

Perception chimique chez la larve de poisson zèbre

Benjamin Gallois

28 août 2020

Abstract

Dedication

Acknowledgements

Table des matières

I	FastTrack	6
1	Introduction	7
1.1	Le tracking vidéo	7
1.2	Le tracking, un problème compliqué	8
1.3	Les logiciels existants	10
1.3.1	Les logiciels propriétaires	10
1.3.2	Les logiciels open-sources	11
1.4	FastTrack : une approche différente	14
2	Conception et réalisation	16
2.1	Outils utilisés	16
2.2	Implémentation	17
2.2.1	Détection	19
2.2.2	Association	22
2.2.3	Post-traitement	23
2.3	Déploiement	24
2.3.1	CI/CD	24
2.3.2	Documentation	25
3	Base de données de films	26
4	Résultats	28
4.1	Analyse de la base de données	28
4.2	Estimation de la charge de travail de correction	29
4.3	Optimisation des paramètres	30
5	Perspective	35

II	Dual	36
6	Introduction	37
6.1	La perception chimique	37
6.1.1	L'olfaction	38
6.1.2	La gustation	40
6.1.3	Le sens chimique commun	41
6.2	Réponse comportementale	41
6.2.1	Dispositifs expérimentaux	41
7	Dispositif expérimental	47
7.1	Dual	47
7.1.1	Introduction	47
7.1.2	Philosophie	47
7.1.3	Construction	48
7.1.4	Utilisation	50
7.2	The Tropical River	51
7.2.1	Description	51
7.2.2	Utilisation et limitation	53
8	Résultats	55

Première partie

FastTrack

Chapitre 1

Introduction

Talk is cheap. Show me the code.

Linux Torvald

1.1 Le tracking vidéo

Le tracking d'objets à partir d'enregistrements vidéo est un problème qui a beaucoup gagné en popularité ces dernières années. Cela est dû en grande partie à son grand potentiel, aussi bien en science que pour des applications commerciales ou de sécurité. On citera par exemple le cas des voitures autonomes [38] qui peuvent se conduire seul, ou bien le projet ATTOL [3] de la compagnie Airbus qui permet un décollage, un atterrissage et roulage au sol entièrement automatisé reposant uniquement sur de l'analyse d'images. Une grande partie des efforts de recherche est mise sur la création d'algorithmes de reconnaissance et de tracking de piétons [], dans le but d'automatiser l'analyse des données de vidéo surveillance. Le tracking est aussi très utilisé dans le domaine du cinéma et des effets spéciaux (VFX). Que ce soit pour stabiliser les prises de vues ou bien réaliser des effets spéciaux (ex. motion capture). Dans ce cas, le tracking automatisé sur images permet de réduire les coûts et les temps de production. En science, l'utilisation du tracking automatisé, surtout en biologie et en écologie [10], est un domaine en plein essor. Il évite de devoir marquer de manière invasive les animaux et donc de les perturber. Il permet de générer un grand nombre de données fiables,

de réduire les biais, en évitant au passage une analyse manuelle longue et laborieuse.

Le tracking d'objets peut être séparé en deux catégories : le Single Objet Tracking (SOT), où le but est de détecter un seul objet dans une scène plus ou moins complexe, et le Multiple Object Tracking (MOT) [21] où il faut cette fois-ci détecter et suivre plusieurs objets. On se placera ici dans le cadre du MOT en quasi deux dimensions appliquée en grande majorité au domaine scientifique. Cela réduit grandement la difficulté du problème, en général la qualité des images est bonne, la caméra est fixe, et on peut optimiser l'éclairage pour faciliter la détection des objets. En contrepartie, la tolérance aux erreurs est faible si on veut pouvoir produire des données fiables, et donc des conclusions scientifiques robustes. Un point déterminant est la performance de l'algorithme qui doit pouvoir analyser les données dans un temps raisonnable comparé à leur débit de production, et répondre aux contraintes matérielles et techniques de l'utilisateur. La facilité d'installation et d'utilisation du logiciel qui intègre l'algorithme ne doit pas être négligée. Les utilisateurs amenés à utiliser ces logiciels ne sont en général pas des experts du domaine informatique et d'analyse d'images et le logiciel doit être facilement installable et utilisable par tous.

On verra d'abord en quoi le tracking est un problème complexe et, comment on peut réduire ou contourner cette complexité. En présentera ensuite une liste non exhaustive des logiciels existants en matière de tracking appliqué aux domaines scientifiques. Enfin on verra en quoi l'approche de FastTrack, le logiciel que nous avons créé pour résoudre le problème du tracking, est différente et dans quels cas elle peut être utile.

1.2 Le tracking, un problème compliqué

Le tracking d'objets à partir d'images repose sur trois étapes clés : l'acquisition des images qui en fonction des paramètres d'acquisition conditionnera la difficulté du tracking et le type d'algorithme utilisable, la détection des objets qui consiste à séparer les objets du fond, et enfin l'assignation des objets d'une image sur l'autre permettant de conserver leurs identités. La tracking d'objet est en général un problème de traitement d'images complexe [10]. D'une part en fonction des objets étudiés, par exemple en biologie les animaux sont des objets grandement déformables ce qui complique la

détection. D'autre part la scène peut être complexe, avec des objets disparaissant derrière des éléments du décor, se superposant (phénomène dit d'occlusion) ou entrant et sortant du champ de vue en permanence, complique l'étape d'association.

Les problèmes de détection d'objets peuvent en général être contournés par la conception de l'expérience quand cela est possible. Un point de vue fixe et une optimisation de l'éclairage permettent en général une simple détection par soustraction d'une image de fond (sans objets) et application d'un seuil. Pour les cas plus compliqués, une grande variété algorithme est disponible [42] et applicable suivant la qualité des images. Les plus courants sont la détection de points d'intérêt de l'objet, invariant par rapport au point de vue et de l'illumination, cette technique nécessite une bonne qualité d'image. La segmentation permet de séparer l'image par zone de similarités et donc de détecter les objets d'intérêts, de nombreux algorithmes et approches existent pour segmenter une image. Le machine learning peut aussi être appliqué pour la détection des objets [43].

Deux grandes classes d'algorithmes se distinguent pour mitiger les problèmes d'association. La première classe d'algorithme utilise les quantités cinématiques de l'objet, comme sa vitesse ou sa position [30, 32], pour prédire ou retrouver à postériori la position de l'objet et ainsi garder son identité. Le taux d'erreurs de cette méthode reste constant quand on augmente le nombre d'individus, elle est en général rapide et cela en fait donc un bon candidat pour les applications de tracking en temps réel. L'inconvénient majeur de cette approche vient du phénomène de propagation des erreurs. Si l'algorithme fait une erreur dans l'assignation, alors il n'a aucun moyen de corriger l'erreur à l'étape suivante et elle se propage jusqu'à la fin de l'analyse. Une deuxième classe d'algorithme est basée sur la reconnaissance de points d'intérêts de l'objet, permettant ainsi de créer une "empreinte digitale" unique à chaque objet, en utilisant soit une méthode classique [5, 29], soit du machine learning [22, 34]. Cela permet de solutionner le problème de la propagation des erreurs d'assignation et également de suivre des objets dans le temps, par exemple un animal d'un jour d'expérience sur l'autre, ce qui peut être très utile notamment pour certaines applications d'étude du comportement. Cette méthode nécessite une image de qualité suffisante pour détecter des marqueurs représentatifs de l'objet. En général, elle demande aussi plus de ressources matérielles, donc une analyse qui ne peut pas être faite en temps réel. Mais la principale limitation est le nombre d'objets qu'elle

peut suivre, il est actuellement limité à une dizaine d'objets par image avant que la performance des algorithmes se dégradent significativement avec les méthodes classiques. L'approche par machine learning permet d'augmenter le nombre d'objets au prix d'un temps de calcul long et la nécessité d'avoir recours à des machines très performantes.

1.3 Les logiciels existants

De nombreux logiciels de tracking existent déjà, on fera ici une liste non exhaustive des plus populaires en les séparant en deux catégories : les logiciels open-sources et les logiciels propriétaires.

1.3.1 Les logiciels propriétaires

Les logiciels propriétaires présentés ici sont closed-sources, l'utilisateur ne peut pas modifier le code pour adapter le logiciel à son projet. En contrepartie ils ne nécessitent aucune connaissance en informatique et bénéficie d'un service d'assistance. Ils sont une bonne alternative à d'autres options parfois difficiles à mettre en place mais leurs prix élevés peuvent être un frein pour certain laboratoire.

EthoVision XT EthoVision XT est un logiciel développé par la société Noldus. Il accompagne l'utilisateur de l'acquisition des images grâce à un système de templates d'expériences, jusqu'à l'analyse des données avec un module permettant de visualiser les quantités intéressantes. Le logiciel est très complet et très largement utilisé. Il est plutôt spécialisé dans le domaine des neurosciences comportementales. Il inclut des modules pour les expériences typiques comme un "module d'interaction social pour rats", "water maze" etc... Il permet aussi d'effectuer le tracking en direct, ce qui permet de ne pas à avoir à enregistrer les images.

EthoVision XT est un logiciel mature, un grand nombre de modules est disponible ainsi qu'un système permettant de créer soit-même des templates d'expériences. Le plus gros inconvénient est que l'utilisateur ne peut pas modifier le logiciel ou contrôler comment le tracking y est effectué. Le prix peut-être une barrière pour certains utilisateurs, le logiciel coûtant au minimum 5 850 USD sans module. Il est compatible avec Windows uniquement.

Any-maze Any-maze est un logiciel développé par la société Stoelting Co. Any-maze est spécialisé dans l'analyse des expériences de neuroscience comportementale. Il intègre directement des outils pour les tests les plus connus tel que le "forced swim test" ou le "fear conditionning test" permettant une analyse entièrement automatique de ces expériences. Il peut faire du tracking en temps réel ou depuis des vidéos déjà enregistrées.

Any-maze est très complet dans l'analyse des expériences typiques de comportement et peut être acheté avec le dispositif expérimental déjà optimisé et calibré pour le logiciel. La suite Any-maze se décompose en trois logiciels, la partie tracking est disponible au prix de 6 495 USD ou 1 500 USD l'année. Le logiciel est disponible pour Windows uniquement.

1.3.2 Les logiciels open-sources

Les logiciels open-sources permettent à l'utilisateur de modifier et distribuer le logiciel. C'est la meilleure alternative aux logiciels commerciaux. En contrepartie aucune assistance n'est fournie, le développement collaboratif de certains logiciels (par exemple sur Github) permet de signaler des bugs et de participer à leur correction et donc d'améliorer le logiciel.

idTracker idTracker [29] est une bibliothèque MATLAB qui permet de tracker plusieurs objets. Il est basé sur l'extraction d'une "empreinte digitale" pour chaque objet, et donc une détection fiable sans propagation d'erreurs. L'avantage d'idTracker est qu'il peut reconnaître un objet sur plusieurs vidéo et après une relativement longue période de temps, ce qui peut être utile pour suivre le comportement d'individus sur plusieurs séries d'expériences s'étalant dans le temps. IdTracker est cependant limité par le nombre d'objets qu'il peut tracker, actuellement une vingtaine, dû à la longueur des vidéos nécessaire à l'extraction optimale de "l'empreinte digitale" de chaque objet, qui peut aller jusqu'à 30 minutes minimum pour une grande densité d'objet. La qualité requise des images est un facteur important et doit au moins être de 150 pixels par animales. Le temps de calcul est relativement long, de l'ordre de 0.5 à 2 secondes par image et nécessite une grande quantité de RAM. L'installation d'idTracker peut se faire sans avoir besoin d'acheter MATLAB grâce au Matlab Run Time Compiler mais uniquement sous Windows. Il est par conséquent nécessaire d'acheter une licence MATLAB pour les autres plateformes, et d'avoir une connaissance minimale du langage pour mettre

en place idTracker.

ToxTrack ToxTrack [33] est un logiciel qui implémente dans une interface graphique l'algorithme ToxId [32]. L'algorithme extrait les objets du fond en appliquant un seuil. Les morceaux de trajectoire entre chaque occlusion sont ensuite divisés en trajectoires courtes et trajectoire longues basé sur une durée seuil définis par l'utilisateur. Un groupe de trajectoires longues où tous les individus sont observés en même temps est alors extrait, dans ce cas là l'assignation est fait grâce à l'algorithme Hongrois. Les trajectoires courtes sont ensuite assignées à l'objet correspondant par corrélation entre une matrice d'identification des trajectoires. Les auteurs rapportent que ToxId est aussi performant que les logiciels déjà existants, très rapide, il est possible de tracker les objets en temps réel. Un désavantage que l'on peut constater dans cet algorithme est qu'il ne marche seulement que pour un nombre constant d'animaux. L'initialisation de l'algorithme nécessite d'avoir à un instant t l'ensemble des objets à tracker simultanément détectable pendant un temps $t + dt$ défini par l'utilisateur. Le gui inclus des outils permettant de définir des zones d'intérêt ainsi qu'une analyse statistique des données recueillies. On constate que la matrice de corrélation de trajectoire ne permet pas une reconnaissance de l'objet et que les erreurs avec cette méthode sont propagatives. Le logiciel est disponible pour Windows et compilable pour les autres systèmes.

DeepLabCut DeepLabCut [22] est un framework qui permet de résoudre le problème dit de "pose estimation" qui consiste à retrouver un objet et sa position, ou une partie d'objet, dans une image. Il peut directement s'apparenter au problème du SOT si les objets à tracker sont différents, exemple une oreille droite et un nez de souris, qui pourront ensuite être retrouvé sur chaque image puis associées dans le cas où il n'y a qu'une seule souris. Dans le cas de plusieurs objets similaires à retrouver et associer d'une image sur l'autre (MOT), il faudra combiner cette détection avec une association pour obtenir le tracking. Même si DeepLabCut répond à un problème légèrement différent, il peut, par sa conception être couplé à un algorithme d'association externe pour en faire un logiciel de tracking.

