



UNIVERSITÉ DE MONTPELLIER

RAPPORT DE STAGE DE FIN D'ÉTUDE
POUR L'OBTENTION DU MASTER 2 INFORMATIQUE
SPÉCIALITÉ IMAGINA

**Suivi de poissons dans des images
sous-marine pour la reconnaissance
et identification de thons et de
requins.**

Benjamin COMMANDRÉ

Encadré par
Marc CHAUMONT
Gérard SUBSOL

Tuteur pédagogique
Michelle JOAB



Remerciements

Je souhaiterais tout d'abord remercier l'ensemble des personnes qui ont rendu ce stage possible et qui m'ont accueilli au sein de l'équipe ICAR.

Je remercie bien sûr tout particulièrement mes tuteurs de stage, MM. Marc CHAUMONT et Gérard SUBSOL pour m'avoir accompagné et conseillé tout au long de la réalisation de ce stage ainsi que Mme Manuela CAPELLO et MM. Fabien FORGET et Laurent DAGORN du laboratoire MARBEC.

Je tiens enfin à remercier MM. Sébastien VILLON et Lionel PIBRE pour leur aide et tous les membres de l'équipe pour l'ambiance présente au sein du laboratoire.

Table des matières

Table des matières	3
Table des figures	5
Liste des Algorithmes	7
1 Introduction	9
1.1 Généralités	9
1.1.1 LIRMM	9
1.1.2 Équipe ICAR	9
1.1.3 UMR-Marbec	10
1.2 Sujet	10
2 Problématique et État de l’art	13
2.1 Problématique du suivi de cible	13
3 Descripteurs	15
3.1 Histogramme couleur et LBP*	15
3.1.1 Histogramme couleur	15
3.1.2 LBP*	15
3.2 Histogramme de gradient orienté	17
3.2.1 HOG* en cascade	19
3.2.2 Pouvoir descriptif	20
3.3 Comparaison de deux HOG	20
4 Filtre à particules	22
4.1 Principe	22
4.1.1 Mise à jour du descripteur	25
4.2 Comment établir un modèle de prédiction ?	27
4.2.1 Prédiction de la position des particules	27
4.2.2 Prédiction des échelles	28
4.2.3 Vignettes de référence	28
4.3 Amélioration par regroupement d’images	30

5 Réseau de neurones convolutif	33
5.1 Définition	33
5.2 Apprentissage supervisé	34
6 Résultats	37
6.1 Méthodologie	37
6.2 Résultats	37
7 Conclusion	41
Bibliographie	42

Table des figures

1.1	Exemples d'images extraites des vidéos utilisées.	11
1.2	Pêche à la senne	11
2.1	Différentes apparences d'un même requin dans une vidéo.	14
3.1	Description LBP*	16
3.2	Voisinage LBP*	16
3.3	Détection LBP*	16
3.4	Image originale	17
3.5	Carte des modules gradients	17
3.6	Bloc d'un HOG*	18
3.7	Déplacement d'un bloc	19
3.8	HOG* en cascade	20
3.9	Faux positif	20
3.10	Description faux positif	21
3.11	Valeur de la comparaison entre deux HOG* représentant un requin à l'aide de la distance de Bhattacharyya.	21
4.1	Initialisation des particules	23
4.2	Dispersion des particules	23
4.3	Redistribution des particules	27
4.4	Poids de la particule à partir du motif à l'instant $t-1$ et d'une vignette de référence.	30
4.5	Groupement d'image.	30
5.1	Architecture standard d'un réseau à convolutions	34
5.2	Exemple de neurone	34
5.3	Application d'un CNN*	35
5.4	Exemple d'un CNN* pour la reconnaissance de poisson en milieu corallien.	36
6.1	Écart entre la vérité terrain et le suivi retourné d'un requin de profil avec un fond uniforme.	38
6.2	Écart entre la vérité terrain et le suivi retourné d'un requin en contre-plongée avec un banc de thons en arrière-plan.	39

6.3 Écart entre la vérité terrain et le suivi retourné d'un requin de face avec d'autres cibles potentielles proches. 40

Liste des Algorithmes

1	Filtre à particule sans a priori	24
2	Algorithme de suivi basé sur une combinaison histogramme couleur et LBP*	26
3	Filtre à particule avec prédiction	29
4	Filtre à particule final	32

Glossaire

Les termes définis dans ce glossaire sont identifiables dans le corps du texte au moyen d'un astérisque ().*

HOG Un Histogram of Oriented Gradients ou Histogramme de gradient orienté est une caractéristique basée sur les différences d'intensité des gradients de l'image. Il s'agit du principal vecteur caractéristique qui sera utilisé pour le suivi.

CNN Les Convolutional Neural Network ou Réseaux de neurones convolutifs sont une méthode d'apprentissage automatique basée sur un empilement multi-couche dont l'objectif est de pré-traiter de petites quantités d'informations.

GOP Group Of Pictures ou Groupement d'images. Ensemble de plusieurs images consécutives dans une vidéo.

LBP Les Local Binary Pattern ou Motifs binaires locaux sont des caractéristiques utilisées pour reconnaître les textures ainsi que pour la détection d'objet en vision par ordinateur.

