre cas, 2,5 écarts-types et  $\tau < c < 1$ .

### 3.3.2.6 Algorithme de Canny

Dans ce projet, on a utilisé les algorithmes de soustraction MOG pour supprimer l'arrière-plan, puis on a utilisé un algorithme « Canny » [90] pour trouver les bords de la souris. En sachant les bords, on a trouvé le cadre de sélection de la souris pour déterminer le comportement de la souris.

Pour trouver les bords de la souris, un algorithme de détection des contours développé par John F. Canny sera utilisé. L'algorithme de détection de bord Canny est composé de 5 étapes :

1. Réduction de bruit.
2. Calcul du gradient.
3. Suppression non maximale.
4. Double seuil.
5. Suivi de bord par hystérésis.

Dans cet algorithme, la première étape consiste à supprimer le bruit dans l'image avec un filtre gaussien, car la détection des contours est sensible au bruit dans l'image. Ensuite, l'image est filtrée avec un noyau Sobel dans le sens vertical et horizontal pour obtenir la première dérivée dans la direction verticale ( $G_y$ ) et la direction horizontale ( $G_x$ ).

L'opérateur Sobel est un opérateur de différenciation discret. Il calcule une approximation du gradient d'une fonction d'intensité d'image. L'opérateur Sobel combine le lissage et la différenciation gaussiens.

À partir de ces deux images, le gradient et la direction des bords de chaque pixel seront trouvés comme suit :

$$\begin{aligned} Edge\_Gradient(G) &= \sqrt{(G_x^2 + G_y^2)} \\ Angle(\theta) &= \tan^{-1}\left(\frac{G_y}{G_x}\right) \end{aligned} \tag{3.13}$$

Un balayage complet de l'image est effectué pour supprimer les pixels indésirables après avoir obtenu la magnitude et la direction du dégradé, ce qui peut ne pas constituer le bord. La dernière étape consiste à décider quels sont tous les bords réels et ceux qui ne le sont pas. Deux valeurs de seuil sont considérées, maxVal et minVal. Toutes les arêtes dont le gradient d'intensité est supérieur à maxVal sont sûres d'être des arêtes et celles situées en dessous de minVal ne le sont pas, donc écartées. En outre, ceux qui se situent entre deux seuils sont des bords classés ou non en fonction de leur connectivité. S'ils sont liés à des pixels de « bord sûr », ils sont considérés comme faisant partie des bords. Sinon, ils sont rejetés [90].

### 3.3.3 Trouver le comportement de la souris

En utilisant le cadre de sélection de la souris à chaque image, nous estimons son comportement. Puisque nous avons placé la caméra au-dessus de la cage, nous pouvons estimer le rapport entre l'axe le plus long et l'axe le plus petit. Lorsque la souris marche, ce rapport est supérieur à 1. Lorsque la souris est debout, il devrait diminuer car la forme de l'image commence à ressembler à un cercle et lorsque la souris se repose, sa position sera la même dans certains frames.

## Chapitre 4

# Résultats expérimentaux

### 4.1 Installation

Nous avons installé le système comprenant une caméra, une base réglable, une cage, un sujet de suivi et un ordinateur portable dans un laboratoire comme la figure 4.1 et avons par la suite de fait des expériences. Nous avons placé la caméra sur le dessus de la cage pour obtenir une vue complète de celle-ci [32]. Pour ce faire, nous avons eu recours à une webcam Logitech C170. Cette webcam légère et facile à utiliser peut fournir une image vidéo de haute qualité. Les caractéristiques techniques de la webcam Logitech C170 sont affichées dans le tableau 4.1.

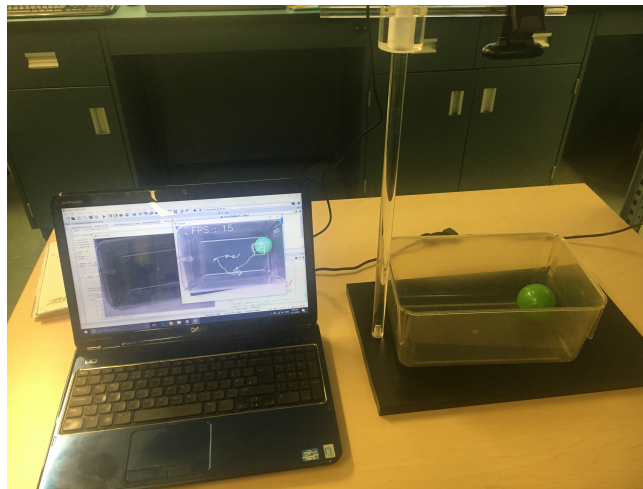


Figure 4.1 – Système de suivi de mouvement.

L'ordinateur qu'on a utilisé dans cette expérience est un laptop Dell Inspiron N5110. Ses caractéristiques techniques sont affichées dans le tableau 4.2.

Nous avons utilisé Python 3.6 et OpenCV sur Spyder 3.2.8 pour la programmation dans ce projet.

Notre système est un système en ligne ; cela signifie que nous capturons une vidéo (qui n'est qu'une

Webcam Logitech C170	
Brand	Logitech
Model	C170
Connectivity	USB 2.0
Video Capture	HD 720p Up to 1280 x 720, 640x480
Photo	Up to 5 megapixel
Image Sensor Type	Cmos
Focal Length	2.3mm
Frame Rate(max)	30

TABLE 4.1 – Les caractéristiques techniques de la webcam Logitech C170.

Laptop Dell Inspiron N5110	
Brand	Dell
Model	15R N5110
Operating System Type	64-bit
Processor	Intel Core i5-2450M (2nd Gen)
Clock-speed	2.5 Ghz
Chipset	Intel HM67 Express
Graphic Processor	NVIDIA GeForce
RAM Capacity	4 GB
RAM type	DDR3
HDD Capacity	500 GB
HDD type	SATA

TABLE 4.2 – Les caractéristiques techniques laptop Dell Inspiron 15R N5110.

séquence d'images), procédons image par image et affichons les résultats. Mais il est facile de modifier les paramètres et de l'utiliser en tant que système hors ligne, ce qui signifie que nous pouvons envoyer un fichier vidéo comme paramètre de fonction, procédons image par image et affichons les résultats. Les deux cas (en ligne / hors ligne) sont vérifiés.

## 4.2 Calibration de la caméra

La calibration de la caméra est le processus d'estimation des paramètres intrinsèques et extrinsèques.

Les paramètres intrinsèques traitent des caractéristiques internes de l'appareil photo, telles que la distance focale, la distorsion, le centre de l'image et l'inclinaison. Les paramètres extrinsèques décrivent sa position et son orientation dans le monde.

La connaissance des paramètres intrinsèques est une première étape essentielle pour la vision numérique, car elle nous permet d'estimer la structure de la scène dans un espace euclidien et d'éliminer les distorsions de l'objectif, qui dégradent la précision.

Pour calibrer une caméra, nous devons fournir quelques exemples d'images d'un motif bien défini (par exemple, un échiquier). Pour obtenir de meilleurs résultats, nous avons besoin d'au moins 10 modèles de test. La figure 4.2 montre une image d'échiquier que nous avons utilisé pour calibrer la caméra.

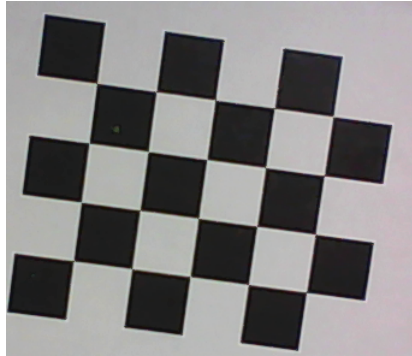


Figure 4.2 – Image d'échiquier utilisé pour calibrer la caméra.