DeepLabCut est directement basé sur l'algorithme de détection de feature du framework DeeperCut [16], spécialisé dans la détection des mouvements du corps humain. Les auteurs de DeepLabCut ont étudié les performances

de cet algorithme appliqué au domaine des neurosciences comportementales, comme la détection du museau d'une souris, ou des pattes d'une drosophile, et ont ajouté des outils permettant de facilement entraîner l'algorithme et de tester sa robustesse. Il tire parti du deep learning pour résoudre le problème de "pose estimation". Pour rappel, le deep learning est un type d'algorithme de machine learning qui consiste à entraîner un réseau de neurones possédant plusieurs couches. Ce réseau est constitué de plusieurs ResNets (residual neural network) qui ont été pré-entraînés sur la base de données ImageNet. Des couches de déconvolution sont placées en sortie des ResNets et permettent d'obtenir la probabilité de présence de l'objet dans l'image. Le réseau est ensuite finement ajusté en l'entraînant sur des images où sont annotées les parties qui doivent être détectées. Les auteurs ont ainsi montré qu'on peut obtenir une performance au moins aussi bonne que celle d'une détection par un être humain avec très peu de données d'entraînements, de l'ordre de 200 images.

DeepLabCut, comme indiqué précédemment, est une framework, et malgré une très bonne documentation [26], il peut être difficile à mettre en place pour un utilisateur possédant peu de compétence en informatique. Le processus d'installation durant de 10 à 60 minutes et nécessitant l'installation d'un GPU et des drivers associés pour tirer au mieux parti du logiciel. De plus, l'algorithme demande beaucoup de puissance de calcul et un GPU (NVIDIA GForce 1080 8GB au moins) est fortement conseillé. Pour donner un ordre d'idée, des images de 682x540 pixels, analysées avec un GPU de dernière génération, permet d'attendre une vitesse d'analyse de 30 images par seconde, sans GPU ce temps peut être multiplié par un facteur 10 ou 100 [23].

On voit donc que DeepLabCut présente un grand intérêt pour retrouver dans une image des objets avec une grande précision. Il s'adresse plus particulièrement aux neurosciences comportementales où il permet de suivre des mouvements complexes (ex. doigt de la main chez une souris). Il ne conviendra pas aux utilisateurs avec peu de connaissances en informatique s'intéressant à des problèmes plus larges et ayant une grande quantité de données à traiter.

idTracker.ai IdTracker.ai [34] est un framework qui permet de tracker des animaux (de l'ordre d'une centaine) avec une précision quasi parfaite. IdTracker.ai tire parti du deep learning pour réaliser le tracking. Dans une première étape, chaque objet est segmenté en appliquant un seuil. Un réseau convo-

lutionnel classe ensuite chaque blob détecté comme contenant un objet seul, ou contenant plusieurs objets. Un autre réseau convolutionnel permet ensuite de trouver l'identité de chaque individu tout au long du film. Pour fonctionner, ce système nécessite assez de données pour pouvoir entraîner le réseau à reconnaître chaque individu. Les auteurs ont trouvé qu'un tracking robuste peut être obtenu avec seulement 30 images isolées de chaque individu comme base, il faut donc prévoir au minimum des vidéos de 500 images pour une dizaine d'individus avec un minimum de 25 fps. Une résolution de 300 pixels par animale est recommandée pour avoir une bonne précision de tracking. Un facteur limitant d'idTracker.ai est qu'il demande beaucoup de temps de calcul et beaucoup de RAM, les auteurs rapportent une durée de vingtaines de minutes pour 8 zebrafish et de l'ordre de 6 heures pour 100 zebrafish sur environ 2000 images hautes définitions. Même si un GUI est disponible pour aider l'utilisateur, des connaissances de base en informatique et en programmation sont nécessaires, tout comme un matériel adapté, l'utilisation d'un GPU est fortement recommandé. Il est recommandé à des utilisateurs souhaitant un tracking parfait, entièrement automatisé depuis des images de grandes qualités et des films assez long possédant une machine puissante et sans contrainte de temps.

1.4 FastTrack : une approche différente

On a vu précédemment les logiciels existant ainsi que les différents paradigmes du tracking appliqués aux sciences. On voit qu'une approche rapide, nécessitant peu de puissance de calcul, et généraliste (pouvant s'appliquer à différents systèmes, nombre d'objets variable, etc...) est manquante. Pour combler ce vide, nous avons conçu un logiciel nommé FastTrack. Ce logiciel contient deux parties distinctes :

- Une interface où sont implémentés les procédures standards d'analyse d'images permettant la détection des objets, et un algorithme de tracking qui permet de conserver l'identité des objets d'une image sur l'autre, rapide et avec un faible taux d'erreur.
- Une interface ergonomique permettant la vérification et éventuellement la correction manuelle de toutes erreurs ayant pu être commise par l'algorithme de tracking.

On remarquera ici la différence d'approche entre FastTrack et les logiciels existants. Au lieu de développer un système nécessitant une grande puissance

de calcul, longue mais apportant des résultats entièrement automatisés et d'une grande fiabilité, FastTrack implémente une méthode rapide et facile à paramétrier, très généraliste. La correction des erreurs restantes est laissé à la charge de l'utilisateur mais peut-être effectué nativement dans le logiciel, l'interface ergonomique permettant une correction rapide et efficace. Cette solution présente plusieurs avantages, le premier étant qu'elle ne nécessite aucune connaissance en programmation, n'importe quel utilisateur peut réaliser une analyse parfaite en très peu de temps. De plus, on a pu montrer que le travail de post-traitement peut être estimé a priori par une analyse des paramètres géométriques et dynamiques du système étudié ce qui permet à l'utilisateur de savoir si le logiciel est adapté à ses besoins. Pour beaucoup de systèmes étudiés, le post-traitement se résume à une rapide vérification. Dans certains cas, si le nombre d'occlusions est trop importantes et qu'une précision de tracking parfaite est nécessaire sans vouloir avoir recours à une correction manuelle, une autre solution doit être envisagée.

FastTrack est distribué sous une licence open-source et implémenté de manière modulaire et entièrement documenté. Chaque utilisateur peut ainsi modifier le logiciel à sa convenance ou y contribuer. L'algorithme de tracking est découplé de l'interface de détection et de correction ce qui rend extrêmement facile l'intégration de FastTrack dans un projet déjà existant. Le logiciel est facilement installable en moins de 5 minutes et est compatible Linux, MacOs et Windows, il peut fonctionner sur des configurations modestes et des Single Board Computer (SBC) tel que le Raspberry Pi.

Chapitre 2

Conception et réalisation

Testing can only prove the presence of bugs, not their absence.

Edsger W. Dijkstra

2.1 Outils utilisés

Le choix des outils et bibliothèques utilisés dans la conception d'un logiciel est primordial et plusieurs facteurs de sélection doivent être pris en compte. Le premier critère à considérer est celui de la licence, j'ai fait le choix de mettre le logiciel FastTrack sous licence libre (GPL3) ce qui implique que le langage utilisé ainsi que les bibliothèques soient elles aussi sous des licences compatibles (MIT, GPL, etc...). Le choix d'une licence open-source est évident dans le cas d'un logiciel scientifique, l'utilisateur pouvant alors vérifier comment le logiciel fonctionne, le modifier pour l'adapter à ses besoins ainsi que le partager. Le deuxième critère est de bien choisir les bibliothèques utilisées en envisageant le futur du logiciel de manière à ne pas avoir à changer de bibliothèques si ses capacités s'avèrent insuffisante à mesure que le logiciel évolue. On privilégiera les bibliothèques matures offrant un support sur le long terme.

Dans cette perspective, FastTrack a été implémenter en C++ [36] en utilisant les bibliothèques Qt [37] et OpenCV [6] respectivement pour l'interface

graphique et l'analyse d'image. Les tests unitaires sont réalisés à partir de la bibliothèque Google Test.

Le C++ est un langage informatique créé par Bjarne Stroustrup en 1985. Offrant de grandes performances, il est standardisé par l'International Organization for Standardization (ISO) et est le langage de choix pour les applications d'analyse d'images ainsi que la création d'interfaces graphiques complexes.

Qt est une bibliothèque d'interface graphique open-source créée par Haavard Nord et Eirik Chambe-Eng, tous deux physiciens, en 1991 alors qu'ils réalisaient un logiciel d'analyse d'images ultrasoniques. Très mature, possédant une vaste documentation, une très grande communauté, elle permet de réaliser des interfaces graphiques pour Linux, Mac et Windows avec le même code source.

OpenCV est une bibliothèque d'analyse d'images open-source créée par Intel en 1999. Très complète et efficace, elle est devenue la référence en matière d'analyse d'images aussi bien en recherche que pour la réalisation d'applications commerciales.

Google test est une suite permettant d'automatiser les tests unitaires en C++, elle est notamment utilisée par OpenCV. Le but des tests unitaires est de vérifier que chaque partie du programme fonctionne comme attendu. Cette pratique présente plusieurs avantages, les principaux étant de détecter plus facilement d'éventuelles erreurs lors de l'implémentation de nouvelles fonctionnalités, et de faciliter le développement du logiciel quand celui-ci grandit en taille pour éviter toutes inclusions d'erreurs. Cette série de tests est automatiquement effectuée à chaque nouveau commit, voir section ??.

2.2 Implémentation

Le fonctionnement de FastTrack peut être séparé en trois parties que sont : la détection des objets, l'association des objets d'une image sur l'autre, et enfin une étape de correction.

Chaque analyse commence par l'ouverture d'une séquence d'images, ou d'un film. L'utilisateur a le choix entre deux types d'interfaces, une interface interactive où il ne peut ouvrir qu'un seul film à la fois. Elle permet de voir en temps réel l'impact des paramètres sur les images ce qui facilite la

détermination des paramètres optimaux d'analyse. Une deuxième interface permet d'ouvrir simultanément une grande quantité de films, soit en donnant un fichier de paramètres ou en sélectionnant les paramètres dans l'interface. Elle est utile quand l'utilisateur veut analyser un grand nombre de films dont il connaît déjà les paramètres optimaux d'analyse.

Les deux interfaces peuvent être utilisées de manière complémentaire. L'utilisateur peut trouver les paramètres optimaux avec l'interface interactive et ensuite automatiser l'analyse d'un grand nombre de films en les ajoutant par lots dans le logiciel.

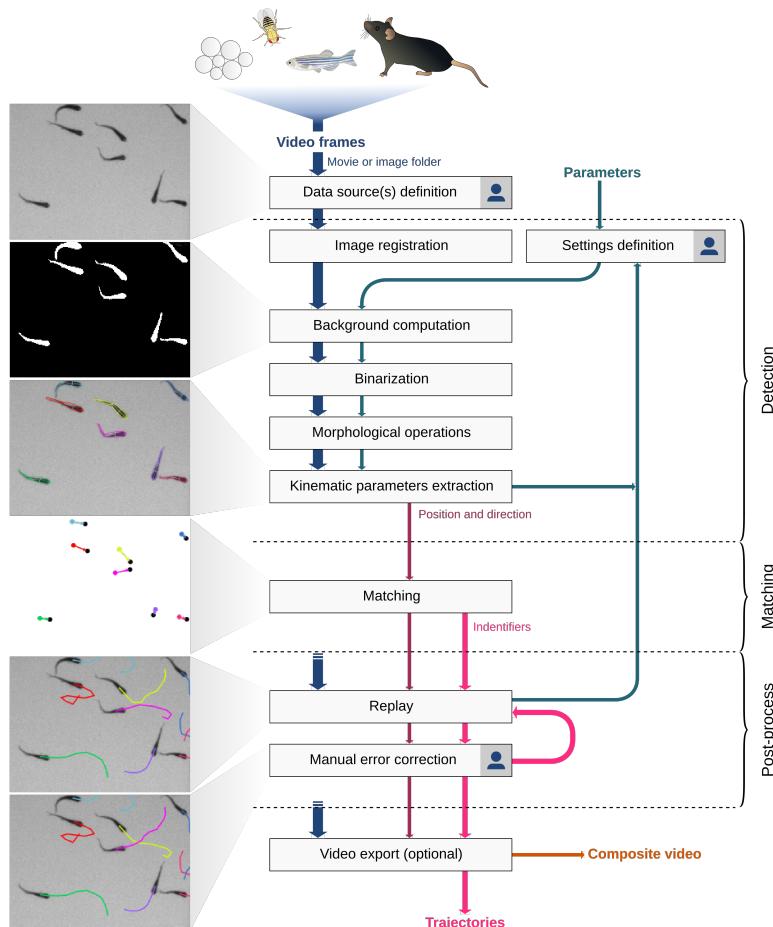


Figure 2.1 – FastTrack flux de traitement. Il se divise en trois parties distinctes : la détection, l'association et le post-traitement.

2.2.1 Détection

L'étape de détection a pour but d'extraire les paramètres cinématiques de chaque objet qui seront ensuite utilisé dans l'étape d'association. FastTrack inclut une collection de filtres d'analyse d'images permettant à l'utilisateur d'optimiser la détection des objets sans avoir à recourir à un logiciel externe.

Calcul du fond Chaque analyse commence par calculer une image de fond. Si l'utilisateur possède déjà une image de fond préalablement enregistrée, il peut directement l'ouvrir dans le logiciel. Sinon trois méthodes de calcul sont possibles :

- Projection du maximum d'intensité.
- Projection du minimum d'intensité.
- Projection de la moyenne d'intensité.