Chapitre 1

Introduction

1.1 Généralités

1.1.1 LIRMM

Le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM). Il s'agit d'une unité mixte de recherche de l'université de Montpellier et du Centre national de la recherche scientifique, département Sciences et technologies de l'information et de la communication.

Les travaux sont menés dans trois départements scientifiques de recherche, eux-mêmes organisés en "équipe-projet" :

- le laboratoire d'informatique,
- le laboratoire de micro-électronique,
- le laboratoire de robotique.

Ce stage fut effectué au sein de l'équipe ICAR du laboratoire d'informatique et fut réalisé en collaboration avec le laboratoire de l'UMR-Marbec.

1.1.2 Équipe ICAR

L'équipe ICAR développe son activité selon trois axes scientifiques associant image et interaction pour la manipulation des données visuelles tels que les images, les vidéos et les objets 3D. Ces trois axes sont : Analyse et Traitement (AT), Codage et Protection (CP), et Modélisation et Visualisation (MV).

Un ensemble de développements théoriques ou applicatifs se retrouvent dans ces trois axes thématiques. Les principaux domaines applicatifs concernent le médical, le patrimoine culturel (artistique et paléontologie), la sécurité, la robotique, l'agronomie et l'industrie manufacturière.

1.1.3 UMR-Marbec

L'Unité Mixte de Recherche (UMR) MARBEC, MARine Biodiversity, Exploitation and Conservation, a été créée le 1er janvier 2015. Ses autorités de tutelle sont l'IRD, l'Ifremer, l'Université de Montpellier et le CNRS.

MARBEC est l'un des plus importants laboratoires travaillant sur la biodiversité marine et ses usages en France avec environ 230 agents, dont 80 chercheurs et enseignants-chercheurs. L'unité est implantée à Sète, Montpellier et Palavas-les-flots, ainsi que dans l'océan Indien, en Asie, en Afrique et en Amérique du Sud.

Elle étudie la biodiversité marine des écosystèmes lagunaires, côtiers et hauturiers, principalement méditerranéens et tropicaux. Ses recherches portent sur différents niveaux d'intégration, des aspects moléculaires, individuels, populationnels et communautaires, aux usages de cette biodiversité par l'Homme.

1.2 Sujet

Ce stage fut encadré par Marc CHAUMONT et Gérard SUBSOL, membres de l'équipe ICAR, avec la collaboration de Manuella CAPELLO, Fabien FORGET et Laurent DAGORN membres du laboratoire MARBEC implanté à Sète.

L'objectif de ce stage consiste à la mise en œuvre d'un algorithme de suivi d'espèces marines et notamment des thons et des requins dans des vidéos fournies par le laboratoire Marbec, prises lors de pêche à la senne aux thons. Ces vidéos proviennent de trois séances de pêche réalisées en 2012, 2013 et 2014 dans l'océan pacifique. Au cours de ces séances, un groupe composé de deux à quatre plongeurs ont enregistré des vidéos aussi bien à l'intérieur qu'à l'extérieur de la senne. La plongée se faisant soit en apnée soit avec du matériel. Cela nous donne donc des vidéos assez hétérogènes que cela soit au niveau de la durée, de quelques dizaines et secondes à plusieurs minutes, des espèces présentes ou de la qualité de l'image. La figure 1.1 montre différentes images extraites de plusieurs vidéos.

La pêche à la senne consiste à capturer les poissons en les encerclant à l'aide d'un filet de pêche appelé senne (ou seine). La senne tournante non coulissante est un filet constitué d'une poche centrale en forme de cuillère et de deux extrémités en forme d'ailes. Les deux ailes permettent de rabattre les poissons dans la poche centrale et de remonter le filet à bord comme montré dans la figure 1.2. L'inconvénient de cette méthode est que de nombreuses espèces non désirées, appelées prises accessoire, sont également capturées et notamment les requins soyeux qui sont généralement attirés par les objets flottants se trouvant avec les agrégations de thons tropicaux.

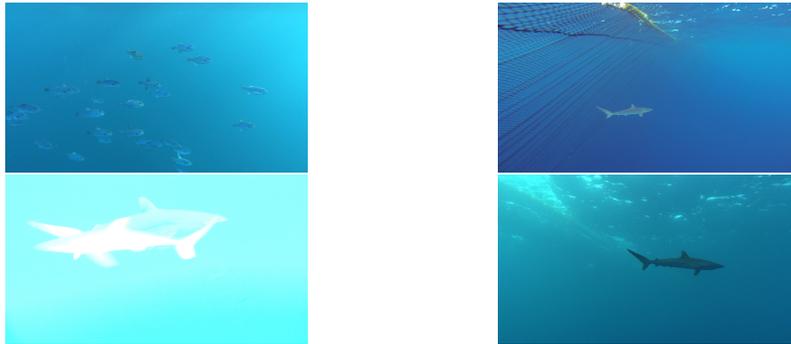


FIGURE 1.1 – Exemples d’images extraites des vidéos utilisées.

