

Université de Montpellier
Faculté des Sciences
L3 informatique.

Suivi de seiches dans des videos sous-marine

Projet de programmation 2

Vaisse Ariane Beldjilali Maxime Young Brun Luis-Miguel
Combe-Ounkham Gabriel

Encadre par
Houda Hammami

2022-2023

Resume

Le suivi de seiche dans des videos sous-marine en conditions reelles est particulierement difficile, du aux mouvements de la camera, au contraste qui peut etre faible, ou encore a la colorimetrie, qui peut varier d'une video a l'autre.

Notre objectif est donc de tester plusieurs approchent afin d'explorer et de proposer les solutions les plus robustes.

Nous avons commence par apprendre ce qui ce fait de mieux dans le domaine du suivi d'objet, afin d'avoir une idee des differentes techniques ainsi que de leur efficacite.

Nous utilisons un filtre a particule comme base de notre algorithme de suivi, il utilisera plusieurs types de descripteurs qui seront compares entre eux afin de determiner le plus efficace. Leur efficacite est mesure en utilisant la methode de Pascal VOC, qui consiste a comparer les bounding box de references avec celles estimees par notre methode.

Le modele d'intelligence artificielle YOLOv7 est utilise pour faire de la detection automatique de seiche dans les premieres frames de la video.

Le suivi est ensuite ameliore grace a une etape de prediction qui permet gerer les cas d'occlusions et de deformation des seiches.

Table des matières

1	Introduction	5
1.1	Énoncé du problème	5
1.2	Motivation	5
1.3	Méthodes	6
1.4	Cahier des charges	7
1.4.1	Besoins fonctionnels	7
1.4.2	Besoins non-fonctionnels	7
1.4.3	Contraintes	7
2	Technologies	8
2.1	Etat de l'art de la detection d'objet	8
2.2	Langage de programmation	8
3	Développements Logiciel : Conception, Modélisation, Implémentation	9
3.1	Développements logiciel	9
3.2	Modules	9
3.3	Structures de données	9
3.4	Statistiques	9
4	Algorithmes et Analyse	10
4.1	Algorithmes	10
4.1.1	SURF	10
4.1.2	Distance de Bhattacharyya	10
4.1.3	Filtre à particule	10
4.2	Complexité algorithmique	10
4.2.1	SURF	10
4.2.2	Distance de Bhattacharyya	10
4.2.3	Filtre à particule	10
5	Analyse des résultats	11
5.1	Performances	11
5.2	Analyse et comparaison	11
5.3	Procédures de test	11
6	Gestion du Projet	12
6.1	Planification	12
6.2	Changements majeurs	12
7	Bilan et Conclusions	13
	Appendices	16

Table des figures

1 Introduction

1.1 Énoncé du problème

Le suivi d'objet par vision par ordinateur a toujours été, et est encore, un problème suscitant beaucoup d'intérêt et qui fait l'objet de beaucoup de sujet de recherche.

En effet, le suivi d'objet apparaît dans plein de domaines, comme par exemple :

- L'aérospatiale, avec l'arrimage de module à l'ISS, ou encore le suivi de débris spatiaux pour ensuite les faire sortir d'orbites sensibles.
- Le militaire, avec le suivi de missile pour interception précise.
- L'astrophysique, avec le suivi de corps célestes.
- L'étude de population animale, comme l'étude de cycle migratoire ou du comportement de certaines espèces.

Le projet se place dans le contexte de l'étude de population animale en milieu aquatique, et plus particulièrement, de seiches.

Il a pour objectif de fournir un outil qui permet de suivre des seiches de façon non intrusive (pas de capteurs sur la seiche suivie), afin de limiter l'interaction humaine avec les seiches.

1.2 Motivation

Le suivi d'objet, de manière générale, étant un problème que l'on rencontre dans beaucoup de domaines liés à l'informatique, les précédents exemples n'étant qu'un petit aperçu, il est donc un problème de choix à étudier lors de notre parcours d'études.

De plus, il permet d'introduire des concepts et algorithmes fondamentaux, comme le concept de filtrage de signaux, ou d'espace de représentation, ou encore, les algorithmes de filtrage, de descripteurs, ou de mesure de similarité.

Ces concepts et algorithmes sont récurrents dans le monde de l'informatique et plus précisément quand il s'agit de faire du traitement du signal ou de la vision par ordinateur.

Ce familiariser dès à présent avec ces différentes notions pourra grandement nous aider dans la suite de notre parcours.

1.3 Méthodes

Il existe beaucoup d'approches possible pour résoudre le problème de suivi d'objet, approches parmi lesquelles on peut noter :

— *Intelligence Artificielle*

Cette approche est de plus en plus populaire, notamment avec des modèles comme YOLO ou SSD. Ces modèles peuvent directement donner la bounding box de l'objet suivi, sans avoir à faire de traitement sur la sortie du modèle.

Cependant, cette approche est peu résistante à l'occlusion de l'objet suivi.

— *Capteurs*

Cette approche utilise, par exemple, des IMUs ou marqueurs infrarouge, qui peuvent donner des informations sur l'accélération linéaire ou angulaire. Ces informations sont ensuite filtrées grâce à des algorithmes de filtrage, comme le filtre de Kalman (linéaire ou non), ou encore le filtre à particule.