Les trois méthodes reposent sur le même principe. L'utilisateur choisit n images dans la séquence, le logiciel va projeter dans la direction perpendiculaire à l'image (le long de la composante temporelle) soit le maximum, soit le minimum ou bien la moyenne de la séquence. En pratique, on projettera le maximum (resp. minimum) si les objets sont plus foncés (resp. clairs) que le fond de manière à faire disparaître les objets et ainsi obtenir le fond. L'utilisateur peut effectuer une registration de chaque image avant projection de manière à corriger un éventuel mouvement de la caméra.

Registration L'utilisateur peut choisir d'effectuer une registration des images, trois méthodes sont proposées dans le logiciel. Chaque méthode est implémentée de manière pyramidale, c'est-à-dire que la registration est d'abord effectuée sur une image dégradée pour corriger de manière grossière le déplacement, puis la correction est affinée en augmentant la qualité de l'image jusqu'à arriver à l'image originelle. Cela permet d'accélérer le processus, la registration étant souvent un procédé relativement coûteux en temps de calcul.

La première méthode proposée est la corrélation de phase. Elle permet de corriger les mouvements de translation entre deux images en utilisant le théorème de Fourier dans le domaine fréquentiel. Cette méthode est très rapide mais reste limité à de petits mouvements de translation uniquement.

La deuxième méthode proposées est la méthode Enhanced Correlation Coefficient (ECC). Dans FastTrack, elle est restreinte à corriger les mouvements de translation et de rotation uniquement. Elle consiste à utiliser le

coefficient de corrélation comme mesure pour trouver la meilleure transformation entre deux images. Cette méthode a pour avantage d'être relativement rapide, le problème d'optimisation non linéaire pouvant être résolu de manière linéaire. Elle est performante pour des images bruitées et ayant des distorsions photométrique (contraste, luminosité).

La troisième méthode est une méthode basée sur le repérage de points clefs. Elle permet de corriger les mouvements et les déformations (homographie). Les points clés (environ 500) sont automatiquement déterminés sur deux images grâce à l'algorithme ORB [1]. Ces points sont ensuite associés deux à deux en utilisant l'algorithme RANSAC [2] permettant de trouver la meilleure transformation entre les deux images. Cette méthode, plus précise nécessite une qualité d'image suffisante pour pouvoir discerner des points clefs.

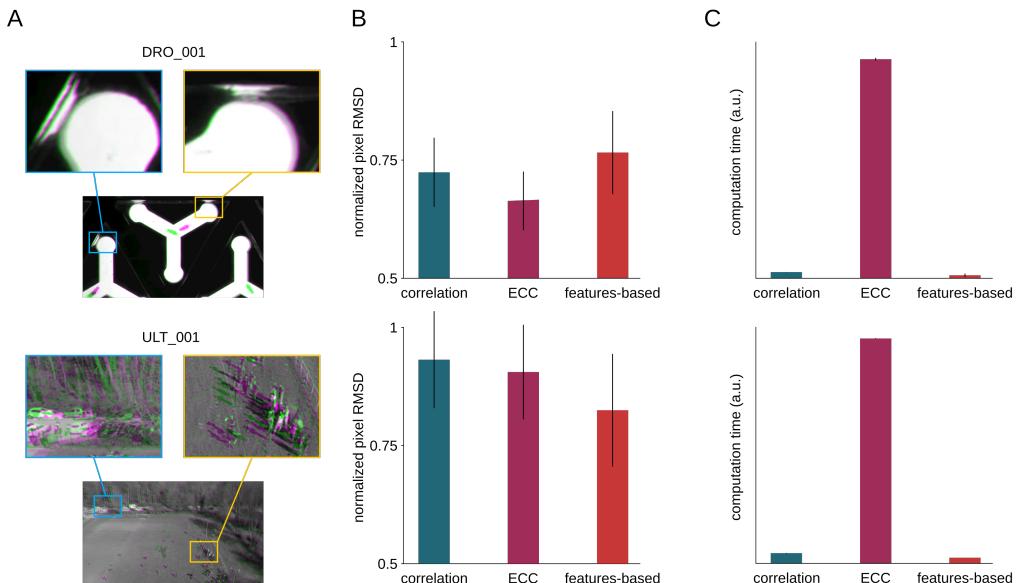


Figure 2.2 – Registrations. Deux films avec une dérive sévère sont utilisés comme référence (DRO_001 en hautn ULT_001 en bas). **(A)** Comparaison entre une image (magenta) avec la première image (vert), avec agrandissement des détails en insert. **(B)** Déviation moyenne carrée (RMSD) de l'intensité des pixels avant et après registration, moyennée sur toutes les images et normalisée par la RMSD sans registration pour les trois méthodes. Barres d'erreur : déviation standard. **(C)** Temps moyen relatif pour les trois méthodes (unités arbitraires). Barres d'erreur : déviation standard.

Binarisation Chaque image est ensuite binarisée en y soustrayant l'image de fond puis en définissant une valeur seuil. Dans le mode interactif, l'utilisateur peut voir l'impact des paramètres sur l'image ce qui permet d'ajuster facilement le seuil de binarisation. Le logiciel détecte également si le fond est plus foncé (resp. clair) que les objets permettant d'avoir à la fin de cette opération une image binaire où les pixels appartenant à l'objet sont égaux à 1 et les pixels appartenant au fond sont égaux à 0.

Opération morphologique Un ensemble d'opérations morphologiques (dilatation, érosion, ouverture, etc...) peut-être effectuées sur l'image binaire pour améliorer la détection et éliminer d'éventuels artefacts. Différentes formes et tailles de noyaux sont disponibles.

ROI L'utilisateur peut sélectionner une région d'intérêt et exclure le reste de l'image de l'analyse. Cela permet d'accélérer le processus d'analyse et d'éviter la détection d'objets parasites. En mode interactif, cette ROI peut être dessiné directement sur l'image.

Tri Pour exclure les objets trop petits correspondant à du bruit ou trop gros, par exemple deux objets superposés, l'utilisateur doit sélectionner deux tailles caractéristiques. En mode interactif, les objets sont coloriés soit en rouge, soit en vert suivant si leur taille appartient à la gamme de tailles sélectionnée.

Extraction des paramètres cinématiques Sur la base des images binaires, le logiciel va détecter le contour de chaque objet []. Une étape importante de toute procédure de tracking est l'extraction des paramètres qui serviront dans l'étape d'association. C'est en général dans le choix de ses quantités que les algorithmes de tracking diffèrent pour se spécialiser à un type d'objet donné. Dans FastTrack, les paramètres extraits sont la position du centre de masse ainsi que l'orientation de l'objet, ses quantités sont rapidement calculées et assez généralistes pour s'adapter à une grande diversité d'objets.

Pour faire cela, FastTrack calcule l'ellipse équivalente de l'objet à partir des moments d'ordre deux de l'image binaire. Cette procédure est accélérée en utilisant directement le contour grâce à la formule de Green []. L'orientation de l'objet est donnée par l'axe majeur de l'ellipse équivalente. Sa direction est

déterminée en projetant chaque pixel de l'objet sur l'axe majeur de l'ellipse. On calcule ensuite la skewness de la distribution des distances des points projetés par rapport au centre de masse. Le signe de la skewness est un indicateur robuste de l'asymétrie de l'objet par rapport à son axe majeur à partir de laquelle on peut déterminer la direction de l'objet.

Pour les objets déformables, la direction calculer précédemment peut être différente de la direction de déplacement. Par exemple dans le cas du poisson zèbre, celui-ci déforme son corps de manière périodique pour se déplacer, seul la tête est dirigée dans la même direction que le mouvement. C'est pourquoi on décompose l'objet en deux ellipses équivalentes, l'utilisateur peut alors choisir quelle ellipse représente le mieux la direction de déplacement.

2.2.2 Association

L'étape d'association a pour but de conserver l'identité des objets d'une image sur l'autre. Pour se faire, FastTrack utilise une méthode dérivée de [], qui tire parti du fait que la position et la direction de chaque objet change très peu d'une image sur l'autre. Pour chaque paire d'objet (i,j) appartenant à deux images successives, deux coûts sont calculés. Le coût "dur" calculé comme suit :

$$\begin{cases} h_{i,j} = 1 & \text{si } r_{i,j} < h_d \\ h_{i,j} = \inf & \text{sinon} \end{cases}$$

avec $r_{i,d}$ la distance entre les objets i et j , h_d un seuil qui représente la distance maximale de déplacement autorisée entre deux images successives. Le coût "mou" calculé comme suit :

$$c_{i,j} = \frac{r_{i,j}}{s_d} + \frac{\delta\alpha_{i,j}}{s_\alpha}$$

où $\delta\alpha_{i,j}$ est la valeur absolue de la différence angulaire entre les directions de i et j , s_d et s_α sont des coefficients de normalisation qui représentent un déplacement et une réorientation typique du système étudié entre deux images successives. Ces deux coûts sont ensuite multipliés élément par élément pour donner la matrice suivant :

$$C_{i,j} = \begin{cases} c_{i,j} & \text{si } r_{i,j} < h_d \\ \inf & \text{sinon} \end{cases}$$

La matrice de coup peut être rectangulaire si le nombre d'objets n'est pas constant comme lors d'occlusion ou d'ajout d'objets. Un paramètre de mémoire

peut être sélectionné de manière à ne plus pouvoir assigner les objets si ceux-ci ont disparu depuis plus de n images. La meilleure association possible est celle dont la somme des coûts est minimale. Ce problème est appelé "the rectangular assignment problem" et peut être résolu de manière exacte en utilisant l'algorithme hongrois, FastTrack utilise l'implémentation Kuhn-Munkres en C++ pour résoudre rapidement ce problème.

2.2.3 Post-traitement

Correction manuelle FastTrack intègre un outil de correction manuel du tracking. Une fois l'analyse terminée, le résultat peut être affiché dans une interface ergonomique créée à cet effet. L'utilisateur peut rejouer le film en y superposant les résultats de l'analyse, il peut sélectionner un objet pour en consulter les paramètres (aire, contour, identité, etc...). L'utilisateur peut aussi directement corriger les erreurs en supprimant des objets ou en échangeant l'identité des objets. Cette interface est conçue dans un souci d'ergonomie et de performance. Des raccourcis clavier ainsi qu'une sélection à la volée des objets directement en cliquant sur la vidéo permettent à l'utilisateur de rapidement vérifier et corriger les analyses. Il est aussi possible d'enregistrer, en plus des données brutes d'analyse, un film avec les résultats du tracking superposés. Cette interface de correction manuel permet de déplacer la charge de travail qui est traditionnellement placé sur le pré-traitement des données, vers le post-traitement. Le logiciel reste alors général et peut s'adapter à une grande diversité d'objets et reste performant grâce à une interface spécialement pensée pour réduire le temps de post-traitement.

Analyse L'analyse statistique des données n'est pas implémenté dans FastTrack. Après chaque analyse, le logiciel génère un dossier contenant les résultats. Le fichier principal est nommé tracking.txt et il contient les données brutes de l'analyse avec une image et un objet par ligne. Ce format est compatible avec tous les logiciels d'analyse les plus utilisés (R, Python, MATLAB), des exemples sont disponibles dans la documentation.

2.3 Déploiement

2.3.1 CI/CD

Le déploiement est une partie à ne pas négliger dans la conception d'un logiciel et deux aspects sont tout particulièrement important à considérer. Du point de vue de l'utilisateur, le logiciel doit pouvoir être installé facilement sur les plateformes supportées. Du point de vue du mainteneur, la partie de déploiement doit être facilement réalisable et reproductible de manière à pouvoir intégrer rapidement les correctifs et nouvelles fonctionnalités développées. C'est dans cette optique que FastTrack suit la philosophie CI/CD [35] [40] en tirant parti du nouveau système GitHub Actions.

L'intégration continue (CI) est un ensemble de pratiques qui a pour but d'intégrer rapidement les changements au projet de manière automatisée. Elle est couplée avec une automatisation des tests unitaires. FastTrack tire parti du système CI/CD de GitHub nommé Actions, à chaque nouveau changement (commit¹) ou nouvelle collaboration (pull-request²), une série de tests est automatiquement déclenchée. Ces tests vont vérifier le bon fonctionnement de l'algorithme de tracking ainsi que le formatage du code source. Seul les changements qui passent les tests peuvent être intégrés au projet ce qui garantie la reproductibilité des analyses ainsi que la cohérence du code source et de la documentation.

La livraison continue (CD) quant à elle automatise la livraison du logiciel dans sa forme finale. Elle permet d'intégrer rapidement les changements au logiciel sans avoir à le faire manuellement pour chaque plateforme supportée. Dans le cas de FastTrack, le CD est implémenté grâce à GitHub Actions et une nouvelle version du logiciel est compilée pour Linux, MacOs et Windows à chaque nouveau commit qui est intégré à la branche principale. Des versions stables du logiciel sont quant à elles compilées à chaque palier de développement. Ce système est un gain de temps majeur pour un logiciel multi-plateformes comme FastTrack, et il permet à l'utilisateur de toujours disposer des derniers correctifs et des dernières fonctionnalités.

FastTrack supporte nativement les trois plateformes majoritairement utilisées : les systèmes Linux avec une AppImage qui supportent toutes les distri-

-
1. Action d'envoyer la liste de modifications effectuées dans le système de gestion de version
 2. Action de demander l'ajout de modifications effectuer dans une branche au projet

butions et un PPA pour Ubuntu uniquement, Windows avec un installateur, MacOS avec une App. La dernière version stable peut-être téléchargée sur le site web <http://www.fasttrack.sh>, la dernière version CD sur <https://github.com/bgallois/FastTrack/releases>. La procédure pour compiler soit même le logiciel est disponible dans la documentation du développeur pour les autres plateformes.

2.3.2 Documentation

Une documentation extensive est disponible, elle se sépare en deux parties : une à l'usage des utilisateurs et une autre à l'usage des développeurs.