Comme nous l'avons mentionné précédemment, en calibrant une caméra, nous allons atteindre les paramètres intrinsèques. Les paramètres intrinsèques consistent en la distance focale ( $f_x, f_y$ ), le centre optique, appelé point principal ( $c_x, c_y$ ), et le coefficient de biais. Il est exprimé sous forme de matrice 3x3. Nous avons installé notre caméra et exécuté le programme de Python pour calibrer la caméra. Nous corrigeons d'abord ces distorsions et puis obtenons les paramètres intrinsèques. La matrice de la caméra obtenue par la calibration est indiquée ci-dessous.

$$\text{Camera Matrix} = \begin{pmatrix} 1.2827e+4 & 0 & 3.6991e+4 \\ 0 & 2.8005e+3 & 8.1547e+3 \\ 0 & 0 & 1 \end{pmatrix}$$

Comme il est indiqué dans le tableau 4.1, la distance focale de la caméra ( $F$ ) est 2.3 mm. La distance de la caméra au sujet ( $D$ ) est de 450 mm. Si nous obtenons la position du sujet, par exemple (260, 150), en utilisant la règle en forme de triangle similaire, la position réelle dans l'unité mondiale(mm) sera :

$$\begin{aligned} 12827/260 &= (2.3 + 450)/P_x \Rightarrow P_x = 9.16 \text{ mm} \\ 2800.5/150 &= (2.3 + 450)/P_y \Rightarrow P_y = 24.22 \text{ mm} \end{aligned}$$

Donc, la position réelle du sujet est (9.16, 24.22)

L'erreur de reprojection est utilisée dans de nombreux documents de vision numérique. La manière correcte de vérifier la précision de la calibration de la camera consiste à utiliser l'erreur de reprojection fournie par OpenCV. L'erreur de reprojection est l'erreur RMS entre l'endroit où les points seraient

projetés à l'aide des coefficients intrinsèques et leur emplacement dans l'image réelle. En règle générale, une erreur RMS doit être inférieure à 0.5 px. Nous avons obtenu  $RMS = 0.18$  px dans cette expérience.

### 4.3 Suivi du sujet

Comme nous l'avons expliqué au chapitre précédent, nous avons utilisé l'algorithme AdaBoost pour suivre un objet. Le classificateur doit être formé au moment de l'exécution avec des exemples positifs (objet) et négatifs (arrière-plan) de l'objet. Le cadre de sélection initial fourni par l'utilisateur est pris comme exemple positif de l'objet et de nombreux correctifs d'image en dehors du cadre de sélection sont utilisés comme arrière-plan. En ayant un nouveau cadre, le classificateur est appliqué à chaque pixel voisin de l'emplacement précédent, et le score du classificateur est enregistré. Le nouvel emplacement de l'objet est celui où le score est maximal. Par conséquent, il y a maintenant un autre exemple positif de classificateur. Le classificateur est mis à jour avec ces données supplémentaires.

Dans la première image, nous déterminons une image de sélection comprenant l'objet affiché à la figure 4.3

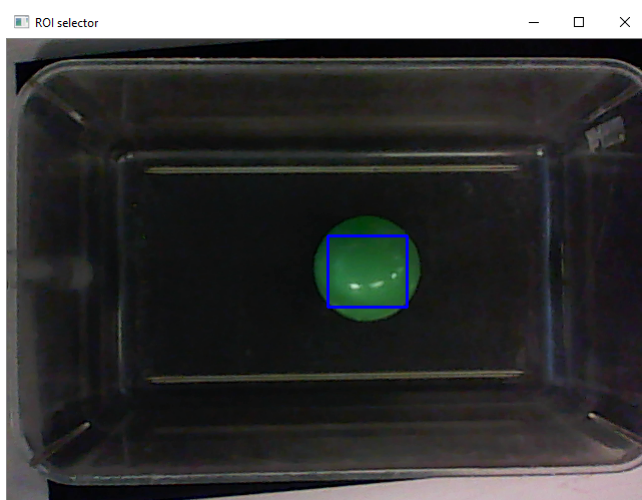


Figure 4.3 – Une image de sélection incluant le sujet dans la première image. La vidéo est en ligne.

Puis nous lisons les images de la vidéo en boucle. L'algorithme de suivi tente de trouver le centre du cadre de sélection du sujet la figure 4.4.

Nous pouvons jouer une vidéo en ligne ou lire une vidéo à partir du fichier. Nous avons vérifié l'algorithme AdaBoost en ligne pour suivre une souris et essayé de suivre la souris dans deux différentes conditions dans une vidéo à partir du fichier.

1- Au début, la position de la caméra est changée pendant le test. Nous voulons juste voir la robustesse d'algorithme, la position réelle de la souris n'est pas importante dans ce test. Cependant, lorsque nous



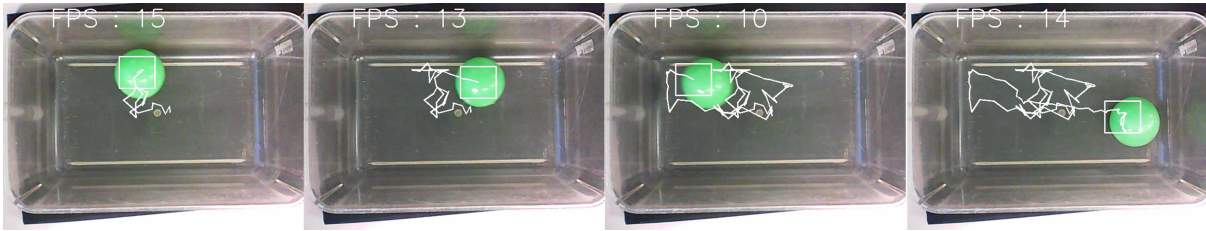


Figure 4.4 – Utiliser l’algorithme AdaBoost en ligne pour suivre un ballon vert dans une cage. La vidéo est en ligne.

changeons la position de la caméra, l’algorithme fonctionne bien et elle trouve correctement le cadre de sélection de l’objet sélectionné. La figure 4.5 montre les résultats du suivi.

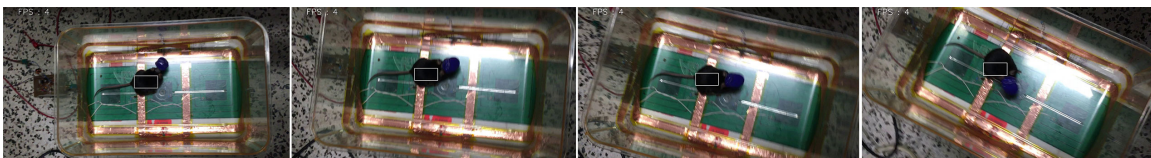


Figure 4.5 – Utiliser l’algorithme AdaBoost en ligne pour suivre une souris noire dans une cage et la position de la caméra est modifiée. La vidéo est hors ligne.

2- Ensuite, nous avons essayé de suivre une souris blanche dans une cage grise [8]. Trouver le cadre de sélection de l’objet dans cette condition est compliqué, la figure 4.6 montre les résultats du suivi. Le taux de réussite de l’algorithme à trouver le sujetL’ est 90.78%.

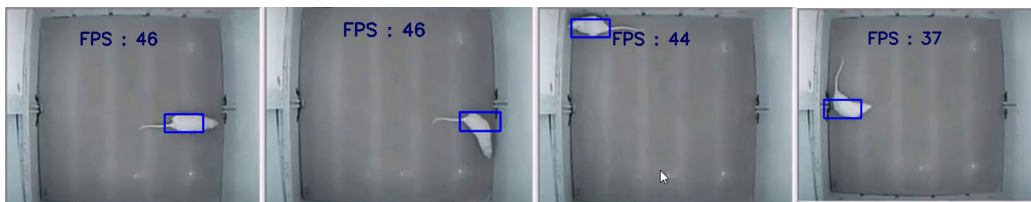


Figure 4.6 – Utiliser l’algorithme AdaBoost en ligne pour suivre une souris blanche dans une cage grise. La vidéo est hors ligne[8].

Le chemin du sujet dans un diagramme en 2D est affiché à la figure 4.7.