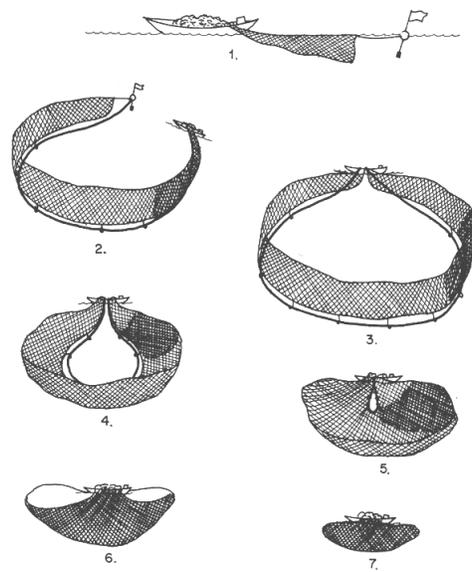


FIGURE 1.2 – Pêche à la senne.

Bien que des mesures possibles d'atténuation de prise accessoire de requins soyeux ont été envisagées, parmi lesquelles la possibilité de les attirer hors de la senne, celles-ci n'ont pas eu de résultats satisfaisant.

L'intérêt de ce stage est donc de fournir aux scientifiques une solution leur permettant d'effectuer une étude plus approfondie du comportement des requins ou d'autres espèces marines pour pouvoir obtenir des indications sur les diverses réactions des requins soyeux face à divers stimuli, comme par exemple les interactions sociales, pour guider les futurs développements des mesures d'atténuation dans la senne et ainsi à réduire le nombre de prise accessoire.

Chapitre 2

Problématique et État de l'art

2.1 Problématique du suivi de cible

Le suivi d'une cible dans une vidéo consiste à définir lors d'un instant t une cible, que cela soit manuellement ou grâce à un algorithme, puis à extraire un ensemble d'information \vec{D} de celle-ci qui servira de descripteur de référence. Nous chercherons ensuite à retrouver automatiquement cette cible dans les images suivantes c'est-à-dire trouver une boîte englobante avec un vecteur caractéristique ayant la plus forte similarité avec le descripteur de référence. Cela peut se voir comme la recherche de la distance minimale, représentant la similarité, entre le vecteur caractéristique de la cible potentielle. Nous cherchons donc dans l'image t , la boîte englobante donc le vecteur descripteur f_t est la plus proche de la caractérisation de la cible f_c . Cela peut être écrit :

$$f_t = \underset{D}{\operatorname{argmin}} f$$

dans l'image t où $D = \text{distance}(f, f_c)$.

Les caractéristiques extraites de la cible sont des informations importantes, car c'est ce qui définit le type d'information que nous allons chercher à retrouver d'une image à l'autre. Dans le cadre de ce stage, nos cibles sont des poissons et plus précisément des thons et des requins, leurs apparences peuvent varier entre deux instants de la vidéo comme le montre la figure 2.1 qui représente différentes apparences d'un même requin au cours d'une vidéo .

Les principales difficultés rencontrées sont les suivantes :

- la présence d'autres poissons pouvant faire dévier le suivi,
- les changements de l'apparence des cibles (contorsion du poisson, changement d'échelle...),
- l'occultation des cibles,
- la proximité et le croisement d'objets similaires à notre cible,
- la disparition et réapparition des cibles à l'écran,
- les mouvements de la caméra.



FIGURE 2.1 – Différentes apparences d'un même requin dans une vidéo.

Les changements d'apparence de la cible entraînent une dissimilarité importante entre le descripteur de référence et celui de la cible candidate, pouvant entraîner l'échec de l'algorithme de suivi, celui-ci choisissant des motifs sans rapports avec la cible, mais dont le vecteur caractéristique possède une ressemblance plus forte.

Pour répondre aux objectifs de ce stage, nous avons donc choisi de mettre en œuvre un algorithme basé sur le filtre à particules et pour pouvoir décrire nos cibles de la manière la plus fiable possible et ainsi avoir un suivi robuste, deux descripteurs ont été testés et seront développés dans le chapitre suivant.

Chapitre 3

Descripteurs

Un descripteur est un vecteur caractéristique c'est-à-dire un ensemble d'informations extraites de l'image et permettant de reconnaître le motif au cours de la vidéo. Il existe plusieurs types d'informations pouvant être extrait d'une image que nous pouvons diviser en deux catégories :

- les caractéristiques de bas niveau (valeur des pixels, contours, gradients...),
- les caractéristiques de haut niveau (pour un visage humain par exemple, deux yeux et un nez).

Dans le cadre de ce stage, seules les caractéristiques de bas niveaux seront utilisées c'est-à-dire que nous ne chercherons pas à créer un modèle de la cible que nous suivons mais que nous nous appuyerons que sur les valeur des pixels de l'image.

3.1 Histogramme couleur et LBP*

3.1.1 Histogramme couleur

Les histogrammes couleurs sont parmi les vecteurs caractéristiques les plus utilisés dans la littérature. On les utilise en général dans l'espace RGB, mais ils peuvent également être utilisé dans un autre espace comme par exemple l'espace HSV, celui-ci ayant l'avantage de décorrélérer l'information issue de la luminosité. Cet histogramme représente la distribution des couleurs de l'image.