Utilisateur La documentation utilisateur est disponible à l'adresse <https://www.fasttrack.sh/UserManual/docs/intro.html>. Cette documentation est mise en forme à partir du logiciel Docusaurus et les utilisateurs peuvent y contribuer <https://github.com/bgallois/FastTrack/>. Elle regroupe l'ensemble des informations nécessaires à l'utilisation du logiciel ainsi que des vidéos d'explication pour aider à prendre le logiciel en mains.

Développer La documentation du développeur est disponible à l'adresse <https://www.fasttrack.sh/API/index.html>. Elle est automatiquement générée grâce au logiciel Doxygen à partir de la documentation présente dans le code source de FastTrack. Elle regroupe l'ensemble des informations nécessaires aux développeurs voulant modifier ou contribuer à FastTrack.

Chapitre 3

Base de données de films

Pour démontrer que FastTrack peut être utilisé pour analyser des films provenant d'une grande diversité de systèmes, nous avons compilé une base de données de films nommée TD^2 . Cette banque de films peut être téléchargée à l'adresse <https://data.ljp.upmc.fr/datasets/TD2/>. Les films proviennent soit de données déjà publiées dans la littérature, soit ils nous ont été fourni par les auteurs eux-mêmes sous licence CC-BY-NC-SA. Chaque film est identifié par un code à 3 lettres définissant le système (ex. ACT : active matter, ZFA : zebrafish adult, etc...) et 3 chiffres pour indexer les films provenant d'un système identique. TD^2 regroupe actuellement 41 films comprenant différents types d'objets de nature et de taille très différentes : 7 espèces d'animaux allant du poisson à la mouche, des cellules, des particules actives, des gouttes microfluidiques ainsi que des objets macroscopiques tel des joueurs d'ultimate ou des voitures. Une vidéo donnant un rapide aperçut de tous les systèmes utilisés est disponible à l'adresse suivante <http://www.fasttrack.sh/images/illustrations/mockups/trackingExample.webm>.

Un autre aspect important à considérer est le nombre d'objets par film ainsi que leurs éventuelles apparitions, disparitions et chevauchements. Dans 22 films sur 41, le nombre d'objets est variable et des objets sortent ou rentrent dans le champ de la caméra durant l'enregistrement. Dans 19 films sur 41, des objets peuvent se chevaucher créant un phénomène d'occlusion que le logiciel doit gérer pour conserver l'identité des objets.

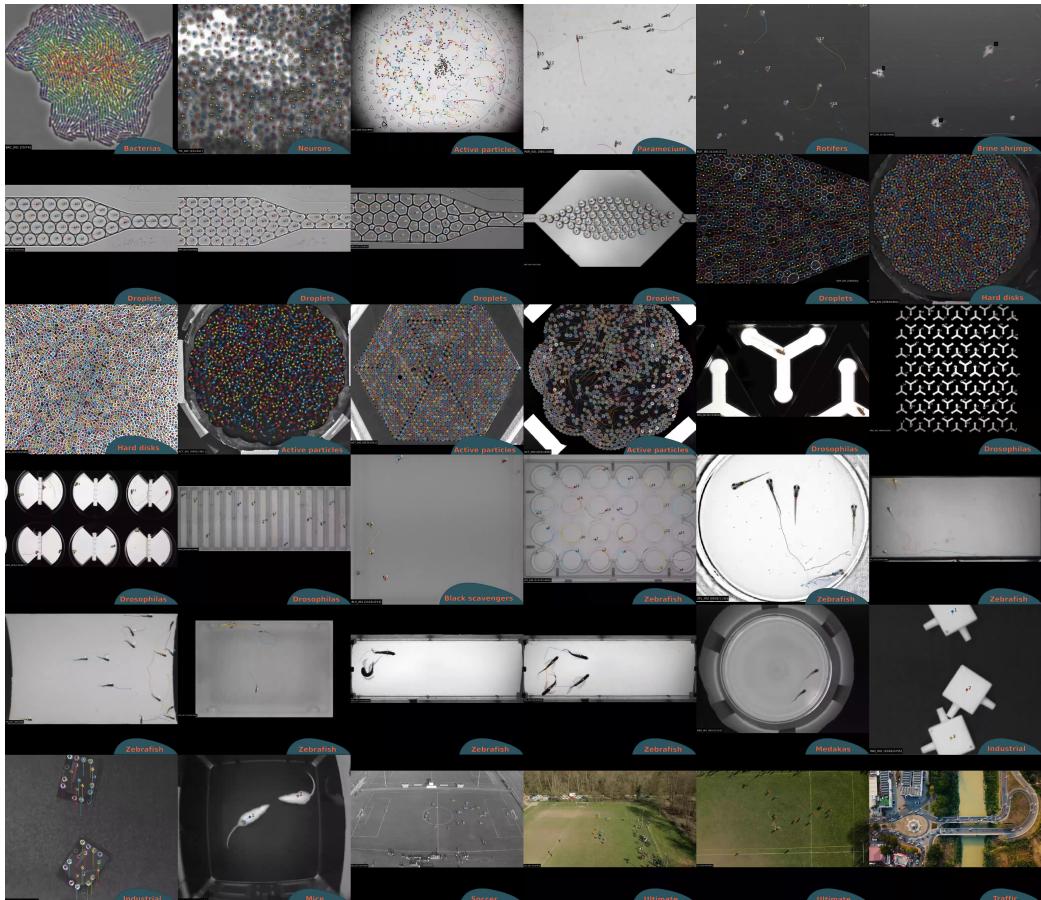


Figure 3.1 – TD².

Chapitre 4

Résultats

4.1 Analyse de la base de données

Analyser des films provenant de systèmes aussi différents que ceux compilés dans TD^2 est un véritable challenge. Cela est dû en partie aux conditions d'enregistrement qui peuvent être très diverses et qui complexifient la tâche de détection des objets. Deux difficultés récurrentes sont à soulever : les variations de l'illumination (ex. réflexion dans GRA_001, ombres dans SOC_001) et le chevauchement des objets (ex. HXB_001). Dans le cas de films provenant du milieu académique, les systèmes sont souvent conçus pour limiter ou contourner ces deux difficultés. Il est commun de trouver des films dont l'illumination est uniforme et constante. de même un confinement quasi-2D ainsi qu'un nombre d'objets restreints dans le champ de la caméra permettent de réduire le nombre d'occlusions. Dans TD^2 , 23 films ont une illumination assez bonne pour être analysés directement avec FastTrack, les autres ayant dû subir un pre-traitement individuel spécifique avant de pouvoir être analysées. Deux films ayant un nombre d'occlusions extrêmement élevé ont été écartés (HXB_001 and ZFL_001) car ne pouvant pas être analysé avec le logiciel. Les 39 films restant ont pu être analysés avec FastTrack sans difficulté. L'algorithme de Kuhn-Munkres étant de complexité $O(n^3)$ le temps de calcul est en général assez rapide. Sur l'ordinateur utilisé pour l'analyse (ordinateur de bureau moderne), on observe une vitesse d'analyse pouvant aller jusqu'à 500 images/secondes. Les films les plus volumineux en taille et nombre d'images ainsi qu'en nombre d'objets n'ont pas pris plus de quelques dizaines de minutes. Chaque film a ensuite été corrigé manuellement grâce à l'outil intégré

dans le logiciel.

4.2 Estimation de la charge de travail de correction

FastTrack est conçu pour que la phase de post-traitement soit la plus faible possible, mais cette charge de travail varie grandement suivant les films analysés. On va ici montrer que cette charge de travail peut être estimée rapidement pour un film donné. On assumera pour la suite que la détection et l'association ont déjà été réalisées.

La méthode repose sur l'évaluation de la probabilité d'incursion P_{inc} . Une incursion étant définie par la sortie d'un objet de sa cellule de Voronoï, prise en t_1 , durant un trajet entre t_1 et t_2 . Le nombre d'incursions dépend de la distribution des déplacements, de la densité d'objets, de la géométrie de la cellule de Voronoï et du degré d'alignement des déplacements. On peut écrire cette probabilité :

où $\rho = r\sqrt{d}$ est le déplacement réduit adimensionné, $R(\rho)$ la distribution des ρ et $p_{inc}(\rho)$ la probabilité géométrique d'incursion qui ne dépend que des propriétés géométriques de la disposition des objets. On calcule p_{inc} en prenant les cellules de Voronoï de tous les objets sur toutes les images, puis en déterminant la proportion des angles pour lesquels un déplacement de ρ implique une incursion dans une cellule de Voronoï voisine. Intuitivement, on peut se convaincre que p_{inc} va de 0 quand $\rho \rightarrow \infty$, à 1 quand $\rho \gg 1$. La forme de cette fonction est sensible à la densité des objets, compacts (ex. ACT_002), clairsemés (ex. PAR_001), et à la taille globale du système quand celui-ci est restreint par des murs (ex. ZFA_001).

Les distributions de ρ et p_{inc} sont représentées figure ?? pour trois films de TD^2 représentatifs de trois systèmes très différents. Il n'est pas nécessaire d'avoir un tracking parfait pour avoir une bonne estimation de cette valeur car les quelques erreurs possibles ont une influence marginale sur la distribution des ρ donc sur P_{inc} .

On a calculé P_{inc} pour tous les films de TD^2 figure ??, de manière à en estimer le nombre d'incursions n_{inc} :

$$n_{inc} = P_{inc} * N_{obj}$$

avec N_{obj} le nombre total d'objet dans le film. Chaque incursion ne résultant pas à une inversion de l'identité de deux objets, donc à une erreur, on gardera en tête que cette valeur représente la borne maximale possible d'erreurs. Dans TD^2 , seulement 5 films ont $n_{inc} > 1$ ce qui suggère que pour la grande majorité des films, le post-traitement se résume à un simple contrôle. Les 5 films restant sont connus pour leur difficulté à être analysé dont 2 (ACT_003 et ACT_004) ont nécessité le développement d'un algorithme spécifique, 2 autres (BAC_001 et ZFA_001) des logiciels dédiés pour être analysés.

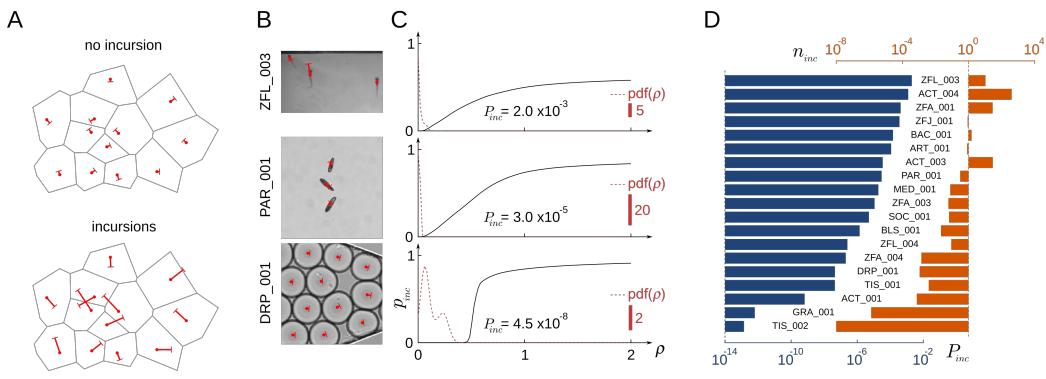


Figure 4.1 – Probabilité d'incursion (A) Définition des incursions : objets faisant de petits déplacements à l'intérieur de leurs cellules de Voronoï (haut), grands déplacements générant des incusions dans les cellules de Voronoï voisines (bas). De telle incusions sont susceptibles de générer des erreurs de tracking. (B) Images extraits de 3 films avec le déplacement instantané des objets (rouge). La position initiale est marqué d'un point. (C) Probabilité géométrique d'incusion p_{inc} en fonction du déplacement normalisé ρ (noir) et de la densité de probabilité de ρ (rouge) pour les trois films correspondant. La probabilité d'incusion P_{inc} correspond à l'aire de recouvrement des deux courbes. (D) Probabilité d'incusion P_{inc} (bleue) et nombre d'incusions n_{inc} (orange) pour 19 films de TD^2 . Le reste des films ayant $P_{inc} < 10^{-20}$

4.3 Optimisation des paramètres

L'optimisation des paramètres d'analyse peut parfois s'avérer être une tâche délicate, on présentera ici une méthode basée sur la minimisation du nombre d'inversions d'identités (swap). Les étapes suivies pour le calcul sont présentés figure ???. Pour compter le nombre d'inversions, il est seulement

nécessaire d'avoir un seul film du système étudié parfaitement analysé, ce qui peut être réalisé par un post-traitement méticuleux. On peut ensuite définir la probabilité d'inversion P_{swap} et varié la paramètre afin de la minimiser. Pour se faire, on utilisera l'interface en ligne de commande FastTrack_cli qui permet d'appeler FastTrack directement à l'intérieur d'un script Python pour automatiser la minimisation. On définit la probabilité d'inversion :

$$P_{swap} = \frac{N_{swap}}{N_{obj} - n_{ap}}$$

avec N_{swap} le nombre total d'inversions, N_{obj} le nombre total d'objets sur toutes les images, et n_{ap} le nombre de fois qu'un nouvel objet apparaît. Si le nombre d'objets est constant et noté n , alors $n_{ap} = n$ et $*N_{obj} = nT$ avec T le nombre d'images dans le film, alors on peut simplifier P_{swap} :

$$P_{swap} = \frac{N_{swap}}{n(T - 1)}$$

On a étudié en premier lieu l'impact du paramètre h_d , la distance maximale de déplacement autorisée entre deux images successives. La figure ?? représente l'évolution de P_{swap} en fonction de h_d pour trois films tirés de TD^2 . On voit que pour un faible h_d , P_{swap} est essentiellement donnée par la distribution des déplacements des objets, ce qui s'explique par le fait qu'un grand nombre d'erreurs est généré quand un objet n'est pas autorisé à se déplacer plus que son déplacement typique. Pour des h_d grands, c'est la distribution des distances aux voisins (défini par la tessellation de Voronoï) qui influence le plus P_{swap} , algorithme devenant plus sensible aux incursions et pouvant faire plus d'erreurs avec les entrées et sorties du champ de vision. Entre ces deux cas extrêmes, il existe en général un minimum, ce qui est particulièrement évident pour les systèmes denses dont la distribution des distances aux voisins est très piquée. Par exemple pour DRP_001 on voit que P_{swap} tombe à 0 pour un intervalle de h_d . La fréquence d'acquisition joue un rôle important ici, pour des films très résolus temporellement, on aura une distribution de déplacement décalée à gauche vers des distances plus courtes, cela va donner une séparation claire et de petite valeurs en P_{swap} . Pour des films avec une résolution temporelle médiocre (ZFJ_001), les deux distributions se recouvrent et P_{swap} reste toujours bloquée à de grandes valeurs.