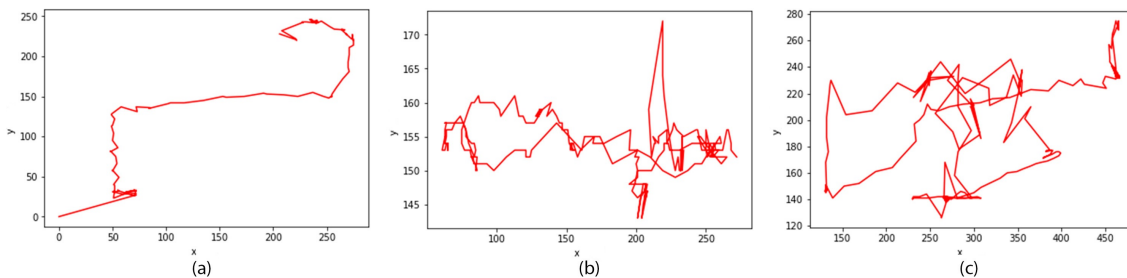


Figure 4.7 – a) Le chemin de la souris blanche dans la cage grise. b) Le chemin de la souris noire. c) Le chemin du ballon vert.



La plupart des approches existantes construisent une représentation du sujet cible avant le début de la tâche de suivi et utilisent donc une représentation fixe pour gérer les modifications d'apparence pendant le suivi. Cependant, l'algorithme AdaBoost en ligne fait les deux, c'est-à-dire s'ajuster aux variations d'apparence lors du suivi et sélectionner les caractéristiques appropriées qui peuvent apprendre tout objet et le différencier de l'arrière-plan environnant. La base est un algorithme AdaBoost en ligne qui permet de mettre à jour les caractéristiques du classificateur pendant le suivi. En outre, le calcul efficace des caractéristiques permet d'utiliser ce suivi dans des applications en temps réel.

Les résultats expérimentaux montrent que l'algorithme AdaBoost en ligne a les caractéristiques de temps réel, de précision et de stabilité.

## 4.4 Détection du sujet

Étant donné que la caméra est fixée sur le dessus de la cage et que le seul objet déplacé est la souris, nous pensons que la meilleure façon consiste à faire soustraction de l'arrière-plan et à trouver le bord de la souris.

La soustraction de fond est une technique courante et largement utilisée pour générer un masque de premier plan (une image binaire contenant les pixels appartenant à des objets en mouvement dans la scène) à l'aide de caméras statiques.

La soustraction d'arrière-plan est une étape de prétraitement majeure dans de nombreuses applications basées sur la vision. Par exemple, considérons le cas d'un compteur de visiteurs où une caméra statique calcule le nombre de visiteurs entrant ou sortant de la salle, ou celui d'une caméra de circulation extrayant des informations sur les véhicules. Dans tous ces cas, nous devons d'abord extraire la personne ou les véhicules seulement. Techniquement, nous devons extraire le premier plan en mouvement de l'arrière-plan statique.

S'il y a une image de l'arrière-plan seul, comme une image de la pièce sans visiteurs ou une image de la route sans véhicules, c'est un travail facile. Il suffit de soustraire la nouvelle image de l'arrière-plan. Il n'y aura ainsi que les objets de premier plan. Toutefois, dans la plupart des cas, il n'y a pas qu'une image, donc il faut extraire l'arrière-plan de toutes les images nécessaires. Cela devient d'autant plus compliqué quand il y a des ombres sur les véhicules. Puisque les ombres bougent aussi, une simple soustraction marquera cela aussi comme avant-plan. Cela complique les choses.

Plusieurs algorithmes permettent de soustraire l'arrière-plan. Les algorithmes présentés dans [88] sont utilisés.

Il s'agit d'un algorithme de segmentation d'arrière-plan / avant-plan basé sur un mélange gaussien. Il utilise une méthode pour modéliser chaque pixel d'arrière-plan par un mélange de  $K$  distributions gaussiennes ( $K = 3$  à  $5$ ). Les poids du mélange représentent les proportions de temps dans lesquelles

ces couleurs restent dans la scène. Les couleurs d'arrière-plan probables sont celles qui demeurent plus longues et plus statiques.

La figure 4.8 montre les résultats des tentatives visant à soustraire l'arrière-plan dans la vidéo montrant le ballon vert.

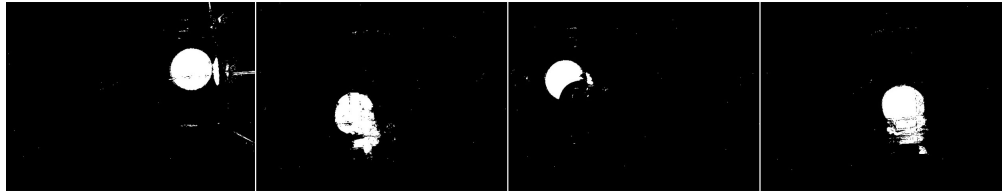


Figure 4.8 – Le ballon vert après la soustraction de l'arrière-plan avec l'algorithme MOG. La vidéo est en ligne.

Nous avons répété la soustraction de l'arrière-plan dans deux différentes conditions de vue comme suit.

1- Nous avons d'abord essayé de soustraire l'arrière-plan pendant que la caméra était placée sur le dessus de la cage. Les résultats sont illustrés à la figure 4.9.



Figure 4.9 – La souris blanche dans la cage grise après la soustraction d'arrière-plan avec l'algorithme MOG. La caméra est placée sur le dessus de la cage. La vidéo est hors ligne [8].

2- Ensuite, nous avons essayé de soustraire l'arrière-plan pendant que la caméra était placée sur le côté de la cage. Les résultats sont montrés à la figure 4.10.

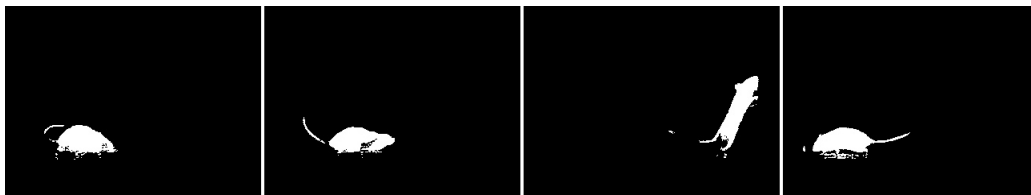


Figure 4.10 – La souris noire après la soustraction d'arrière-plan avec l'algorithme MOG. La caméra est placée sur le côté de la cage. La vidéo est hors ligne [9].

Ensuite, nous avons trouvé le bord de la souris par l'algorithme Canny et dessiné le cadre de sélection de la souris.

L'algorithme de détection des contours développé par John F. Canny sera utilisé pour trouver les bords

de la souris. Nous avons constaté que lorsque l'algorithme de soustraction d'arrière-plan fonctionne bien, l'algorithme de Canny s'avère aussi efficace pour trouver les contours de la souris.

La détection Canny Edge est une technique permettant d'extraire des informations structurelles utiles de différents objets de vision et de réduire considérablement la quantité de données à traiter. Elle a été largement appliquée dans divers systèmes de vision numérique. Canny a constaté que les exigences pour l'application de la détection de contour sur divers systèmes de vision sont relativement similaires. Ainsi, une solution de détection de bord pour répondre à ces exigences peut être mise en œuvre dans un large éventail de situations. Les critères généraux de détection des contours sont les suivants :

- 1- Détection des contours avec un taux d'erreur faible, ce qui signifie que la détection doit capturer avec précision autant de contours que possible dans l'image
- 2- Le point de bord détecté par l'opérateur doit localiser précisément au centre du bord.
- 3- Un bord donné de l'image ne doit être marqué qu'une seule fois et le bruit de l'image ne doit pas créer de faux bords.

Les contours de la souris sont affichés à la figure 4.11.

En disposant des contours de la souris, nous avons tenté de trouver le cadre de sélection de la souris.

Nous commençons à trouver le sujet avec deux algorithmes AdaBoost en ligne et Canny.