3.1.2 LBP*

Les LBP* ont été utilisés pour augmenter l'efficacité du suivi basé sur les histogrammes couleurs. En effet, les LBP* permettent d'extraire des caractéristiques liées à la texture de l'image. En le couplant avec l'histogramme couleur de cette même image nous obtenons une meilleure description de celle-ci.

Le principe général est de comparer le niveau de luminance d'un pixel avec les niveaux de ses voisins comme expliqué dans l'article [5]. La figure 3.1 est

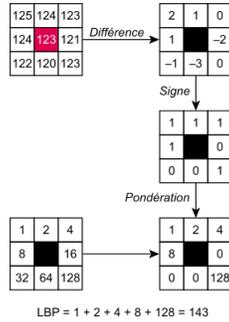


FIGURE 3.1 – Description des LBP*.

une explication concrète des LBP*. Soit un pixel $p(x,y)$, nous allons soustraire l'intensité de tous les pixels de son voisinage avec celle de p . Dans le cas où le résultat est strictement supérieur à zéro nous considérons que ce pixel a comme valeur 1. Pour finir, nous calculons la valeur du pixel dans l'image LBP*. Cela rend donc compte d'une information relative aux textures dans l'image. Il est possible d'utiliser différentes échelles du voisinage utilisé comme montré dans la figure 3.2 qui montre trois exemples de voisinage possible.

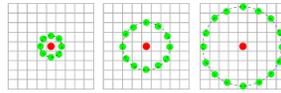


FIGURE 3.2 – Différents voisinages pour les LBP*.

Nous pouvons détecter des zones d'intérêt, tels des coins ou des bords, avec ce descripteur. Dans la figure 3.3 nous pouvons voir qu'en fonction de la répartition des pixels choisis certaines caractéristiques peuvent se démarquer.

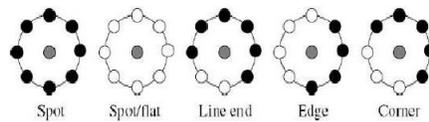


FIGURE 3.3 – Détection de zones d'intérêt avec les LBP*.

Le suivi basé sur une combinaison entre les histogrammes couleur et les LBP* est très utilisé dans la littérature. Cependant, l'environnement dans lequel nous effectuons celui-ci, c'est en dire en milieu marin réel, apporte plusieurs problèmes. En effet, il existe de grandes variations entre les vidéos et notamment



FIGURE 3.4 – Image originale



FIGURE 3.5 – Carte des modules des gradients.

en terme de luminosité dues aux différentes profondeurs auxquelles ont été prises les vidéos mais également problèmes de visibilité due à la présence de nombreux parasites, tel que des micro-algues ou du plancton, amoindrissant l'efficacité des LBP*. Tout cela fait que l'histogramme couleur de deux requins sera très différent d'une vidéo à l'autre et entraînera donc des difficultés lors du suivi.

3.2 Histogramme de gradient orienté

Les histogrammes de gradients orientés ou HOG* sont des descripteurs utilisés dans la vision par ordinateur qui ont été proposés par Navneet DALAL et Bill TRIGGS, chercheurs à l'INRIA de Grenoble [3]. Les HOG* correspondent aux variations de l'intensité du gradient dans l'image. Les HOG* définis par DALAL et TRIGGS sont construits de la manière suivante :

La première étape consiste en un pré-traitement de l'image ayant pour objectif de normaliser les couleurs et d'appliquer une correction gamma à l'image puis à la convertir en niveaux de gris. Cependant, cette étape c'est révélé être facultative la normalisation du HOG* étant suffisante.

Plusieurs espaces couleur peuvent être utilisés en notamment l'espace HSV. Cependant, nous resterons dans l'espace RGB qui permet d'obtenir de meilleurs résultats notamment en appliquant le HOG* non plus sur l'image en niveaux de gris mais, en le calculant sur chaque canal couleur de l'image nous permettant ainsi d'obtenir un meilleur vecteur représentatif de l'image.

Il s'agit ensuite de calculer la carte des gradients en x correspondant à la variation des gradients horizontalement et celle en y correspondant à la variation des gradients verticalement. Pour ce faire, nous effectuons une convolution centré avec le pixel cible. Il existe plusieurs filtres possible. Nous avons choisi les masques $[-1 \ 0 \ 1]$ pour les gradients en x et $[-1 \ 0 \ 1]^T$ pour ceux en y . Ces filtres bien que basiques restent très performant et sont donc très utilisés dans la littérature.

Avec les cartes des modules des gradients, nous allons pouvoir construire l'histogramme. Cependant, dans le but d'augmenter l'efficacité du descripteur, le HOG* ne sera par construit directement à partir de l'image.

Nous allons en effet préférer diviser l'image en sous-régions de petites tailles que nous appellerons "cellules". Celles-ci seront ensuite regroupées en "bloc". On retrouve dans la littérature plusieurs tailles différentes, que cela soit pour les cellules ou les blocs. Dans notre cas, une cellule sera définie par un ensemble de 6*6 pixels et un bloc un groupe de 3*3 cellules adjacentes.