On peut faire une analyse similaire pour les autres paramètres. On a représenté h_d en fonction de h_t (nombre d'images où l'objet peut disparaître), on voit qu'on peut trouver une bande de paramètres où P_{swap} est minimale.

Pour résumer, on peut faire cette analyse pour un film parfaitement analysé grâce à FastTrack et à l'outil de post-traitement, puis en dériver un jeu de paramètres optimaux à appliquer sur des films similaires de manière à réduire la charge de travail en post-traitement.

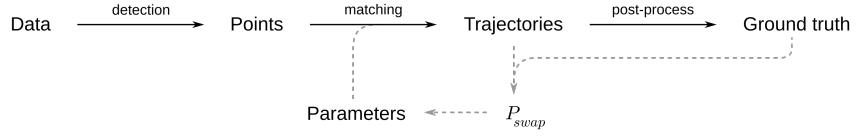
Une autre question souvent posée est de savoir avec quelle résolution temporelle acquérir le film. Une grande résolution permet de diminuer P_{swap} et donc de réduire le post-traitement mais n'est pas toujours envisageable (limitation de la caméra, de l'éclairage, nécessité d'un grand espace de stockage). Pour étudier l'impact de la fréquence d'enregistrement, on a sélectionné 7 films avec une bonne résolution temporelle et nous les avons dégradés pour à chaque fois calculer P_{swap} . Tracé en fonction du facteur de dégradation adimensionné $(f - f^*)\tau$, on voit que tous les P_{swap} suivent une courbe maîtresse :

$$L(x) = \frac{1}{1 + e^{-1}a}$$

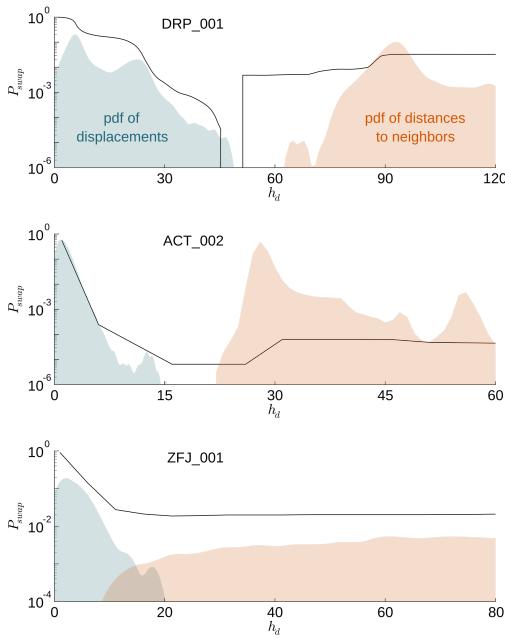
qui est une fonction logistique standard où f^* correspond au facteur de dégradation où $P_{swap} = 0.5$, τ est un facteur de normalisation qui dépend des propriétés dynamiques de chaque système.

A l'aide de ses résultats, on peut donc proposer une méthode permettant de déterminer la fréquence d'acquisition optimale. Cette méthode nécessite un seul enregistrement de bonne qualité déjà analysé avec FastTrack. Il faut ensuite répéter la procédure de dégradation et calculer le P_{swap} résultant. Les P_{swap} peuvent être ajustés à la fonction logistique pour en extraire les paramètres f^* et τ . Il suffit ensuite de choisir la fréquence d'acquisition en fonction de la quantité de travail que l'utilisateur est prêt à fournir en post-traitement $N_{swap} = P_{swap}(N_{obj} - n_{ap})$.

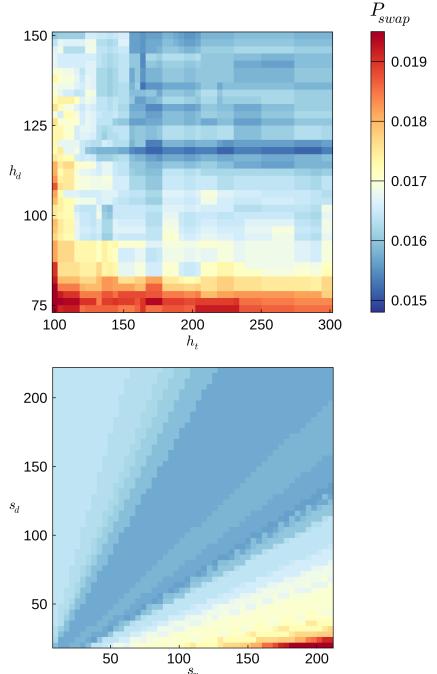
A



B



C



D

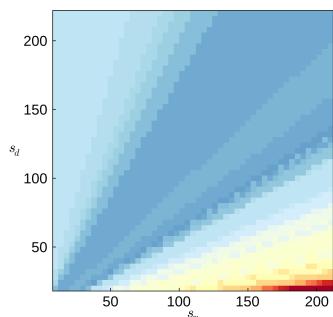


Figure 4.2 – Optimisation des paramètres d’analyse basé sur P_{swap} (A)
 Schéma du workflow d’optimisation : le résultat parfait de tracking est utilisée pour calculer P_{swap} et créer une boucle de rétroaction sur les paramètres d’analyse. (B) P_{swap} (noire) en fonction de la distance maximale h_d (en pixels) pour trois films typiques. Les lignes verticales pour DRP_001 indique que P_{swap} descend à 0. Les distributions des déplacements entre deux images successives (bleue), et de distance aux voisins (orange) sont montrées pour comparaison. (C) P_{swap} en fonction de la distance maximale h_d et du temps de disparition maximal h_t (en nombre d’images) pour PAR_001 pour $s_d = 95$ et $s_\alpha = 60$. (D) P_{swap} en fonction de la distance s_d (en pixels) et de l’angle de normalisation s_α (en degrés) pour PAR_001 avec $h_d = 210$ et $h_t = 90$.

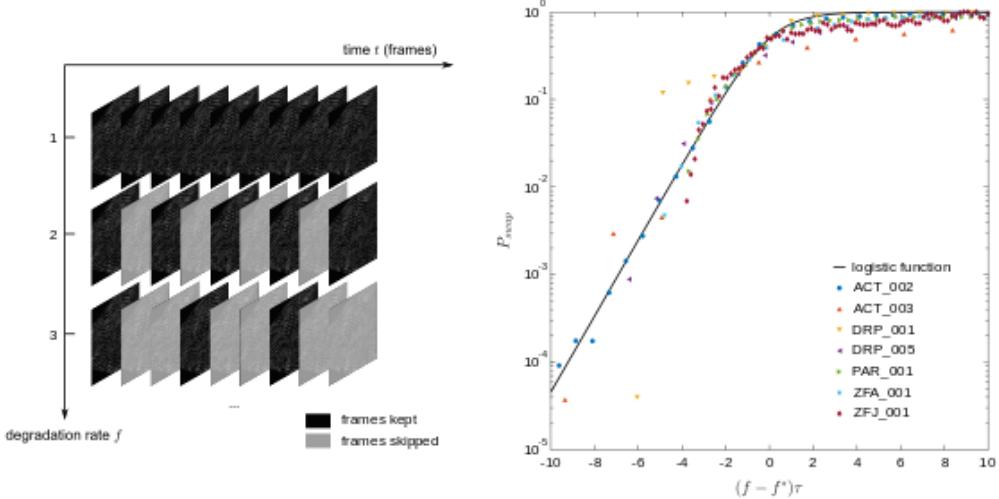


Figure 4.3 – P_{swap} et vitesse d’acquisition **(A)** Schéma de la procédure de dégradation. **(B)** P_{swap} en fonction du taux réduit de dégradation $(f - f^*)\tau$ pour 7 systèmes très différents. f^* et τ ont été déterminés par un ajustement sur la fonction logistique standard (noire).

Chapitre 5

Perspective

On a vu dans cette partie le logiciel FastTrack. On a détaillé les outils utilisés ainsi que l'implémentation en elle-même. On a ensuite testé le logiciel sur une grande quantité de films présentant des systèmes extrêmement variés et évalué sa performance. On a décrit deux méthodes permettant de trouver facilement les paramètres optimaux d'analyses et la fréquence d'acquisition optimale qui permet de guider l'utilisateur de la conception de l'expérience jusqu'à l'analyse des films.

Les codes du logiciel sont entièrement disponibles à l'adresse .. Cela permet d'une part de collaborer au projet pour régler des bogues où ajouter des fonctionnalités. D'intégrer tout ou une partie du projet dans un logiciel ou une expérience déjà existante. Cela renforce la pérennité du projet qui ne dépend pas d'une seule personne pour rester à jour. Enfin la mise en place de la livraison continue (CD) permet de s'affranchir d'un temps dédié à la création de chaque version pour les trois plateformes supportées.

La collaboration pour implémenter de nouvelles fonctionnalités, participer à la documentation ou régler des bogues est encouragée et peut se faire grâce au système de pull-request de GitHub.

Deuxième partie

Dual

Chapitre 6

Introduction

6.1 La perception chimique

La perception chimique est le plus ancien système sensoriel datant de 500 millions d'années. Il fait partie des sens nécessaires à la reproduction et à l'alimentation deux des activités les plus fondamentales d'une espèce animale. Chez le poisson, la perception chimique passe par trois organes distincts sont l'olfaction, la gustation et le sens chimique commun. Contrairement aux espèces terrestres où les substances perçues par l'olfaction et la gustation diffèrent par le moyen de transport des molécules, le poisson goûte et sens par le même milieu : l'eau. De fait, c'est la solubilité des composés dans l'eau qui détermine le type de composés pouvant être transportés et perçu. La distance parcourue ainsi que la concentration (qui déterminera le seuil de perception donc le temps de résidence du composé dans l'environnement) dépendra de la diffusion du composé ainsi que le la convection du milieu. La perception chimique est très spécifique, présent à l'état de mélange dans le milieu naturel, le sens chimique étant spécifique aux structures moléculaires des composés odorants. On s'attend donc à observer des comportements d'excitation non directionnel ou des réponses orientés de remontée de gradient pour trouver des sources, par exemple pour s'orienter ou pour retrouver un lieu de ponte lors de migration. Cette navigation est une tâche complexe et ne peut s'apparenter à une simple remonté de gradient, le milieu aquatique étant en général turbulent et la perception parcellaire.

La larve de poisson-zèbre est le modèle de choix pour étudier la navigation orientée. Étant transparente à l'état larvaire, il est possible d'observer

l'intégralité du cerveau avec une résolution cellulaire grâce à la microscopie à feuille de lumière pendant que l'animale effectue une tâche ce qui permet de relier les comportements à l'activité neuronale. Les organes sensoriels de la perception chimique ont été bien caractérisés chez le poisson-zèbre, il existe cependant peu d'étude comportementale sur la perception chimique et la navigation chimique. Le modèle est très utilisé dans l'étude des phénomènes d'dépendance dû au fait de la facilité de réaliser des manipulations génétique. La caractérisation d'un modèle animale pour étudier la navigation orientée par perception chimique reliant le comportement à l'activité neuronale serait un ajout puissant qui permettrait de mieux comprendre comment les poissons perçoivent et remonte à des sources de nourritures, à des conspécifiques pour la reproduction et à des sites de migration comme chez le saumon.

6.1.1 L'olfaction

L'organe olfactif du poisson figure 6.1 est constitué de deux structures situées dans le museau de l'animale. Chaque structure est constituée d'une cavité appelée chambre olfactive raccordée à l'extérieur par une narine d'entrée et une narine de sortie. L'intérieur de la chambre olfactive est tapissée de la rosette olfactive constituée de deux rangées de lamelles olfactives. L'épithélium olfactif, où sont situés les récepteurs olfactifs est placé sur ces lamelles. L'organisation exacte ainsi que la position de l'organe olfactif peut varier suivant l'espèce de poisson, avec par exemple l'ajout d'une cavité de ventilation dans le prolongement de la cavité olfactive.

L'épithélium olfactif à une structure en colonne stratifiée épais de $100\mu m$. Il peut être séparé en un épithélium sensoriel et un épithélium non-sensoriel. L'épithélium sensoriel est constitué de trois types de cellules : les cellules réceptrices, de supports et basales. L'épithélium non-sensoriel est constitué de cellules de Goblet et de cellules ciliées non-sensorielles.