Si l'algorithme AdaBoost en ligne peut trouver le sujet, nous gardons cette position pour dessiner le chemin du suivi. Si deux algorithmes Adaboost en ligne et Canny trouvent le même sujet, nous estimons le comportement de la souris et continuons la processus.

Si l'algorithme AdaBoost en ligne ne peut pas trouver le sujet, nous gardons la position obtenue de l'algorithme Canny pour dessiner le chemin du suivi, estimons le comportement de la souris et continuons la processus.

Si ces deux algorithmes (Adaboost en ligne et Canny) ne trouvent pas le même sujet, nous sautons d'estimer le comportement de la souris et continuons la processus.

Les résultats sont affichés à la figure 4.12.

En utilisant le cadre de sélection de la souris à chaque image, nous estimons son comportement. En comparant le rapport entre l'axe le plus long et l'axe le plus petit, nous avons estimé trois comportements de souris différents (redressement, marché et se reposer) avec cette approche. Les rectangles encadrants ne sont détectés que pour des directions parallèles aux axes de l'image qui ils sont suffisant pour trouver les trois comportements (redressement, marché et se reposer).

Le comportement de la souris sont affichés à la figure 4.13.

Nous pouvons jouer une vidéo en ligne ou lire une vidéo à partir du fichier. Si nous utilisons une vidéo

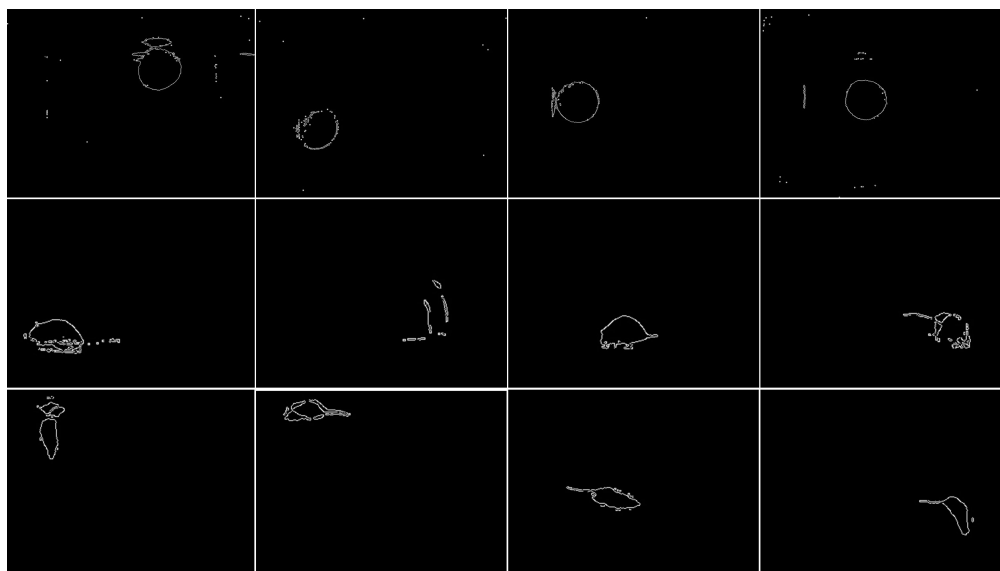


Figure 4.11 – Les bords des sujets avec l’algorithme Canny. La première rangée montre les bords du ballon vert, la vidéo est en ligne. La deuxième rangée montre les bords de la souris noire, la vidéo est hors ligne [9]. La troisième rangée montre les bords de la souris blanche dans la cage grise, la vidéo est hors ligne [8].

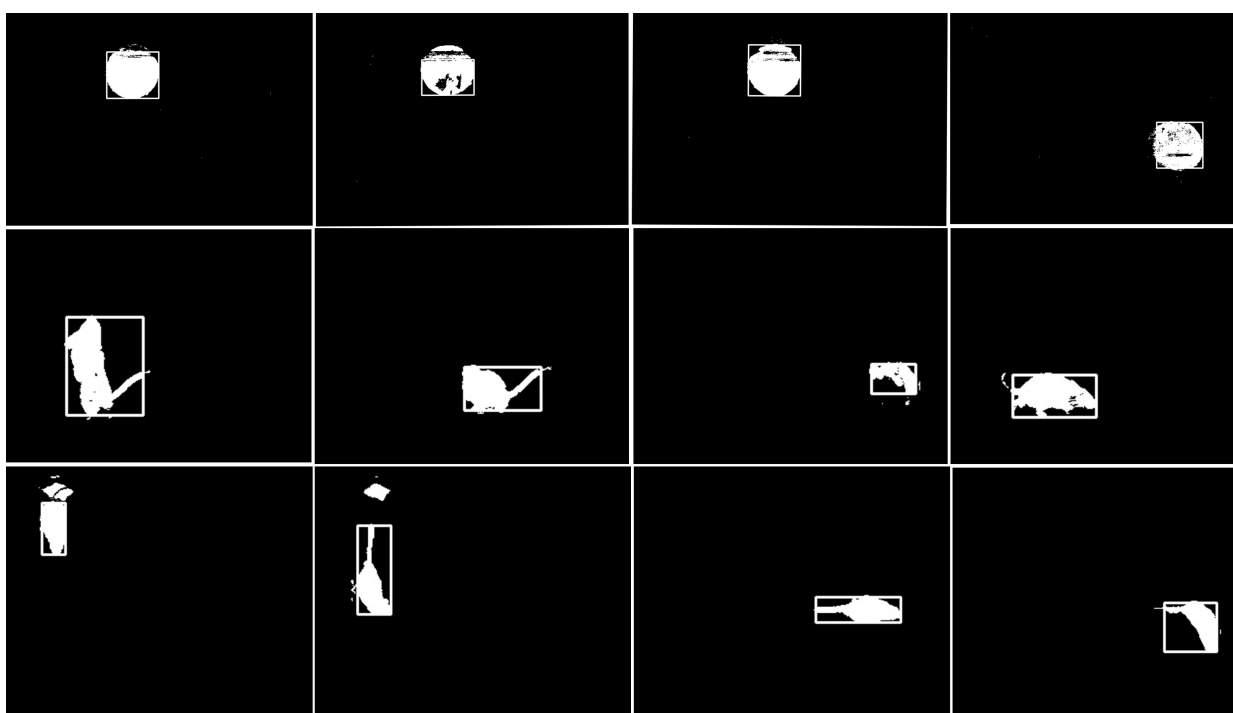


Figure 4.12 – Les cadres de sélection des sujets en appliquant les algorithmes MOG and Canny. La première rangée montre le cadre de sélection du ballon vert, la vidéo est en ligne. La deuxième rangée montre le cadre de sélection de la souris noire, la vidéo est hors ligne [9]. La troisième rangée montre le cadre de sélection de la souris blanche dans la cage grise, la vidéo est hors ligne [8].

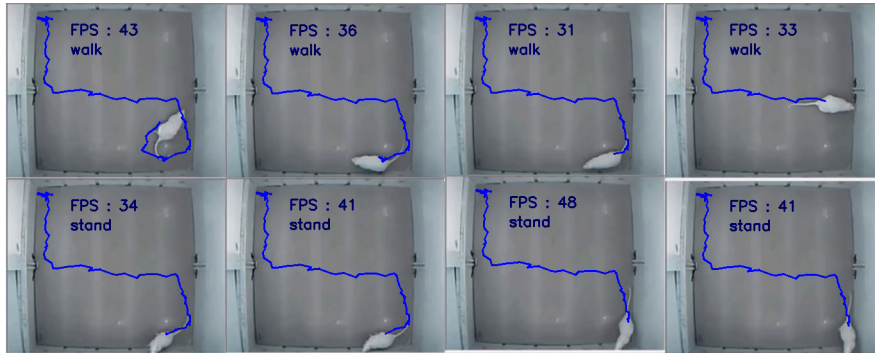


Figure 4.13 – Le comportement de la souris et le chemin du souris en exécutant la solution proposée. La vidéo est hors ligne [8]

en ligne, celle-ci sera enregistrée à la fin du processus de suivi.