Pour tous les blocs, nous calculerons le HOG* de chaque cellule, le HOG* du bloc correspondant à la concaténation des HOG* des cellules ce celui-ci. Pour une meilleure invariance à l'illumination et à l'ombrage, il est utile d'effectuer une normalisation du HOG* du bloc. Celle-ci se fait grâce à une norme L2 également connue sous le nom de norme euclidienne. Pour un vecteur x, nous normalisons par :

$$\sqrt{\sum_{k=0}^n (x_k)^2}$$

Le HOG* de l'image correspond donc à la concaténation de l'ensemble des HOG* calculés pour chacun de ses blocs.



FIGURE 3.6 – Bloc d'un HOG*.

Les histogrammes de chaque cellule sont construits grâce aux angles des gradients et à leurs intensités. De même que pour la taille des blocs et des cellules, il existe plusieurs structures différentes pour l'histogramme. Nous pouvons en effet modifier le nombre de classe dans l'histogramme, dans notre cas nous utiliserons neuf classes, mais il est également possible d'utiliser des angles signés, c'est-à-dire compris en $[0; 180^\circ]$ degrés ou non signés compris entre $[0; 360^\circ]$ degrés. Pour un pixel $p(x,y)$, les angles non signés sont définies, selon la formule :

$$angle_p = |\arctan(\vec{\nabla}y, \vec{\nabla}x)|$$

Cependant nous utiliserons des angles signés calculé comme suit :

$$angle_p = |\arctan(\vec{\nabla}y, \vec{\nabla}x) * 180/\pi|$$

et l'intensité suivant la formule :

$$intensite_p = \sqrt{\vec{\nabla}x^2 + \vec{\nabla}y^2}$$

L'intensité de chaque gradient sera réparti proportionnellement entre les deux classes les plus proches de l'angle trouvé. Par exemple, dans le cas où $\vec{\nabla}x = 38$ et $\vec{\nabla}y = 38$ nous obtiendrons une intensité de 53,74 et un angle de 45°. L'intensité de ce pixel sera donc distribué proportionnellement entre les deux classes les plus proche dans notre cas la classe situé en 30 recevra un quart de l'intensité et celle centrée en 50 les trois quarts restant.

Enfin pour obtenir le meilleur descripteur possible, nous effectuons un déplacement du bloc suivant :

$$\frac{T_{cellule} * T_{bloc}}{2}$$

Où $T_{cellule}$ est la taille en pixel d'une cellule et T_{bloc} est la taille d'un coté du bloc en cellule. Ceci induit un chevauchement entre les blocs comme montré dans la figure 3.7. L'entrelacement des blocs permettra d'obtenir une plus grande précision et ainsi un vecteur caractérisant au mieux a cible car un pixel aura été pris en compte plusieurs fois avec une normalisation en fonction du bloc.

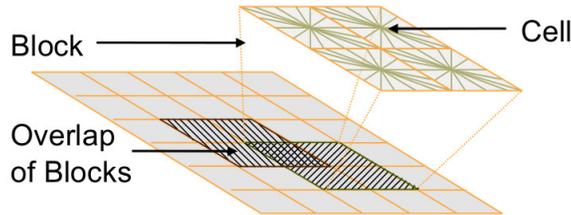


FIGURE 3.7 – Déplacement d'un bloc.

3.2.1 HOG* en cascade

L'intérêt du HOG* en cascade est d'augmenter la taille de vecteur de caractéristique du motif en ajoutant une information spatiale. Nous voulons obtenir les caractéristiques de sous-région du motif. Pour cela, nous le découpons en plusieurs sous-régions de plus en plus grandes et inclusives sur lesquelles nous calculerons le HOG* et que nous ajouterons au HOG* global de manière à obtenir le maximum de caractéristique possible comme montré dans la figure 3.8.

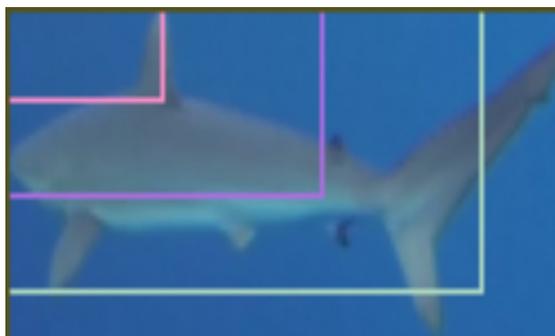


FIGURE 3.8 – HOG* en cascade.

3.2.2 Pouvoir descriptif

Bien que les HOG* soient à l'heure actuelle un des meilleurs descripteurs existant, il est cependant possible d'avoir des erreurs. Les faux positifs sont un des inconvénients majeurs du descripteur HOG*. Nous pouvons voir que dans la vision par ordinateur deux motifs étant de prime abord très différents peuvent avoir des HOG* relativement proches. Comme nous pouvons le voir avec les figures 3.9 et 3.10 provenant de l'article [9], un algorithme de détection utilisant les HOG* détectera une voiture, car le descripteur du motif sera visualisé comme une voiture. Cependant, la mise en œuvre des HOG* en cascade permet de limiter ce problème grâce à l'ajout d'une dimension spatiale.