Les cellules réceptrices sont des neurones bipolaires ciliés, microviliés [15] ou pear-shaped [39] qui expriment des récepteurs olfactifs des familles OR, V1R, V2R et TAAR. Les cellules réceptrices projettent directement dans le bulbe olfactif situé dans le cerveau, qui envoie lui-même des signaux dans le télencéphale et dans le diencéphale. Le bulbe olfactif chez les téléostéens est une structure constituée de quatre couches : ONL (olfactory nerve layer), GL (glomerular layer), MCL (mitrac cell layer) et ICL (internal cell layer). L'information olfactive est transmise par les cellules réceptrices au bulbe ol-

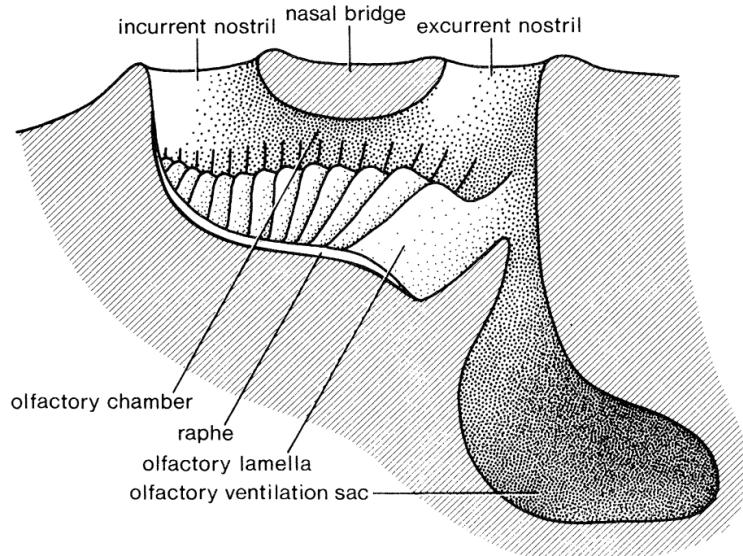


Figure 6.1 – Le système olfactif du poisson tiré de [13].

factif comme un carte topographique d'odeurs. Les connexions neuronales du bulbe olfactif ont particulièrement été étudiées chez le poisson-zèbre. Chaque cellule réceptrice exprime un seul type de récepteur olfactif. Toutes les cellules qui expriment le même récepteur projettent dans le même glomérule du bulbe olfactif. Les odeurs sont donc encodées dans une carte spatiale d'activation des glomérules. Le bulbe olfactif du poisson zèbre comprend 140 glomérules, dont 27 glomérules invariants dans leur arrangement suivant les individus [8]. Les axones des neurones du bulbe olfactif se terminent ensuite dans le télencéphale et le diencéphale.

Chez le poisson-zèbre [12], l'organe olfactif se développe à partir des plaques olfactives au stade 6-10 somites (environ 15 heures après fertilisation) du développement embryonnaire. La cavité olfactive commence à apparaître au stage 28-30 somites (31 heures après fertilisation). Environ 50 heures après fertilisation, on observe l'apparition de l'épithélium olfactif et des cellules réceptrices. Quand l'embryon émerge de l'oeuf, 4 jours après fertilisation, l'organe olfactif continue son développement morphologique mais l'organisation cytologique change peu. A 40 jours après fertilisation, le pont entre la narine d'entrée et la narine de sortie se forme, séparant ainsi les courants sortant et entrant de la cavité olfactive. L'ajout de lamelles sur la rosette olfactive se poursuit tout le long de la vie du poisson zèbre.

6.1.2 La gustation

L'organe gustatif du poisson est constitué des papilles gustatives qui sont au contact direct des substances chimiques. Elles sont distribuées sur tout le corps du poisson et plus particulièrement dans la bouche, sur les lèvres et sur la peau. Leur distribution et leur concentration varient suivant les espèces. Elles sont innervées par 3 nerfs crâniaux différents, faciale (VII), glossopharyngiale (IX) et vagale (X). Le nerf facial transmet les informations provenant des papilles gustatives extra-oral, le nerf glosopharyngial des informations provenant de l'intérieur de la cavité orale et le nerf vagal de la cavité oropharyngiale. Le système gustatif est anatomiquement divisé en deux parties distinctes, les nerfs IX et X projetant dans le lobe vagal du cerveau et le nerf IV dans le lobe facial. Les connexions à des aires plus élevées du cerveau diffèrent légèrement d'une espèce à l'autre. Il a été montré chez *Ictalurus nebulosus* [4] que ces deux systèmes ont des rôles distincts dans les comportements d'alimentation du poisson. Les projections du système gustatif du poisson-zèbre ont été étudié en détail [41] et forme un réseau complexe qui peut être résumé graphiquement figure 6.2.

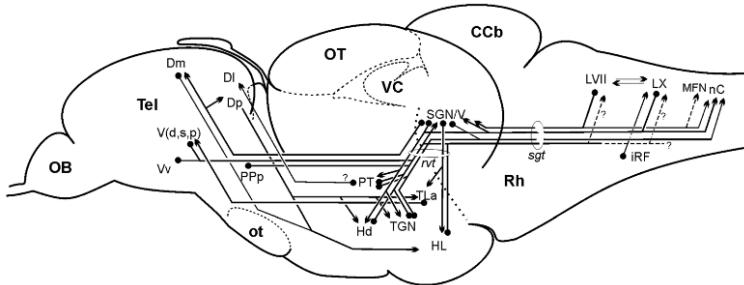


Figure 6.2 – Projection du système gustatif du poisson-zèbre tiré de [41].

Chez le poisson-zèbre, les papilles gustatives sont situées sur les lèvres, dans la cavité oropharyngiale, sur les barbillons et sur la face ventrale et dorsale de la tête [11]. Les premières papilles gustatives apparaissent à 3 où 4 jours après fertilisation, elles sont situées sur les lèvres et sur les arcs branchiaux. Les papilles gustatives dans la bouche et la cavité oropharyngiale apparaissent entre 4 et 5 jours après fertilisation. Les papilles gustatives sur la tête n'apparaissent qu'à 12 jours après fertilisation et il faudra attendre l'âge juvénile (30 à 40 jours après fertilisation) pour voir l'apparition des

barbillons. On remarquera que l'apparition des papilles gustatives coïncide avec l'apparition de l'alimentation autonome chez la larve.

6.1.3 Le sens chimique commun

Le poisson possède un troisième sens chimique nommé le sens chimique commun. Il est constitué de cellules bipolaires nommées "solitary chemosensory cells" (SCCs) présentes dans l'épiderme. Leur répartition et leur nombre varient grandement suivant les espèces.

Chez le poisson-zèbre [19], les SCCs ont été décrite comme soit un ensemble de 2-7 villosités de 0.5 à $1 \mu m$ de longueur émergeant du corps cellulaire, majoritairement présentes chez les larves, soit une seule villosité de $3\mu m$ de longueur présente en majorité chez les adultes.

Les premières SCCs apparaissent à 3 jours après fertilisation et leur densité augmente jusqu'à 25 jours après fertilisation où leur nombre se stabilise à 1.10^6 par mm^2 avec 2 à 5 fois plus de SCCs sur la tête du poisson-zèbre que sur le corps.

6.2 Réponse comportementale

Le poisson zèbre reste le modèle de choix pour étudier la réaction comportementale vis-à-vis de stimuli chimique. De fait, la larve de poisson zèbre est utilisé pour étudier des comportements tels que la réponse à des stimuli visuel, de chaleur, l'étude des réflexes de l'équilibre. Elle est transparente et on peut donc observer l'ensemble du cerveau grâce à un microscope à feuille de lumière et des rapporteurs calciques. On peut donc en théorie étudier facilement le lien entre réponse à un stimuli et activité neuronale. Pour la perception chimique, peu d'études ont été faites sur les larves de poisson zèbre, la majorité des travaux étant concentré sur le screening de drogue a visé médicale.

6.2.1 Dispositifs expérimentaux

L'étude comportementale de la perception chimique du poisson-zèbres, adultes où à l'état larvaire a été étudié par le biais de dispositif divers expérimentaux que nous presenterons dans la suite.

Conditional place preference Le conditional place preference (CPP) est un type de conditionnement Pavlovien, c'est-à-dire qu'il consiste à associer un stimulus conditionné qui a été appris à l'animale, avec un stimulus non conditionné que l'on cherche à tester. L'exemple le plus classique est d'associer le son du cloche avec le relâchement d'une odeur de nourriture. Après apprentissage l'animale répond au stimulus conditionné seul. Cette approche classique de psychologie a été appliquée chez le poisson-zèbre adulte pour tester la réponse à divers stimuli chimiques [25]. Les expériences de ce type présentent le même dispositif expérimental et son constitué de 3 étapes. Une première étape consiste à évaluer la préférence du poisson, il est placé dans un aquarium séparé en deux ou trois parties, chaque partie présente des murs avec un motif et une couleur différente. On teste le poisson pour savoir le côté qu'il préfère, généralement on élimine les poissons ayant une trop grande préférence. La deuxième étape est celle de conditionnement, pratiqué en contraignant le poisson du côté de l'aquarium qu'il préfère le moins et en y injectant la substance à tester, cette étape peut être répétée plusieurs fois. La troisième étape de test consiste à répéter la première étape pour évaluer le changement de préférence de l'animale. [9] ont étudié la sensibilité de la cocaïne sur des poissons zèbres adultes. Ils montrent sur des poissons WT une réponse forte et robuste du CPP induite par la cocaïne avec 85% des poissons changeant de préférence pour une concentration en cocaïne de $10mg.L^{-1}$. Les concentrations plus basses et plus hautes entraînant des réponses plus faibles. [24] ont étudié la réponse de poissons zèbres adultes à l'éthanol dans un dispositif avec 2 chambres avec des motifs différents et une allée centrale avec un motif uniforme permettant de placer le poisson dans un environnement neutre. L'expérience dure 2 jours avec pour le premier jour la préférence de chaque poisson est calculé à partir d'un enregistrement de 5 minutes. Le deuxième jour chaque animal est conditionné avec l'éthanol dans le côté le moins préféré pendant 20 minutes et dans le côté préféré avec juste de l'eau, l'expérience étant faite en alternant l'ordre de conditionnement. Le troisième jour, la préférence de chaque animale est calculé à partir d'un enregistrement de 5 minutes. Les auteurs montrent une réponse positive au conditionnement dès une concentration de 1.5% sans effet significatif de l'ordre du conditionnement. C'est aussi la première étude à utilisé un système automatisé de tracking pour calculer la préférence de l'animale. Des expériences similaires ont été pratiquées pour tester la réponse à divers stimuli chimiques. [27, 28] montrent une réponse positive au CPP pour la D-amphétamine. [7] montrent une réponse positive à la salvinorin A une

substance hallucinogène, à la cocaïne et à la spiradoline. [?, 17] montrent une réponse positive du CPP à la nicotine et à l'éthanol.

On voit que le CPP a été beaucoup utilisé pour étudier la réponse à un stimulus chimique du poisson zèbre. L'accent est fortement mis sur des produits provoquant des pathologies d'addiction chez l'humain. Mais ce protocole présente plusieurs défauts, le premier étant qu'il repose sur un apprentissage du poisson impliquant plusieurs systèmes de perception ainsi que la mémoire. Durant la phase de conditionnement, l'apprentissage repose sur la perception visuelle de l'environnement (motif sur les murs de l'aquarium), la perception du stimulus chimique, l'association des deux stimuli provenant d'organes sensitifs différent et dans un troisième temps d'une mémorisation de ces perceptions. Deuxièmement, la mise en place du dispositif qui dure au minimum 3 jours avec une seule phase de conditionnement est un frein pour une utilisation au débit de ce système pour étudier l'effet d'un grand nombre de produits chimiques. On remarque aussi que la mise en place d'une expérience entièrement automatisée où la préférence est calculée automatiquement à partir des enregistrements des expériences a été mise en place tardivement en 2011, rendant l'analyse des expériences laborieuses avec un comptage manuel du temps passé dans chaque compartiment.

Multiplates Un dispositif expérimental très utilisé pour quantifier l'effet d'un composé chimique sur les larves de poissons zèbres est celui du multiplate. Une larve de poisson zèbre est placé dans chaque puit d'un multiplate et un composé chimique est introduit dans le puit. On enregistre ensuite la larve nager dans le composé chimique et on extrait les paramètres cinématiques de l'animale. L'avantage de cette technique est qu'elle nécessite peu de matériel, elle produit rapidement une grande quantité de donnée avec des multiplates allant jusqu'à 48 trous par plaque. Avec ce dispositif, de nombreux composés chimiques ont été testé pour connaître leur toxicité.

Diffusion Certains auteurs ont essayé de quantifier l'effet de composé chimique en introduisant directement dans l'aquarium un composé chimique et en regardant le pourcentage de temps passé proche de la source. Notamment [39] a montré une attraction concentration dépendante du poisson zèbre adulte à l'adénosine. L'ATP l'ADP produit par les organismes vivant sont déphosphorylés dans l'épithélium olfactif et l'adénosine résultante est perçu par une cellule sensorielle nommées *pear-shaped* exprimant des récepteurs

spécifiques à l'adénosine. L'adénosine agit comme une substance attractive pour le poisson. [20] montrent chez la larve de poisson zèbre une attraction concentration dépendante au GCDA. En utilisant un dispositif avec une cloison séparant deux compartiments et une zone intermédiaire où le poisson peu choisir dans quelle zone aller une attraction à la nicotine. [14] montrent avec un système en deux zones séparées et une zone intermédiaire proche de [20] une aversion forte à la cadavérine, une odeur associée à des corps en décomposition.

Le point faible de ce type de dispositif expérimentale très facile à mettre en place est le manque de contrôle dans la concentration réelle de produit perçu par l'animale. La diffusion et la convection sont négligées dans les expériences et la concentration perçue par l'animale est mal connue et peu reproducible. [14, 20] mitige les effets de la diffusion et convection avec une paroi séparant les deux zones, laissant toujours une incertitude dans la zone intermédiaire. De plus, ces montages exclus la réalisation d'expériences longues dû à l'homogénéisation de la concentration par diffusion.