La vidéo du résultat de notre projet est disponible sur YouTube [91].

Nous avons présenté une méthode précise qui est simple, peu dispendieuse et ne nécessite pas de matériel complexe, de cage, de caméra ou tout matériel spécial. Il n’y a pas de dépendance au type de sujet pour détecter.

Suivi d’un objet détecté dans la trame précédente, et les informations sur l’apparence de l’objet seront connues. L’emplacement dans le cadre précédent, la direction et la vitesse de son mouvement seront également connus. Ainsi, dans le cadre suivant, l’outil de suivi utilise toutes ces informations pour prédire l’emplacement de l’objet dans l’image suivante et effectue une petite recherche autour de l’emplacement attendu de l’objet pour localiser l’objet avec précision.

## 4.5 Évaluation des performances

Pour évaluer notre système, nous avons utilisé trois vidéos différentes, les vidéos qui sont hors ligne [8, 9] et en ligne. Nous avons tenté de trouver l’objet dans chaque vidéo, image par image. Nous avons considéré 450 images par vidéo, car nous pensons que la plupart des mouvements d’objet seront observés dans les 450 premières images. Le suivi fonctionne sur de plus longues périodes de temps.

L’arrière-plan utilisé dans les expériences est réaliste, on a utilisé la vidéo de notre expérience de suivi la souris au centre de recherche CERVO de ville de Québec et les vidéos pour étudier au le comportement des souris [9].

Des paramètres pour l’algorithme sont une vidéo, la vue de la caméra (vue de dessus ou vue latérale), bord (carte de bord de sortie) et seuil1 (premier seuil pour la procédure d’hystérésis). Nous utilisons la vue latérale lorsqu’il est difficile de voir la cage en vue de dessus. On utilise  $bord = 30$  et  $seuil1 = 200$ .

Comme nous l’avons expliqué précédemment, pour trouver un objet dans une image, nous avons

utilisé deux algorithmes différents ensemble : Adaboost en ligne (un algorithme de suivi d'objet), et Canny (un algorithme de détection d'objet).

Pour mieux comprendre les performances de notre système, 1- nous avons essayé de trouver les objets dans les 450 premières images des trois mêmes vidéos en utilisant uniquement l'algorithme Adaboost en ligne.

Le cadre de délimitation initial fourni par l'utilisateur est pris comme exemple positif de l'objet, et de nombreux correctifs d'image en dehors du cadre de sélection sont utilisés comme arrière-plan. En ayant une nouvelle trame, le classificateur est appliqué à chaque pixel voisin de l'emplacement précédent et le score du classificateur est enregistré. Le nouvel emplacement est celui où le score est le plus grand. Ainsi, maintenant, il y a un autre exemple positif du classificateur. Par conséquent la sensibilité du système implémenté est aussi élevée qu'un pixel.

Comme il a été expliqué précédemment, nous avons vérifié 450 images dans trois cas différents. Dans le tableau 4.3, on montre le taux de réussite d'AdaBoost en ligne à trouver le sujet.

2- Nous avons ensuite essayé de trouver les mêmes objets dans les 450 premières images pour chaque vidéo avec l'algorithme Canny. Dans le tableau 4.3, on présente le taux de réussite de la détection (Canny) à trouver le sujet dans trois cas différents. Pour chaque cas, les mêmes images ont été vérifiées.

Les algorithmes à gradient ont l'inconvénient majeur d'être très sensibles au bruit. La taille du filtre de noyau et les coefficients sont fixes et ne peuvent pas être adaptés à une image donnée. Un algorithme de détection de bord adaptatif est nécessaire pour fournir une solution robuste adaptable aux niveaux de bruit variables de ces images afin de permettre de distinguer les images valides des artefacts visuels introduits par le bruit.

Les performances de l'algorithme de Canny dépendent en grande partie d'un paramètre ajustable,  $\sigma$ , qui est l'écart type du filtre gaussien. Une valeur plus grande pour  $\sigma$  mène à une taille plus grande du filtre gaussien. Des valeurs plus petites de  $\sigma$  impliquent un filtre gaussien plus petit qui limite le flou, tout en maintenant des bords plus fins dans l'image. L'utilisateur peut personnaliser l'algorithme en ajustant ce paramètre pour s'adapter à différents environnements.

3- Enfin nous avons essayé de trouver les mêmes objets dans les 450 premières images pour chaque vidéo avec notre solution proposée.

Comme nous l'avons expliqué dans la section de présentation de la solution proposée, nous avons implémenté les algorithmes de suivi du sujet et de détection du sujet ensemble, car le suivi peut aider en cas d'échec de la détection et vice et versa. Nous avons vérifié les mêmes images que nous avons utilisées précédemment et comparé les résultats de notre approche avec les résultats de algorithmes Adaboost en ligne et de l'algorithme Canny.

Le tableau 4.3 présente le taux de réussite de notre approche à trouver le sujet, et ce, dans trois cas

	AdaBoost en ligne	Canny	la solution proposée
Suivre une souris noire	94.44%	86.88%	97.33%
Suivre une souris blanche dans une cage grise	90.78%	87.71%	96.49%
Suivre un ballon vert	84.3%	80.31%	89.73%

TABLE 4.3 – Taux de réussite de notre approche à trouver le sujet, et ce, dans trois cas différents, en comparant les algorithmes AdaBoost en ligne et Canny.

différents, en comparant les algorithmes AdaBoost en ligne et Canny. Comme nous l’avons mentionné précédemment, nous avons utilisé deux algorithmes différents pour trouver un objet. Lorsque l’algorithme AdaBoost ne parvient pas à trouver l’objet, nous supprimons l’arrière-plan et utilisons l’algorithme Canny pour trouver le bord de l’objet. Selon les résultats présentés, notre approche fonctionne mieux si nous utilisons ces deux algorithmes séparément.

Selon nos expériences, l’algorithme AdaBoost peut échouer lorsque la couleur de l’objet est proche de la couleur d’arrière-plan. C’est pourquoi nous avons décidé d’utiliser un autre algorithme de détection qui ne dépend pas de la couleur de l’objet pour augmenter la précision de notre système de suivi.

En comparant les résultats de la solution présentée avec des algorithmes de suivi ou de détection pour trouver un sujet dans une image, nous avons constaté que notre approche fonctionne mieux qu’en essayant de trouver l’objet avec des algorithmes de suivi ou de détection séparément.

Pour trouver les comportements de la souris, on utilis la vidéo avec la souris blanche dans la cage grise. Le taux de réussite d’estimation du comportement de la souris est 82.7%.

#### 4.5.1 Erreur quadratique moyenne

L’erreur quadratique moyenne (*MSE*) est utilisée comme métrique pour déterminer les performances d’un algorithme. La formule pour calculer le MSE est la suivante :

$$MSE = 1/n(\sum_{i=1}^n (y_i - y_i^-)^2) \quad (4.1)$$

$n$  est le nombre total de termes pour dont l’erreur doit être calculée.

$y_i$  est la valeur observée de la variable.

$y_i^-$  la valeur prédite de la variable

L’erreur quadratique moyenne est la moyenne du carré de la différence entre les valeurs observées et prédites d’une variable.

Notre système obtient environ  $MSE = 0.03$  d’erreur sur le suivi de la souris et  $MSE = 0.17$  sur l’estimation de son comportement, lorsque la caméra est placée sur le dessus de la cage, et à la vue latérale

$MSE = 0.02$  d'erreur sur le suivi de la souris et  $MSE = 0.12$  sur l'estimation de son comportement.

## 4.5.2 Le taux de trame

En informatique, un système numérique en temps réel est un système qui garantit une réponse avant une durée limite préalablement fixée. Pour un flux d'images, cela signifie qu'un système en temps réel est un système garantissant qu'il a fini de traiter chaque image avant qu'un délai défini. Un système numérique peut être "en temps réel" si il peut traiter des images à la vitesse à laquelle elles sont acquises.