FIGURE 3.9 – Exemple de faux positif.

3.3 Comparaison de deux HOG

Pour pouvoir comparer deux HOG*, la distance choisie fut la distance de Bhattacharyya. Cette mesure est régulièrement utilisée dans des problèmes de classification, en particulier dans le domaine de la vision par ordinateur ainsi que pour le suivi de cible, fut définie par le statisticien indien Anil Bhattacharya [2]. Il s'agit d'une mesure de la similarité entre deux distributions de probabilités

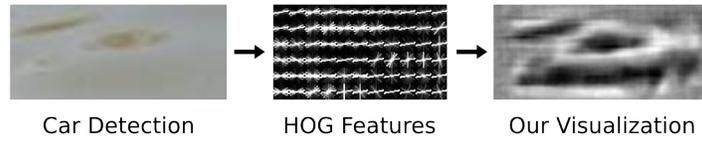


FIGURE 3.10 – Explication d’un faux positif.

discrètes bornée entre 0, dans le cas où les deux histogrammes sont totalement décorrélés et 1 dans le cas de deux vecteurs identiques. Pour deux vecteurs p et q , la distance de Bhattacharyya est définie comme :

$$D_B(p, q) = \exp(-\lambda * BC(p, q)^2) \quad (3.1)$$

où $BC(p, q)$ est le coefficient de Bhattacharyya exprimé comme :

$$BC = \sqrt{1 - \frac{1}{\sqrt{\bar{p} * N_p * \bar{q} * N_q}} * \sum_{i=0}^N \sqrt{p(i) * q(i)}}$$

Où \bar{p} et \bar{q} sont respectivement la moyenne des vecteur p et q . N la dimension des histogrammes et où λ est une constante définissant la précision de la distance. Plus λ sera élevé plus les différences entre les deux vecteurs auront un impact important.

Cette mesure est la plus utilisée pour la mise en correspondance entre deux observations basées sur l’histogramme de couleur. Bien que dans notre cas, nous n’utiliserons pas d’histogramme de couleur, mais les histogrammes de gradient.



FIGURE 3.11 – Valeur de la comparaison entre deux HOG* représentant un requin à l’aide de la distance de Bhattacharyya.

À titre d’exemple, en appliquant la distance de Bhattacharyya sur les HOG* provenant des images de la figure 3.11 nous obtenons une similarité de 16,84 %

Chapitre 4

Filtre à particules

Le filtre à particule, également connu sous le nom de bootstrap ou algorithme de condensation, est une méthode probabiliste capable de gérer des densités de probabilités multimodales c'est-à-dire des hypothèses multiples, dans le but d'améliorer la robustesse de l'algorithme. Elle permet d'approcher une distribution, impossible à estimer directement, grâce aux particules qui la représente. [1]

Cet algorithme a été utilisé en combinaison avec les descripteurs HOG* mais peut également être utilisé en combinaison avec les histogrammes couleur qui ont été le premier type de descripteur utilisé.

Le filtre particulaire s'appuie sur des éléments appelés "particules". Nous pouvons définir une particule comme le centre d'une boîte englobante c'est-à-dire un rectangle délimitant notre cible. À l'instant t , une particule est donc caractérisée par un vecteur :

$$X^t = \{x, y, l, h\}_t$$

Où $\{x, y\}$ correspond à sa position dans l'image et l et h correspondant à la largeur et la hauteur de sa boîte englobante. Nous affectons également à la particule un poids w^t qui correspond à la similarité entre le descripteur de la particule à l'instant t et le descripteur définissant la cible recherchée.

4.1 Principe

Bien qu'il existe plusieurs variantes de cet algorithme, le principe général reste le même. Il s'agit de s'appuyer sur les particules définies précédemment pour déterminer la position de la cible à l'instant t . L'intérêt de l'utilisation des particules est que celles-ci sont dispersés tout autour du centre de la cible. Nous considérons que la cible à un instant t est obligatoirement dans le voisinage de sa position à l'instant $t-1$. En effet, la cible est soumise à des contraintes physiques

et malgré le mouvement de la cible ainsi que celui de la caméra il serait illogique que la cible effectue des bonds d'une image à l'autre. Le fait de disperser les particules permet donc de localiser plus précisément notre cible.

La première étape consiste à définir la cible sur laquelle sera appliquée l'algorithme de suivi. Cela peut se faire manuellement ou grâce à un algorithme de détection. Nous récupérons donc sa position et calculons son descripteur de référence. Les particules sont ensuite initialisées au centre de la cible avant d'appliquer le filtre particulaire représenté par le point blanc au center de la figure 4.1.



FIGURE 4.1 – Initialisation des particules au centre de la cible lors de la création de celle-ci.



FIGURE 4.2 – Dispersion des particules.