Flux Une autre classe de dispositif pour étudier la perception chimique chez le poisson zèbre est un dispositif sous flux. Divers montages expérimentaux ont été reporté dans la littérature sans qu'un standard n'émerge permettant de comparer les résultats. [1, 2] utilise un dispositif sous flux laminaire qui permet de séparer un aquarium en 2 compartiments distincts sans mur physique testé dans [31]. L'animale peut alors choisir entre les deux compartiments sans contraintes. Plusieurs substances psychoactives ont été testé sur des poissons zèbre adultes, le temps passé dans le produit, le nombre de croisements ainsi que les paramètres cinématiques de l'animale sont enregistrés. Ils montrent une attraction par le diazepam, fluoxetine, risperidone et buspirone. Une aversion pour les pH acide et pour deux extraits d'odeurs de nourriture. Un comportement neutre pour l'éthanol. Aucun changement dans la paramètres cinématique de nage sauf pour les deux extraits d'odeurs de nourriture. Toujours avec le même dispositif expérimental, les auteurs testent l'attraction ou la répulsion face à une eau conditionnée. Dans un premier temps, des poissons sont placés dans un aquarium et soumis soit à un stress chimique (diminution du pH), soit à un stress physique (chasser par un filet), soit à une image de prédateur où enfin à une privation de nourriture plus où moins importante. Dans un deuxième temps, des poissons sont testés dans le dispositif à flux laminaire pour regarder la préférence à cette

eau conditionnée. Les auteurs trouvent une aversion significative pour l'eau conditionnée par un stress chimique et physique. Aucune différence significative est trouvé pour le conditionnement par l'image d'un prédateur. Ils montrent une aversion pour l'eau conditionnée avec des poissons privés de nourriture pendant 48 heures et aucune différence pour des poissons suivant un nourrissage normal où un jeun chronique.

[18] utilisent un dispositif à flux qui au lieu de séparer l'aquarium en deux parties distinctes permet de rapidement changer la concentration en produit dans l'intégralité de l'aquarium. En enregistrant les paramètres cinématiques de l'animale, ils peuvent ensuite conclure sur l'impact du composé chimique. Les résultats ne sont plus cette fois-ci exprimés en terme d'attraction répulsion mais peuvent être comparé à ceux du multiplate, l'intérêt étant de pouvoir faire l'expérience avec des animaux adultes qui nécessite un grand volume d'eau en trois dimensions pour évoluer de manière réaliste. Avec ce dispositif, les auteurs montrent que les poissons modulent leur trajectoire en influant sur 7 paramètres indépendants que sont la vitesse, le nombre de burst, le nombre de virages, le déplacement horizontal et vertical, le temps passé figé ainsi que la position verticale. À partir de ces valeurs, ils peuvent classer les odeurs en regroupant celle entraînant la même réponse comportementale. Les odeurs de nourritures entraîne une augmentation significative de la vitesse et du nombre de burst, les odeurs sociales (provenant d'autres poissons) entraîne une réponse similaire à celle des odeurs de nourriture, les odeurs d'alertes entraîne une descente vers le fond de l'aquarium et une augmentation du temps passé figé, et les odeurs de décomposition entraîne plus de virage. Le point essentiel relevé par cette étude est la variabilité inter et intra expérience. Les auteurs montrent que moins d'un tiers des odeurs utilisées dans l'étude produisent des résultats reproductible entre essai d'un même individu. Certaines odeurs comme la cadavérine, le sang, la peau et les odeurs de nourriture entraîne des réponses consistantes pour un même individu. Le plupart des odeurs produisent des résultats peu reproductibles pour différents poissons (corrélation inférieure à 0.5). Ce type de dispositif expérimentale permet une meilleure gestion de la concentration perçue par l'animale qui est parfaitement contrôlé après un régime transitoire de mise en place. Elle permet aussi la réalisation d'expériences longues, le flux permettant de stopper la diffusion et la convection du composé chimique.

On voit donc à travers cet aperçu de la littérature scientifique que l'étude de la perception chimique et de la réponse comportementale au stimulus chi-

mique n'est pas standardisé. Aucun montage expérimental n'est standardisé et ne permet une comparaison directe entre les études effectuées dans divers laboratoires indépendants. C'est dans ce contexte que nous avons mis au point Dual, un montage expérimental facile à construire soit-même, documenté et permettant d'étudier de manière standardisée et comparable la réponse comportementale chez la larve et le poisson-zèbre juvénile.

Chapitre 7

Dispositif expérimental

7.1 Dual

7.1.1 Introduction

La reproductibilité des études comportementales est un problème ouvert en neuroscience. La conception et la réalisation de montages expérimentaux permettant un grand débit d'expériences en évitant tout biais est essentielle à la caractérisation d'un comportement. L'absence de standard en matière d'étude de la réponse comportementale à des stimuli chimiques ne permet pas à ce jour de comparer facilement les résultats obtenus au travers des diverses études menées jusqu'à présent. C'est dans ce contexte que nous présentons Dual, un montage expérimental à haut débit d'expérience facile à réaliser et à mettre en place.

On verra dans un premier temps la philosophie du dispositif, dans quel but il a été conçu et pour quelle utilisation. Dans un second temps les étapes de construction et enfin on verra son utilisation et son coût en temps pour réaliser une campagne d'expérience.

7.1.2 Philosophie

Dual est un système du même type que [31] qui consiste à créer grâce à un flux laminaire deux compartiments virtuels dans l'aquarium. Comme on l'a vu précédemment, ce système permet une connaissance rigoureuse de la concentration du composé chimique auquel est soumis l'animale. Il permet

aussi d'éviter la convection due au mouvement d'eaux engendré par la nage du poisson, toutes les perturbations étant entraînées par le flux. On a donc deux compartiments virtuels bien défini par une interface qui se "répare" d'elle-même dans un temps caractéristique donné par la vitesse de l'écoulement laminaire.

Pour se faire, Dual utilise un système de quatre seringues (2 seringues infectantes et 2 seringues aspirantes) qui permet de créer un flux laminaire constant dans l'aquarium où évolue le poisson. Le rôle des seringues peut être inversés grâce à un système de valves pour pouvoir les remplir et choisir le produit qu'elles contiennent. On peut alors construire une expérience en réalisant plusieurs cycles de remplissage-injection. Par exemple en aspirant en premier de l'eau dans les deux seringues ce qui constituera en injection un flux laminaire d'eau homogène (contrôle) et dans un second temps un produit dans une seringue et de l'eau dans l'autre ce qui en injection permettra de créer un flux laminaire de produit et un flux laminaire d'eau créant deux compartiments virtuels dans l'aquarium.

Dual est inspiré du mouvement Do It Yourself qui permet de construire des choses sans l'aide d'expert du domaine. C'est dans ce sens que tous les composants, plans et autres fichiers CAD nécessaires à la création et au montage de Dual sont disponibles en annexe. Dual peut être construite à moindre coût sans connaissance préalable en mécanique ou en électronique. Les machines nécessaires à la réalisation des pièces peuvent être trouvé dans un FabLab et à moindre coût.

7.1.3 Construction

Overview Dual est constitué de quatre parties distinctes : une partie mécanique, une partie hydraulique, une partie électronique et enfin un logiciel informatique permettant de contrôler tout le dispositif.

Système mécanique La partie mécanique de Dual comprend un pousse seringue motorisé, un support de caméra et un aquarium isolé de l'environnement extérieur. Le pousse-seringue motorisé permet d'accorder quatre seringues figure ?? qui marche deux à deux en opposition de phase. Quand les seringues supérieures injectent, les seringues inférieures se remplissent et inversement. Cela va permettre de créer le flux laminaire nécessaire à la création

des deux compartiments virtuels. Tous les plans et fichiers nécessaires à la réalisation des différents éléments est disponible en annexe.

L'aquarium figure ?? est constitué d'une puce millifluidique en XXXX réalisée à la découpe laser et coller grâce à de l'acide acétique. Le XXXX est un matériau plastique transparent ce qui permet de conserver une bonne qualité d'image quand on filme en transmission. Elle est constituée en son centre d'une zone délimitée par deux grilles en plastique réalisées en impression 3D où le poisson pourra évoluer. La puce comprend deux entrées et deux sorties qui seront connectées par le système hydraulique aux seringues. Cela permet de créer deux flux laminaires à volume constant dans l'aquarium. Le tronçon reliant les entrées et les sorties à la zone où le poisson est confinée présente un profil évasé pour éviter toutes perturbations et un raccordement optimal entre les deux flux laminaires. Enfin, une plaque en plastique est placée au-dessus du système et permet d'éviter que le poisson ne s'échappe et les déformations optiques lors de l'enregistrement.

La puce millifluidique est placée dans une boîte réalisée avec une structure en MakerBeam et des panneaux en MDF découpé à la découpe laser. Cette boîte permet d'isoler l'expérience du milieu environnant et de placer l'animale dans un éclairage contrôlé (ou bien obscurité complète). Cette boîte contient figure ?? de bas en haut : deux LED permettant d'éclairer en lumière visible et en infrarouge (nécessaire pour l'enregistrement), un diffuseur permettant un éclairage homogène, la puce microfluidique et sur la face supérieure une fenêtre amovible constituée d'un filtre laissant uniquement passer la lumière IR, cela permet de bloquer toutes lumières extérieures tout en pouvant enregistrer les images de l'expérience en lumière IR.

Un support de caméra est placé au-dessus de la boîte et une caméra Chameleon3 FLIR Systems permet d'enregistrer les expériences en lumière IR.

Tous ces éléments sont fixés solidement et à niveau à la table pour éviter toutes vibrations ou biais qui pourraient perturber l'animale pendant l'expérience. La facilité de réalisation de la puce microfluidique en moins de 12 h permet de la remplacer facilement en cas de détérioration ou de contamination.

Système hydraulique Le système hydraulique permet de relier les seringues à la puce microfluidique avec un système de valves permettant de choisir quelle seringue injectera dans quelle entrée de la puce microfluidique,

et depuis quel récipient se remplira la seringue, un schéma du circuit est disponible figure ???. Ce système permet de créer des expériences construites en mettant bout à bout des cycles de remplissage et d'injection. Les valves sont contrôlées électroniquement ce qui permet d'automatiser complètement le système.

Système électronique Le système électronique permet de faire l'interface entre les éléments à contrôler que sont les valves, les LED et le moteur du pousse-seringue et le logiciel informatique. Le circuit imprimé a été créé sur-mesure ???. Il accueille un Arduino Nano servant de microcontrôleur, Six relais permettant le contrôle des valves à partir des signaux logiques de l'Arduino. Un contrôleur de moteur pas à pas EasyDriver permettant de contrôler la vitesse du moteur depuis une sortie logique de l'Arduino. Un système de potentiomètre permet de contrôler l'intensité des LED d'éclairage. Tout le système électronique ainsi que le moteur du pousse-seringue est alimenté par une alimentation ATX d'ordinateur 550W connectée au circuit imprimé par un montage inspiré du Benchtop Power Board de Sparkfun.

Système informatique Un logiciel a été développé spécialement pour contrôler le dispositif. L'interface graphique a été créée à l'aide de Qt et la caméra est interfacé au logiciel grâce au SDK de FLIR Systems. Le logiciel permet de contrôler manuellement chaque élément du système comme les valves, la caméra et le moteur du pousse-seringue. Il permet de contrôler l'éclairage ainsi que d'enregistrer les images provenant de la caméra. Il est possible (et conseillé) de créer des templates d'expériences, un simple fichier texte, contenant les instructions nécessaires à l'automatisation des cycles d'injection et d'aspiration. Le logiciel permet de contrôler depuis la même interface quatre Duals.

7.1.4 Utilisation

Pour nos besoins, nous avons construit quatre Duals que nous avons fait tourner en parallèle. La construction complète du montage nécessite une découpe laser, une imprimante 3D, du matériel électronique (fer à souder, etc.) et du matériel d'atelier (tournevis, vis, perceuse, etc.) et les matériaux et plan (disponible en annexe ???. Deux semaines ont été nécessaires à la construction du dispositif à partir des plans. La construction ne demande pas

de connaissance spécifique et la construction par exemple dans un FabLab permettra de trouver le matériel et l'aide nécessaire en cas de difficultés à réaliser le projet.

L'utilisation du Dual est extrêmement simple en pratique. Une fois le template d'expériences créé, la seule tâche manuelle est de placer le poisson dans la puce microfluidique, de refermer le dispositif et de faire jouer le template d'expérience. Il est nécessaire de contrôler que les récipients d'aspiration (eau où produit) sont remplis. Il est alors possible d'enchaîner les expériences avec peu de temps mort et une intervention manuelle minimale.

Un problème récurrent que nous avons rencontré est l'encrassage du circuit hydraulique. Le colorant utilisé pour visualiser l'écoulement ?? fini par boucher les valves. Une solution trouvée est de régulièrement passer les valves dans un bain ultrasonique pour les déboucher et de changer de temps en temps les tuyaux qui raccordent les valves, les seringues et la puce microfluidique. Le fait d'effectuer un protocole de rinçage qui peut être lancé automatiquement à la fin de la journée nous a aussi permis de réduire ce problème.

7.2 The Tropical River

7.2.1 Description

La perception chimique dans un environnement aquatique turbulent comme lequel sont soumis les poissons nécessite un dispositif expérimental permettant de créer de manière contrôlée des écoulements et de délivrer des stimuli chimique. Pour se faire, nous avons créé un dispositif expérimental permettant de délivrer un écoulement laminaire à température contrôlée, permettant d'imager les poissons aussi bien en lumière visible qu'en lumière infrarouge pour les placer dans le noir total. Une buse d'injection permet de créer des panaches turbulent ou laminaire à l'intérieur de cet écoulement.