Le traitement des images en temps réel est lié à la fréquence d'images typique. La norme actuelle de capture est généralement de 30 images par seconde. Le traitement en temps réel nécessiterait de traiter toutes les images dès leur capture.

En moyenne, l'œil humain peut traiter 12 images distinctes par seconde. Cela signifie qu'un taux de trame<sup>1</sup> de 12 images par seconde peut afficher le mouvement, mais semblera instable. Le taux de trame de 24 images par seconde est couramment utilisée pour les films car il crée une apparence lisse. De nombreuses caméras vidéo enregistrent à 30 ou 60 images par seconde, ce qui permet un mouvement encore plus fluide.

Dans cette expérience, nous avons calculé le taux de trame. Lorsque nous avons utilisé la webcam Logitech C170 et la laptop Dell Inspiron 15R N5110, le taux de trame de capture obtenu par notre programmation de Python est 20 images par seconde et en moyenne, le taux de trame obtenu pour le traitement des images dans notre système est 14.

10 taux de trame est souvent suffisant pour suivre des objets tels que des personnes ou des animaux. Cela signifie que le taux de trame de notre système est suffisant et on peut dire que notre système est en temps réel.

## 4.5.3 Analyse de la complexité des algorithmes

### 4.5.3.1 Complexité d'algorithme Boosting en ligne

Pour la partie apprentissage, les algorithmes classiques nécessitent d'évaluer la matrice de noyau  $K$ , la matrice dont le terme général est  $K(x_i, x_j)$  où  $K$  est le noyau spécifié. On suppose que  $K$  peut être évalué avec une complexité  $O(p)$ , comme c'est le cas pour les noyaux communs (gaussien, polynômes, sigmoïde...). Alors, résoudre le programme quadratique sous contrainte est « moralement équivalent à » inverser une matrice carrée de taille  $n$ , dont la complexité est supposée être  $O(n^3)$ .

---

1. On utilise le taux de trame pour mesurer les performances du traitement des images vidéo. Le taux de trame est utilisée pour mesurer le nombre d'images consécutives en plein écran qui sont affichées chaque seconde.



#### 4.5.3.2 Complexité d'algorithme Canny

La détection de bord canny consiste en :

- 1- Une convolution de l'image avec un noyau brouillé
- 2- Quatre convolutions de l'image avec des noyaux de détection de bord
- 3- Calcul de la direction du gradient
- 4- Suppression non maximale
- 5- Seuil avec hystérésis

Les étapes (1), (2), (3) et (4) sont toutes réalisées en matière de convolutions de l'image avec des noyaux de taille fixe. Il est possible d'implémenter des convolutions dans le temps  $O(n \log n)$ , où  $n$  est le nombre d'éléments. Si l'image a des dimensions  $m \times n$ , la complexité temporelle sera  $O(mn \log mn)$  pour ces étapes. La dernière étape consiste à post-traiter l'image pour supprimer toutes les valeurs hautes et basses, puis à supprimer tous les autres pixels qui ne se trouvent pas à proximité d'autres pixels. Cela peut être fait dans le temps  $O(mn)$ . Par conséquent, la complexité en temps globale est  $O(mn \log mn)$ .

#### 4.5.3.3 Complexité d'algorithme proposée

Puisque nous avons utilisé ces deux algorithmes ensemble, nous devons additionner les deux complexités, comme  $O(mn \log mn) + O(n^3)$ . Bien que  $O(n^3)$  soit beaucoup plus  $O(mn \log mn)$ , la complexité de notre solution est  $O(n^3)$ , ce qui est identique à la complexité de l'algorithme Adaboost.

# Conclusion

Le comportement des animaux est important pour promouvoir la recherche en pharmacologie et en neurosciences. Les rongeurs sont utilisés comme modèles pour les maladies humaines, donc étudier le comportement des rongeurs est utile pour identifier et développer de nouveaux médicaments pour les troubles psychiatriques et neurologiques.

Dans le chapitre 1, nous avons passé en revue l'application de surveillance cérébrale et avons expliqué que l'optogénétique était précieuse dans cette application et les raisons pour lesquelles il fallait étudier un animal en mouvement libre.

Dans le chapitre 2, les approches récemment mises en œuvre sur le comportement de reconnaissance des rongeurs et les méthodes d'étalonnage des caméras ont été passées en revue. Pour connaître la base neuronale du comportement, il est nécessaire d'enregistrer le mouvement libre des animaux. De plus, les systèmes de suivi commerciaux ont été présentés. Nous avons introduit les systèmes IntelliCage, EthoVision XT, CleverSys et CMUcam. Les systèmes IntelliCage, EthoVision XT, CleverSys ont besoin de leurs cages individuelles et leur propre matériel, ce qui les rend coûteux. De plus, étant donné que nous utilisons notre cage, ces systèmes ne sont pas utiles dans le cadre de ce projet. L'utilisation de CMUcam n'est pas fiable non plus pour les besoins de notre projet parce que la méthode de mise en œuvre n'est pas assez précise pour voir la fourrure de la souris, sa taille et son comportement.

Au chapitre 3, nous avons expliqué les méthodes de détection d'objet et de suivi d'objet implémentées dans ce projet. Nous estimons que la meilleure façon consiste à utiliser un ensemble d'algorithmes de détection et de suivi d'objets parce que les algorithmes de suivi peuvent aider quand la détection échoue et inversement. Pour suivre la souris, nous avons utilisé l'algorithme « AdaBoost en ligne ». La caméra est fixée sur le dessus de la cage pour détecter la souris, puis nous supprimons l'arrière-plan avec les algorithmes présentés dans [88]. Par la suite, nous avons trouvé le bord de la souris avec l'algorithme Canny et dessiné le cadre de sélection de la souris, puis, une fois que nous avons obtenu le cadre de sélection de la souris, nous avons estimé le comportement de la souris.

Dans le chapitre 4, les résultats expérimentaux ont été exposés. Les résultats obtenus montrent que notre système est précis, il fonctionne assez rapidement et il n'est pas cher, car il n'a pas besoin de matériel complexe comme cage ou de caméra spéciale. En comparant les performances de la solution présentée avec des algorithmes de suivi AdaBoost en ligne ou de détection Canny pour trouver un

sujet dans une image, nous avons constaté que notre approche fonctionne mieux que les tentatives de trouver le sujet avec des algorithmes AdaBoost en ligne ou Canny séparément.

# Bibliographie

- [1] G. Gagnon-Turcotte, Y. LeChasseur, C. Bories, Y. De Koninck, and B. Gosselin, “An optimized adaptive spike detector for behavioural experiments,” in *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, pp. 1098–1101, IEEE, 2016.
- [2] R. Ameli, A. Mirbozorgi, J.-L. Néron, Y. LeChasseur, and B. Gosselin, “A wireless and battery-less neural headstage with optical stimulation and electrophysiological recording,” in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pp. 5662–5665, IEEE, 2013.
- [3] T.-H. Ou-Yang, M.-L. Tsai, C.-T. Yen, and T.-T. Lin, “An infrared range camera-based approach for three-dimensional locomotion tracking and pose reconstruction in a rodent,” *Journal of neuroscience methods*, vol. 201, no. 1, pp. 116–123, 2011.
- [4] R. Farah, J. P. Langlois, and G. Bilodeau, “Catching a rat by its edglets,” *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 668–678, 2013.
- [5] “Computer, video, and data acquisition systems.” [Online]. Available : <https://www.noldus.com/animal-behavior-research/accessories/computer-video-and-daq-systems>. [accessed April 17, 2019].
- [6] “Video tracking software.” [Online]. Available : <http://www.noldus.com/animal-behavior-research/products/ethovision-xt>. [accessed April 17, 2019].
- [7] H. Grabner, M. Grabner, and H. Bischof, “Real-time tracking via on-line boosting.,” in *Bmvc*, vol. 1, p. 6, 2006.
- [8] “White mouse in graycage.” [Online]. Available : <https://www.youtube.com/watch?v=zIuenL5Fek>. [accessed March 17, 2019].
- [9] “Crim13 caltech resident-intruder mouse dataset, [http://www.vision.caltech.edu/video\\_datasets/crim13/crim13/main.html](http://www.vision.caltech.edu/video_datasets/crim13/crim13/main.html).” [Online]. Available : [http://www.vision.caltech.edu/Video\\_Datasets/CRIM13/CRIM13/Main.html](http://www.vision.caltech.edu/Video_Datasets/CRIM13/CRIM13/Main.html). [accessed April 17, 2019].
- [10] G. Gagnon-Turcotte, A. A. Kisomi, R. Ameli, C.-O. D. Camaro, Y. LeChasseur, J.-L. Néron, P. B. Bareil, P. Fortier, C. Bories, Y. De Koninck, *et al.*, “A wireless optogenetic headstage with multichannel electrophysiological recording capability,” *Sensors*, vol. 15, no. 9, pp. 22776–22797, 2015.