Le filtre particulaire le plus basique effectue un tirage systématique des particules. C'est-à-dire qu'à chaque image t , nous effectuons une dispersion des

Algorithme 1 : Filtre à particule sans a priori

Données : Le descripteur de la cible X

Résultat : La liste des positions de la cible dans la vidéo

```
1 pour tous les Images  $t$  faire
2   si Première image alors
3     Calcul du descripteur de référence :  $D$ 
4   sinon
5     Dispersion des particules dans l'image suivant une distribution
      gaussienne de centre  $y^{t-1}$  et de variance  $V$ 
6     pour tous les Particules  $p$  faire
7       Calcul du descripteur de la particule :  $D_p^t$ 
8       Calcul du poids de la particule :
9
          
$$w_p^t = d_{BHATTACHARYYA}(D_p^t, D) \quad 3.1$$

10      fin
11      Ré-échantillonnage des particules pondérées par leur poids
12      Estimation de la position de la cible dans l'image  $t$  :
13
          
$$X^t = \frac{1}{NB_{particule}} * \sum_{p=0}^{NB_{particule}} x_p^t * w_p^t$$

14    fin
15 fin
```

particules suivant une distribution gaussienne centré sur la position de la cible dans l'image précédente et, selon une variance V telle que montré dans la figure 4.2. Le poids de chaque particule est ensuite calculé, celui-ci étant la similarité entre la particule et la cible.

La phase de ré-échantillonnage, ligne 11 de l'algorithme 1, va permettre de supprimer les particules de poids plus faible et d'augmenter l'importance celles de poids fort. Pour ce faire, nous commencerons par normaliser le poids des particules :

$$somme = \sum_{i=0}^{NB_{particule}} w_i$$
$$\forall \text{ Particule } p : w_p = \frac{w_p}{somme}$$

Nous les ordonnons ensuite de façon décroissante, selon leur poids. Chaque particule est dupliquée n fois où n est définie par la formule :

$$\forall \text{ Particule } p : n = w_p * nb_{particule}$$

Les particules de poids fort ou les particule ayant une forte similarité avec la cible de référence seront ainsi répliquées plusieurs fois tandis que celle de poids faible seront supprimées.

La position de la cible dans l'image courante est ensuite estimée en calculant le barycentre pondéré des particule en fonction de leur poids.

4.1.1 Mise à jour du descripteur

Le filtre particulaire basé sur l'algorithme 1 montre cependant vite ses limites. En effet, il n'y a, dans cette version, aucune mise à jour du descripteur décrivant la cible. C'est d'autant plus problématique dans notre cas, que l'apparence d'un poisson peut énormément varier au cours du suivi. Sans effectuer une mise à jour, notre descripteur ne correspondra plus à l'apparence de la cible entraînant de fait une perte du suivi.

Nous allons donc effectuer une mise à jour du modèle après avoir calculé la nouvelle position de la cible pour obtenir un suivi acceptable. Une fois la position de la cible estimée dans l'image t , nous remplacerons le descripteur décrivant la cible à l'instant $t-1$ par le vecteur de celle-ci dans l'image courante.

La première mise en œuvre du filtre fut faite avec une combinaison des histogrammes couleur et des LBP*, voir Algorithme 2.

Dans le cas où le nombre de particule copié est inférieur à celui défini pour le filtre, nous dupliquons la particule de poids le plus fort.

Nous pourrions ensuite estimer la position de la cible à l'instant t , en faisant une somme des particules pondérée par leur poids.

Dans le cas où le suivi a perdu la cible, que cela soit à cause d'une dérive du suivi ou d'une sortie de l'écran, nous effectuons une redistribution des particules sur la totalité de l'image telle que montré dans la figure 4.3. Après avoir calculé le poids des particules, nous estimons que la position de la cible est confondue avec la particule. Nous pouvons améliorer cette méthode en prenant en compte la dernière position correcte de la cible et en ne dispersant plus les particules dans toute l'image mais seulement dans une région proche de cette position. Cependant, cela peut entraîner des erreurs dans le cas où plusieurs cibles sont présentes dans l'image, le suivi risquant de se calquer sur une mauvaise cible.

Algorithme 2 : Algorithme de suivi basé sur une combinaison histogramme couleur et LBP*

Données : Le descripteur de la cible X

Résultat : La liste des positions de la cible dans la vidéo

```
1 pour tous les Images  $t$  faire
2   si Première image alors
3     Calcul de l'image LBP*
4     Calcul du descripteur de référence :
            $D = \text{Histogramme couleur} + \text{histogramme LBP*}$ 
5   sinon
6     Dispersion des particules dans l'image suivant une distribution
       gaussienne de centre  $y^{t-1}$  et de variance  $V$ 
7     pour tous les Particules  $p$  faire
8       Calcul du LBP* de la particule
9       Calcul de l'histogramme couleur de la particule :  $D_p^t$ 
10      Calcul du poids de la particule :
11           $w_p^t = d_{BHATTACHARYYA}(D_p^t, h^{t-1})$  3.1
12     fin
13     Ré-échantillonnage des particules pondérées par leur poids
14     Estimation de la position de la cible dans l'image  $t$  :
15          
$$X^t = \frac{1}{NB_{particule}} * \sum_{p=0}^{NB_{particule}} x_p^t * w_p^t$$

16   fin
17 fin
```



FIGURE 4.3 – Redistribution des particules.