Dispositif mécanique La partie mécanique du dispositif est constitué d'un canal assemblé à partir de plaque de PVC transparent. Une plaque lumineuse est placé sous le canal permettant un éclairage en lumière visible par le dessous pour éviter toutes réflexions sur la surface de l'eau. Le canal peut être éclairé par le dessus par un ensemble de LED infrarouge. Une caméra est placée au-dessus du canal et permet d'enregistrer les expériences.

Un miroir est placé à 45 degrés sur le côté du canal ce qui permet de récupérer les positions verticales des poissons sur la même image que leurs positions horizontales. L'ensemble du canal est placé dans une boîte pour l'isoler de la lumière et de toutes perturbations liées à l'environnement immédiat de l'expérience.

Dispositif hydrodynamique Le canal est alimenté en eau depuis un robinet connecté au réseau d'eau du bâtiment. Avant d'arriver dans le canal, elle est filtrée par un filtre au charbon actif, chauffé grâce à un système de bain marie puis injecté par une extrémité du canal. Un réseau de paille est placé dans le canal permettant ainsi de régulariser l'écoulement turbulent en écoulement laminaire. Le débit peut être ajusté grâce à une valve solénoïde. L'autre extrémité du canal est laissé libre et l'eau sortante est redirigé vers le réseau d'eau usé du bâtiment, les produits testés ne nécessitant pas de traitement particulier. Il est possible de diluer un produit en amont du canal grâce à une buse d'injection située directement à la sortie de l'alimentation en eau, un agitateur magnétique est placé directement dans le canal pour faciliter la dilution. Une autre buse d'injection peut être placé dans le canal pour créer un panache dans l'écoulement, il est alimenté depuis un réservoir placé en hauteur et le débit peut être ajusté par gravité.

Dispositif de contrôle Tous les composants du système sont contrôlés depuis un logiciel maison.

Température Un capteur de température est placé dans l'écoulement et envoie la température instantanée au logiciel de contrôle. Le logiciel va alors sélectionner la température du bain marie en amont de l'injection dans le canal grâce à un système de rétro action par PID. Cela permet de garder une température constante de le canal malgré des variations de température dans le système d'eau du bâtiment.

Injection L'injection de différent produit peut être sélectionné de manière automatisé grâce à un système de valves contrôlé par le logiciel. Cela permet de mettre en place facilement des protocoles d'injection. La vitesse de l'écoulement laminaire dans le canal est aussi configurable dans le logiciel et est modulé grâce au contrôleur de débit à l'entrée du dispositif.

Caméra Le réglage et l'acquisition des images par la caméra est fait directement dans le logiciel de contrôle, ce qui permet d'inclure dans les fichiers des données tel que le temps relatif à l'expérience, la température instantanée et d'autres variables.

Logiciel Le logiciel de contrôle permet de récupérer et de contrôler toutes les variables de l'expérience. Il permet de sélectionner la vitesse, la température de l'écoulement et les paramètres de capture de la caméra. Une fonction clef du logiciel est de pouvoir construire des protocoles d'expérience. Il s'agit de simple fichier texte précisant la valeur souhaitée des variables à un temps donné. Le logiciel accepte nativement d'autres capteurs Arduino répondant à un standard prédéfini permettant de facilement rajouter un ou des capteurs sans avoir à modifier le logiciel.

7.2.2 Utilisation et limitation

On voit que ce dispositif expérimental est très versatile. La capacité à moduler séparément ou simultanément la vitesse, la température de l'écoulement laminaire, ainsi que l'injection de produits dans l'écoulement de manière automatisée et quantifiée tout en filmant les poissons à la lumière ou dans le noir permet d'étudier une grande variété de comportement de rhéotaxie à la chimiotaxie en passant par la thermotaxie. La hauteur d'eau dans le canal peut être modulé par l'ajout d'une digue en sortie ce qui permet de faire évoluer dans le canal des larves ainsi que des poissons zèbres adultes sans difficulté.

Limitations L'insertion dans l'écoulement de produit chimique ne nous permet pas de réutiliser l'eau après passage dans le canal, c'est pour cela que l'apport d'eau est effectué via le réseau d'eau du bâtiment. Bien que filtré, la qualité de l'eau dépend de celle du réseau ce qui ne pose pas en principe de problème pour les poissons juvéniles et adultes qui sont élevés dans l'eau du robinet filtrée dès 2 semaines. Cela peut en revanche poser plus de problème pour les larves qui sont plus fragiles et nécessite une eau calibrée particulière (E3). Bien qu'en principe la durée de l'expérience n'a comme limite que l'espace de stockage disponible à l'enregistrement des images, on remarque au bout de quelque heure l'apparition de bulles dans le canal qui viennent perturber les poissons, l'ajout du filtre au charbon actif permet une

diminution de ce phénomène mais il reste présent.

Chapitre 8

Résultats

Bibliographie

- [1] M. S. Abreu, A. C. V. Giacomini, D. Gusso, G. Koakoski, T. A. Oliveira, A. Marqueze, R. E. Barreto, and L. J. Barcellos. Behavioral responses of zebrafish depend on the type of threatening chemical cues. *Journal of Comparative Physiology A*, 202(12) :895–901, 2016.
- [2] M. S. Abreu, A. C. V. Giacomini, D. Gusso, J. G. Rosa, G. Koakoski, F. Kalichak, R. Idalêncio, T. A. Oliveira, H. H. Barcellos, C. D. Bonan, et al. Acute exposure to waterborne psychoactive drugs attract zebrafish. *Comparative Biochemistry and Physiology Part C : Toxicology & Pharmacology*, 179 :37–43, 2016.
- [3] Airbus. *ATTOL : Autonomous Taxiing, Take-Off and Landing*, 2020. <https://www.airbus.com/newsroom/stories/autonomy-aerial-mobility.html>.
- [4] J. Atema. Structures and functions of the sense of taste in the catfish (*ictalurus natalis*). *Brain, Behavior and Evolution*, 4(4) :273–294, 1971.
- [5] Y.-X. Bai, S.-H. Zhang, Z. Fan, X.-Y. Liu, X. Zhao, X.-Z. Feng, and M.-Z. Sun. Automatic multiple zebrafish tracking based on improved hog features. *Scientific reports*, 8(1) :1–14, 2018.
- [6] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [7] D. Braida, V. Limonta, S. Pegorini, A. Zani, C. Guerini-Rocco, E. Gori, and M. Sala. Hallucinatory and rewarding effect of salvinorin a in zebrafish : κ -opioid and cb 1-cannabinoid receptor involvement. *Psychopharmacology*, 190(4) :441–448, 2007.
- [8] O. R. Braubach, A. Fine, and R. P. Croll. Distribution and functional organization of glomeruli in the olfactory bulbs of zebrafish (*danio rerio*). *Journal of comparative neurology*, 520(11) :2317–2339, 2012.

- [9] T. Darland and J. E. Dowling. Behavioral screening for cocaine sensitivity in mutagenized zebrafish. *Proceedings of the National Academy of Sciences*, 98(20) :11691–11696, 2001.
- [10] A. I. Dell, J. A. Bender, K. Branson, I. D. Couzin, G. G. de Polavieja, L. P. Noldus, A. Pérez-Escudero, P. Perona, A. D. Straw, M. Wikelski, et al. Automated image-based tracking and its application in ecology. *Trends in ecology & evolution*, 29(7) :417–428, 2014.
- [11] A. Hansen, K. Reutter, and E. Zeiske. Taste bud development in the zebrafish, *danio rerio*. *Developmental dynamics : an official publication of the American Association of Anatomists*, 223(4) :483–496, 2002.
- [12] A. Hansen and E. Zeiske. Development of the olfactory organ in the zebrafish, *brachydanio rerio*. *Journal of Comparative Neurology*, 333(2) :289–300, 1993.
- [13] T. J. Hara. *Fish chemoreception*, volume 6. Springer Science & Business Media, 2012.
- [14] A. Hussain, L. R. Saraiva, D. M. Ferrero, G. Ahuja, V. S. Krishna, S. D. Liberles, and S. I. Korsching. High-affinity olfactory receptor for the death-associated odor cadaverine. *Proceedings of the National Academy of Sciences*, 110(48) :19579–19584, 2013.
- [15] M. Ichikawa and K. Ueda. Fine structure of the olfactory epithelium in the goldfish, *carassius auratus*. *Cell and tissue research*, 183(4) :445–455, 1977.
- [16] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepcut : A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016.
- [17] X. Kedikian, M. P. Faillace, and R. Bernabeu. Behavioral and molecular analysis of nicotine-conditioned place preference in zebrafish. *PLoS One*, 8(7) :e69453, 2013.
- [18] F. Kermen, L. Darnet, C. Wiest, F. Palumbo, J. Bechert, O. Uslu, and E. Yaksi. Stimulus-specific behavioral responses of zebrafish to a large range of odors exhibit individual variability. *BMC biology*, 18(1) :1–16, 2020.
- [19] K. Kotrschal, W.-D. Krautgartner, and A. Hansen. Ontogeny of the solitary chemosensory cells in the zebrafish, *danio rerio*. *Chemical senses*, 22(2) :111–118, 1997.

- [20] S. Krishnan, A. S. Mathuru, C. Kibat, M. Rahman, C. E. Lupton, J. Stewart, A. Claridge-Chang, S.-C. Yen, and S. Jesuthasan. The right dorsal habenula limits attraction to an odor in zebrafish. *Current Biology*, 24(11) :1167–1175, 2014.
- [21] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim. Multiple object tracking : A literature review. *arXiv preprint arXiv :1409.7618*, 2014.
- [22] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge. Deeplabcut : markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9) :1281–1289, 2018.
- [23] A. Mathis and R. Warren. On the inference speed and video-compression robustness of deeplabcut. *BioRxiv*, page 457242, 2018.
- [24] P. Mathur, M. A. Berberoglu, and S. Guo. Preference for ethanol in zebrafish following a single exposure. *Behavioural brain research*, 217(1) :128–133, 2011.
- [25] P. Mathur, B. Lau, and S. Guo. Conditioned place preference behavior in zebrafish. *Nature protocols*, 6(3) :338–345, 2011.
- [26] T. Nath, A. Mathis, A. C. Chen, A. Patel, M. Bethge, and M. W. Mathis. Using deeplabcut for 3d markerless pose estimation across species and behaviors. *Nature protocols*, 14(7) :2152–2176, 2019.
- [27] J. Ninkovic and L. Bally-Cuif. The zebrafish as a model system for assessing the reinforcing properties of drugs of abuse. *Methods*, 39(3) :262–274, 2006.
- [28] J. Ninkovic, A. Folchert, Y. V. Makhankov, S. C. Neuhauss, I. Sillaber, U. Straehle, and L. Bally-Cuif. Genetic identification of ache as a positive modulator of addiction to the psychostimulant d-amphetamine in zebrafish. *Journal of neurobiology*, 66(5) :463–475, 2006.
- [29] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G. De Polavieja. idtracker : tracking individuals in a group by automatic identification of unmarked animals. *Nature methods*, 11(7) :743–748, 2014.
- [30] Z.-M. Qian, S. H. Wang, X. E. Cheng, and Y. Q. Chen. An effective and robust method for tracking multiple fish in video image based on fish head detection. *BMC bioinformatics*, 17(1) :251, 2016.

- [31] G. D. Readman, S. F. Owen, J. C. Murrell, and T. G. Knowles. Do fish perceive anaesthetics as aversive? *PLoS One*, 8(9) :e73773, 2013.
- [32] A. Rodriguez, H. Zhang, J. Klaminder, T. Brodin, and M. Andersson. Toxid : an efficient algorithm to solve occlusions when tracking multiple animals. *Scientific reports*, 7(1) :1–8, 2017.
- [33] A. Rodriguez, H. Zhang, J. Klaminder, T. Brodin, P. L. Andersson, and M. Andersson. Toxtrac : a fast and robust software for tracking organisms. *Methods in Ecology and Evolution*, 9(3) :460–464, 2018.
- [34] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. Heras, and G. G. de Polavieja. Idtracker. ai : tracking all individuals in small or large collectives of unmarked animals. *Nature methods*, 16(2) :179–182, 2019.
- [35] M. Shahin, M. A. Babar, and L. Zhu. Continuous integration, delivery and deployment : a systematic review on approaches, tools, challenges and practices. *IEEE Access*, 5 :3909–3943, 2017.
- [36] StandardCppFoundation. *Cpp Language*. <https://isocpp.org/>.
- [37] TheQtCompany. *The Qt Framework*, 2020. <https://www.qt.io/>.
- [38] V. Viswanathan and R. Hussein. Applications of image processing and real-time embedded systems in autonomous cars : a short review. *International Journal of Image Processing (IJIP)*, 11(2) :35, 2017.
- [39] N. Wakisaka, N. Miyasaka, T. Koide, M. Masuda, T. Hiraki-Kajiyama, and Y. Yoshihara. An adenosine receptor for olfaction in fish. *Current Biology*, 27(10) :1437–1447, 2017.
- [40] A. Wikström et al. Benefits and challenges of continuous integration and delivery : A case study. 2019.
- [41] J. Yáñez, Y. Souto, L. Piñeiro, M. Folgueira, and R. Anadón. Gustatory and general visceral centers and their connections in the brain of adult zebrafish : a carbocyanine dye tract-tracing study. *Journal of Comparative Neurology*, 525(2) :333–362, 2017.
- [42] A. Yilmaz, O. Javed, and M. Shah. Object tracking : A survey. *Acm computing surveys (CSUR)*, 38(4) :13–es, 2006.
- [43] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu. Object detection with deep learning : A review. *IEEE transactions on neural networks and learning systems*, 30(11) :3212–3232, 2019.