- [11] K. L. Montgomery, A. J. Yeh, J. S. Ho, V. Tsao, S. M. Iyer, L. Grosenick, E. A. Ferenczi, Y. Tanabe, K. Deisseroth, S. L. Delp, *et al.*, “Wirelessly powered, fully internal optogenetics for brain, spinal and peripheral circuits in mice,” *Nature methods*, vol. 12, no. 10, p. 969, 2015.
- [12] G. Gagnon-Turcotte and B. Gosselin, “Multichannel spike detector with an adaptive threshold based on a sigma-delta control loop,” in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pp. 7123–7126, IEEE, 2015.
- [13] B. Gosselin and M. Sawan, “Adaptive detection of action potentials using ultra low-power cmos circuits,” in *2008 IEEE Biomedical Circuits and Systems Conference*, pp. 209–212, IEEE, 2008.
- [14] V. Villette, M. Levesque, A. Miled, B. Gosselin, and L. Topolnik, “Simple platform for chronic imaging of hippocampal activity during spontaneous behaviour in an awake mouse,” *Scientific reports*, vol. 7, p. 43388, 2017.
- [15] B. Gosselin and M. Sawan, “A low-power integrated neural interface with digital spike detection and extraction,” *Analog Integrated Circuits and Signal Processing*, vol. 64, no. 1, pp. 3–11, 2010.
- [16] G. Turcotte, C.-O. D. Camaro, A. A. Kisomi, R. Ameli, and B. Gosselin, “A wireless multichannel optogenetic headstage with on-the-fly spike detection,” in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, pp. 1758–1761, IEEE, 2015.
- [17] I. Obeid and P. D. Wolf, “Evaluation of spike-detection algorithms for a brain-machine interface application,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 905–911, 2004.
- [18] B. Gosselin and M. Sawan, “An ultra low-power cmos automatic action potential detector,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 17, no. 4, pp. 346–353, 2009.
- [19] J. Choi and T. Kim, “Neural action potential detector using multi-resolution teo,” *Electronics Letters*, vol. 38, no. 12, p. 1, 2002.
- [20] S. Farshchi, A. Pesterev, P. Nuyujukian, E. Guenterberg, I. Mody, and J. W. Judy, “Embedded neural recording with tinyos-based wireless-enabled processor modules,” *Ieee Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 2, pp. 134–141, 2010.
- [21] “Triangle biosystems international.” [Online]. Available : <https://www.trianglebiosystems.com/>. [accessed April 17, 2019].
- [22] “Blackrock microsystems.” [Online]. Available : <https://blackrockmicro.com/>. [accessed April 17, 2019].
- [23] C. T. Wentz, J. G. Bernstein, P. Monahan, A. Guerra, A. Rodriguez, and E. S. Boyden, “A wirelessly powered and controlled device for optical neural control of freely-behaving animals,” *Journal of neural engineering*, vol. 8, no. 4, p. 046021, 2011.
- [24] “Kendall research systems.” [Online]. Available : <http://kendallresearchsys.com>. [accessed April 17, 2019].

- [25] M. Kiani, U.-M. Jow, and M. Ghovanloo, "Design and optimization of a 3-coil inductive link for efficient wireless power transmission," *IEEE transactions on biomedical circuits and systems*, vol. 5, no. 6, pp. 579–591, 2011.
- [26] A. K. RamRakhyani, S. Mirabbasi, and M. Chiao, "Design and optimization of resonance-based efficient wireless power delivery systems for biomedical implants," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 5, no. 1, pp. 48–63, 2011.
- [27] R. S. Bains, H. L. Cater, R. R. Sillito, A. Chartsias, D. Sneddon, D. Concas, P. Keskkivali-Bond, T. C. Lukins, S. Wells, A. Acevedo Arozena, *et al.*, "Analysis of individual mouse activity in group housed animals of different inbred strains using a novel automated home cage analysis system," *Frontiers in behavioral neuroscience*, vol. 10, p. 106, 2016.
- [28] Y. Shemesh, Y. Sztainberg, O. Forkosh, T. Shlapobersky, A. Chen, and E. Schneidman, "High-order social interactions in groups of mice," *Elife*, vol. 2, p. e00759, 2013.
- [29] R. S. Bains, S. Wells, R. R. Sillito, J. D. Armstrong, H. L. Cater, G. Banks, and P. M. Nolan, "Assessing mouse behaviour throughout the light/dark cycle using automated in-cage analysis tools," *Journal of neuroscience methods*, vol. 300, pp. 37–47, 2018.
- [30] G. Gagnon-Turcotte, M. N. N. Khiarak, C. Ethier, Y. De Koninck, and B. Gosselin, "A 0.13- $\mu$ m cmos soc for simultaneous multichannel optogenetics and neural recording," *IEEE Journal of Solid-State Circuits*, no. 99, pp. 1–14, 2018.
- [31] G. Gagnon-Turcotte, Y. LeChasseur, C. Bories, Y. Messaddeq, Y. De Koninck, and B. Gosselin, "A wireless headstage for combined optogenetics and multichannel electrophysiological recording," *IEEE transactions on biomedical circuits and systems*, vol. 11, no. 1, pp. 1–14, 2017.
- [32] E. Maghsoudloo, G. Gagnon-Turcotte, Z. Rezaei, and B. Gosselin, "A smart neuroscience platform with wireless power transmission for simultaneous optogenetics and electrophysiological recording," in *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*, pp. 1–5, IEEE, 2018.
- [33] S. A. Mirbozorgi, H. Bahrami, M. Sawan, and B. Gosselin, "A smart cage with uniform wireless power distribution in 3d for enabling long-term experiments with freely moving animals," *IEEE transactions on biomedical circuits and systems*, vol. 10, no. 2, pp. 424–434, 2016.
- [34] H. Pistori, V. V. V. A. Odakura, J. B. O. Monteiro, W. N. Gonçalves, A. R. Roel, J. de Andrade Silva, and B. B. Machado, "Mice and larvae tracking using a particle filter with an auto-adjustable observation model," *Pattern Recognition Letters*, vol. 31, no. 4, pp. 337–346, 2010.
- [35] W. N. Gonçalves, J. B. O. Monteiro, J. de Andrade Silva, B. B. Machado, H. Pistori, and V. Odakura, "Multiple mice tracking using a combination of particle filter and k-means," in *Computer Graphics and Image Processing, 2007. SIBGRAPI 2007. XX Brazilian Symposium on*, pp. 173–178, IEEE, 2007.