Cependant, cette approche nécessite de poser des capteurs sur l'objet à suivre.

— *Photogrammétrie*

Cette approche utilise des descripteurs d'image pour extraire des features importantes et ensuite, matcher ces features avec d'autres images pour pouvoir estimer le déplacement de la caméra.

Cependant, cette approche est plus utilisée dans le cas où l'on cherche à savoir où est-ce que le cameraman se situe, plutôt qu'un objet qui se trouve dans une image (comme le SLAM).

— *Hybride*

Cette approche combine différentes parties des méthodes déjà présentées et est celle sur laquelle ce projet est basé.

On utilise l'intelligence artificielle pour détecter un objet d'intérêt à suivre dans une séquence d'image, la partie filtrage de l'approche avec des capteurs pour améliorer nos estimations de l'état de l'objet suivi, et enfin, la photogrammétrie pour récupérer les features intéressantes dans une image et les comparer avec les features d'une image de référence.

Cette approche est cependant assez sensible aux paramètres que nous lui donnons, ainsi qu'à certaines caractéristiques des images données en entrée, comme le contraste, la résolution ou encore la colorimétrie.

Le détail de cette approche sera donné en partie 3.

1.4 Cahier des charges

1.4.1 Besoins fonctionnels

Les besoins peuvent etre separees en 5 categories :

1. Le besoin d'une intelligence artificielle pour effectuer la detection initiale.
2. Le besoin de descripteurs pour recuperer un vecteur qui decrit une image donnee.
3. Le besoin de mesures de similarite pour comparer un vecteur caracteristique d'une image avec une descripteur de reference.
4. Le besoin d'un filtre a particule permettant d'estimer certaines proprietes de la seiche que l'on suit.
5. Le besoin d'un programme principal permettant d'agencer chacune des parties ensemble.

Les differents descripteurs et mesures de similarite devront pouvoir etre utilise par le filtre a particule afin de mettre a jour l'etat de chacune des particules. Par extension, le filtre a particule devra etre modulable, afin de fonctionner avec ces differents descripteurs et mesures de similarite, ainsi que de repondre aux demandent du programme principale.

Le programme principale ce charge de l'initialisation des differents modules ainsi que de l'affichage de donnees clefs (visualisation de resulats).

1.4.2 Besoins non-fonctionnels

Les formats videos acceptes sont libres.

La resolution des videos est egalement libre, mais une preference sera porte pour la resolution 640x640 (resolution utilise pour entrainer l'intelligence artificielle).

Le programme doit pouvoir tourner sur Windows, Linux et OSX.

Le programme doit pouvoir sauvegarder le resultat obtenu en une video et egalement sauvegarder les bounding box dans un fichier texte.

1.4.3 Contraintes

Aucun budget n'a ete alloue pour le projet, le travail s'effectuera sur nos machines personnelles, ou sur les machines misent a disposition par l'universite.

Le projet doit etre complete en 4 mois, avec une vingtaine de jour supplementaire pour la redaction du rapport.

2 Technologies

2.1 Etat de l'art de la detection d'objet

2.2 Langage de programmation

3 Développements Logiciel : Conception, Modélisation, Implémentation

3.1 Développements logiciel

3.2 Modules

3.3 Structures de données

3.4 Statistiques

4 Algorithmes et Analyse

4.1 Algorithmes

4.1.1 SURF

4.1.2 Distance de Bhattacharyya

4.1.3 Filtre à particule

4.2 Complexité algorithmique

4.2.1 SURF

4.2.2 Distance de Bhattacharyya

4.2.3 Filtre à particule

5 Analyse des résultats

5.1 Performances

5.2 Analyse et comparaison

5.3 Procédures de test

6 Gestion du Projet

6.1 Planification

6.2 Changements majeurs

7 Bilan et Conclusions

[7] [4] [5] [3] [2] [1] [6]

Bibliographie

- [1] Roger Labbe (RLABBE). *Kalman and Bayesian Filters in Python*. <https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python>. 2014.
- [2] A. BHATTACHARYYA. « On a Measure of Divergence between Two Multinomial Populations ». In : *Sankhyā : The Indian Journal of Statistics (1933-1960)* 7.4 (1960), p. 401-406.
- [3] N. DALAL et B. TRIGGS. « Histograms of Oriented Gradients for Human Detection ». In : *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. T. 1. San Diego, CA, USA : IEEE, 2005, p. 886-893.
- [4] Xiaofang KONG et al. « Particle filter-based vehicle tracking via HOG features after image stabilisation in intelligent drive system ». In : *IET Intelligent Transport Systems* 13.6 (juin 2019), p. 942-949.
- [5] QIANG ZHU et al. « Fast Human Detection Using a Cascade of Histograms of Oriented Gradients ». In : *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*. T. 2. New York, NY, USA : IEEE, 2006, p. 1491-1498.
- [6] Chien-Yao WANG, Alexey BOCHKOVSKIY et Hong-Yuan Mark LIAO. « YOLOv7 : Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors ». In : ().
- [7] Fen XU et Ming GAO. « Human detection and tracking based on HOG and particle filter ». In : *2010 3rd International Congress on Image and Signal Processing*. Yantai, China : IEEE, oct. 2010, p. 1503-1507.

Appendices