- [36] H. Ishii, M. Ogura, S. Kurisu, A. Komura, A. Takanishi, N. Iida, and H. Kimura, “Development of autonomous experimental setup for behavior analysis of rats,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 4152–4157, IEEE, 2007.
- [37] N. Yuman, I. Idaku, Y. Kenkichi, T. Takeshi, O. Kensuke, and M. Hiroshi, “High-speed video analysis of laboratory rats behaviors in forced swim test,” in *Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on*, pp. 206–211, IEEE, 2008.
- [38] Y. Nie, I. Ishii, K. Yamamoto, K. Orito, and H. Matsuda, “Real-time scratching behavior quantification system for laboratory mice using high-speed vision,” *Journal of real-time image processing*, vol. 4, no. 2, pp. 181–190, 2009.
- [39] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatiotemporal features,” in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pp. 65–72, IEEE, 2005.
- [40] S. Belongie, K. Branson, P. Dollár, and V. Rabaud, “Monitoring animal behavior in the smart vivarium,” in *Measuring Behavior*, pp. 70–72, Wageningen The Netherlands, 2005.
- [41] Q. Shi, S. Miyagishima, S. Fumino, S. Konno, H. Ishii, and A. Takanishi, “Development of a cognition system for analyzing rat’s behaviors,” in *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, pp. 1399–1404, IEEE, 2010.
- [42] Q. Shi, H. Ishii, S. Konno, S. Kinoshita, and A. Takanishi, “Image processing and behavior planning for robot-rat interaction,” in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, pp. 967–973, IEEE, 2012.
- [43] E. A. van Dam, J. E. van der Harst, C. J. ter Braak, R. A. Tegelenbosch, B. M. Spruijt, and L. P. Noldus, “An automated system for the recognition of various specific rat behaviours,” *Journal of neuroscience methods*, vol. 218, no. 2, pp. 214–224, 2013.
- [44] P. Dollar, Z. Tu, and S. Belongie, “Supervised learning of edges and object boundaries,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 1964–1971, IEEE, 2006.
- [45] K. Branson and S. Belongie, “Tracking multiple mouse contours (without too many samples),” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 1039–1046, IEEE, 2005.
- [46] D. R. Martin, C. C. Fowlkes, and J. Malik, “Learning to detect natural image boundaries using local brightness, color, and texture cues,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 5, pp. 530–549, 2004.
- [47] J. Brodtkin, D. Frank, R. Grippo, M. Hausfater, M. Gulinello, N. Achterholt, and C. Gutzen, “Validation and implementation of a novel high-throughput behavioral phenotyping instrument for mice,” *Journal of neuroscience methods*, vol. 224, pp. 48–57, 2014.

- [48] B. Babenko, M.-H. Yang, and S. Belongie, “Visual tracking with online multiple instance learning,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 983–990, IEEE, 2009.
- [49] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” in *European conference on computer vision*, pp. 702–715, Springer, 2012.
- [50] Z. Kalal, K. Mikolajczyk, and J. Matas, “Forward-backward error : Automatic detection of tracking failures,” in *Pattern recognition (ICPR), 2010 20th international conference on*, pp. 2756–2759, IEEE, 2010.
- [51] Z. Kalal, K. Mikolajczyk, J. Matas, *et al.*, “Tracking-learning-detection,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, p. 1409, 2012.
- [52] D. Held, S. Thrun, and S. Savarese, “Learning to track at 100 fps with deep regression networks,” in *European Conference on Computer Vision*, pp. 749–765, Springer, 2016.
- [53] E. Whitmire, T. Latif, and A. Bozkurt, “Kinect-based system for automated control of terrestrial insect biobots,” in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pp. 1470–1473, IEEE, 2013.
- [54] B. Lee, M. Kiani, and M. Ghovanloo, “A smart homepage system with 3d tracking for long-term behavioral experiments,” in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pp. 2016–2019, IEEE, 2014.
- [55] E. Machida, M. Cao, T. Murao, and H. Hashimoto, “Human motion tracking of mobile robot with kinect 3d sensor,” in *SICE Annual Conference (SICE), 2012 Proceedings of*, pp. 2207–2211, IEEE, 2012.
- [56] P. Aguiar, J. S. Cardoso, *et al.*, “A depth-map approach for automatic mice behavior recognition,”
- [57] Z. Wang, S. A. Mirbozorgi, and M. Ghovanloo, “Towards a kinect-based behavior recognition and analysis system for small animals,” in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*, pp. 1–4, IEEE, 2015.
- [58] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2000.
- [59] Y. Liu, T. S. Huang, and O. D. Faugeras, “Determination of camera location from 2-d to 3-d line and point correspondences,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 1, pp. 28–37, 1990.
- [60] M. A. Fischler and R. C. Bolles, “Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.



- [61] M. Keller and G. C. Tewinkel, *Space resection in photogrammetry*. US Department of Commerce, Environmental Science Services Administration, Coast and Geodetic Survey, 1966.
- [62] R. K. Lenz and R. Y. Tsai, “Calibrating a cartesian robot with eye-on-hand configuration independent of eye-to-hand relationship,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 9, pp. 916–928, 1989.
- [63] “Intellicage.” [Online]. Available : <https://www.tse-systems.com/product-details/intellicage>. [accessed April 17, 2019].
- [64] “Ethovision xt.” [Online]. Available : <https://www.noldus.com/animal-behavior-research/products/ethovision-xt>. [accessed April 17, 2019].
- [65] “Home cage environment.” [Online]. Available : <http://cleversysinc.com/products/hardware/home-cage-environment/>. [accessed April 17, 2019].
- [66] Y. Liang, V. Kobla, X. Bai, Y. Zhang, L. S. Crnic, and W. Wolf, “Unified system and method for animal behavior characterization from top view using video analysis,” Oct. 19 2010. US Patent 7,817,824.
- [67] M. J. Black and A. D. Jepson, “Eigentracking : Robust matching and tracking of articulated objects using a view-based representation,” in *European conference on computer vision*, pp. 329–342, Springer, 1996.
- [68] J. Ho, K.-C. Lee, M.-H. Yang, and D. Kriegman, “Visual tracking using learned linear subspaces,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, pp. I–I, IEEE, 2004.
- [69] D. Ross, J. Lim, and M.-H. Yang, “Adaptive probabilistic visual tracking with incremental subspace update,” in *European conference on computer vision*, pp. 470–482, Springer, 2004.
- [70] J. Wang, X. Chen, and W. Gao, “Online selecting discriminative tracking features using particle filter,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 1037–1042, IEEE, 2005.
- [71] S. Avidan, “Support vector tracking,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.
- [72] H. T. Nguyen and A. Smeulders, “Tracking aspects of the foreground against the background,” in *European Conference on Computer Vision*, pp. 446–456, Springer, 2004.
- [73] S. Avidan, “Ensemble tracking,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 2, 2007.
- [74] R. T. Collins, Y. Liu, and M. Leordeanu, “Online selection of discriminative tracking features,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1631–1643, 2005.

- [75] O. Williams, A. Blake, and R. Cipolla, "Sparse bayesian learning for efficient visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1292–1304, 2005.
- [76] L. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 6, pp. 810–815, 2004.
- [77] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 10, pp. 1025–1039, 1998.
- [78] D. Comaniciu and P. Meer, "Mean shift analysis and applications," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1197–1203, IEEE, 1999.
- [79] J. Wu, J. M. Rehg, and M. D. Mullin, "Learning a rare event detection cascade by direct feature selection," in *Advances in Neural Information Processing Systems*, pp. 1523–1530, 2004.
- [80] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–I, IEEE, 2001.
- [81] K. Levi and Y. Weiss, "Learning object detection from a small number of examples : the importance of good features," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–II, IEEE, 2004.
- [82] F. Porikli, "Integral histogram : A fast way to extract histograms in cartesian spaces," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 829–836, IEEE, 2005.
- [83] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.
- [84] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [85] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using adaptive tracking to classify and monitor activities in a site," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pp. 22–29, IEEE, 1998.
- [86] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *cvpr*, p. 2246, IEEE, 1999.
- [87] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

- [88] P. KaewTraKulPong and R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection,” in *Video-based surveillance systems*, pp. 135–144, Springer, 2002.
- [89] T. Horprasert, D. Harwood, and L. S. Davis, “A statistical approach for real-time robust background subtraction and shadow detection,” in *Ieee iccv*, vol. 99, pp. 1–19, Citeseer, 1999.
- [90] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.
- [91] “Motion tracking system.” [Online]. Available : [https://www.youtube.com/watch?v=F14CAm0wb\\_M](https://www.youtube.com/watch?v=F14CAm0wb_M). [accessed April 17, 2019].