4.2 Comment établir un modèle de prédiction ?

L'étape de prédiction n'est pas présente dans toutes les versions du filtre à particules. Nous pouvons en effet considérer le cas où nous n'avons aucun a priori c'est-à-dire que les mouvements de la cible sont trop aléatoires et donc indépendants les uns des autres. Ce cas fut envisagé, car les cibles étant des poissons leurs mouvements pouvant être très erratique notamment lorsqu'ils sont effrayés. Cependant, dans la majorité des cas, nous pouvons déterminer une trajectoire probable de la cible. À noter une spécificité de ce projet, les mouvements liés à la caméra. En effet, il ne s'agit pas d'une caméra fixe. N'ayant aucune information sur les mouvements de celle-ci, la trajectoire n'est pas celle de la cible dans son environnement, mais la trajectoire dans l'image.

Il existe différentes méthodes de prédiction. Nous utiliserons celle des différences finies d'ordre trois c'est-à-dire que nous considérerons les trois dernières positions de la particule pour déterminer sa position actuelle.

4.2.1 Prédiction de la position des particules

Le modèle de prédiction choisie est celui s'appuyant sur la méthode des différences finies basée sur la formule :

$$X(t + dt) = X(t) + X'(t).dt + X''(t) * 1/2 * dt^2 + ...$$

Avec : $X'(t)$ approximé comme $X(t) - X(t - 1)/dt$

Nous pouvons donc écrire :

$$X''(t) = X'(t) - X'(t - 1)/dt$$

D'où $X(t + dt)$ est combinaison linéaire de $X(t)$, $X(t - 1)$ et $X(t - 2)$.

Nous rajoutons enfin un bruit gaussien et nous obtenons le modèle de prédiction :

$$X(t + dt) = X(t) + X'(t).dt + X''(t) * 1/2 * dt^2 + \epsilon$$

4.2.2 Prédiction des échelles

Au cours des vidéos, il arrive que des cibles se rapproche ou s'éloigne de la caméra. Il faut donc adapter les échelles de la boîte englobante pour suivre la taille de la cible qui évolue ceci afin d'éviter d'inclure dans notre motif trop d'éléments parasites ou au contraire avoir un motif ne représentant qu'une partie de la cible et pouvant induire une erreur.

La méthode mise en œuvre consisté à appliquer une prédiction sur les échelles de chaque particule, puis à agir comme pour l'estimation de la position c'est-à-dire que les échelles de la cible étaient la moyenne de celles des particules pondérées par le poids de celles-ci.

4.2.3 Vignettes de référence

Comme nous l'avons vu précédemment, nous effectuons une mise à jour du modèle pour pallier aux modifications d'apparence de la cible. Cependant, cette opération peut poser problème. En effet dans le cas où la cible est occultée, la mise à jour du descripteur ne représentera plus sur la cible, mais sur l'obstacle l'ayant camouflé. Pour pallier ce problème, nous utiliserons des vignettes de référence. Celles-ci ont été extraite des différentes vidéos fournies et doivent représenter de manière exhaustive la cible recherchée. Ainsi au lieu de ne nous baser uniquement sur le vecteur caractéristique à l'instant $t-1$ pour déterminer le poids de la particule, nous utiliserons la formule suivante :

$$w_i^t = \alpha * w_{ref}^t + (1 - \alpha) * w_i^{t-1}$$

Où w_{ref}^t est le poids le plus élevé entre le motif à l'instant t et les vignettes de référence et w_i^{t-1} le poids en comparaison avec le descripteur de la cible à l'instant $t-1$.

De cette manière, nous assurons une meilleure stabilité de l'algorithme. En effet, en se basant sur des images de référence, le suivi aura de plus grande chance de choisir la bonne cible potentielle. De plus, en combinant comparant également avec le descripteur à l'image précédente ainsi qu'avec la trajectoire, nous évitons les confusions avec d'autres objets pouvant être confondue avec la cible. En effet, si deux motifs ayant des scores élevés avec les vignettes de référence sont proches l'un de l'autre, il est plus probable que la cible voulue ait un meilleur score avec la comparaison avec le motif en $t-1$ lui octroyant un score plus élevé.

Algorithme 3 : Filtre à particule avec prédiction

Données : Le descripteur de la cible X

Résultat : La liste des positions de la cible dans la vidéo

```
1 pour tous les Images  $t$  faire
2   si Première image alors
3     | Calcul du descripteur de référence :  $D$ 
4   sinon
5     pour tous les Particules  $p$  faire
6       | Prédiction de la position de  $p$  dans  $t$  en fonction de ses
7         | positions précédentes.
8         | Calcul du descripteur de  $p$  :  $D_p^t$ 
9         | Calcul du poids de  $p$  dans  $t$  :
          |
          | 
$$w_p^t = d_{BHATTACHARYYA}(D_p^t, D_{ref}^{t-1})$$

10        |
11      fin
12      Ré-échantillonnage des particules pondérées par leur poids
13      Estimation de la position de la cible dans l'image  $t$  :
          |
          | 
$$X^t = \frac{1}{NB_{particule}} * \sum_{p=0}^{NB_{particule}} x_p^t * w_p^t$$

14      fin
15      Mise à jour du descripteur de référence :
          |
          | 
$$D = D^t$$

15 fin
```

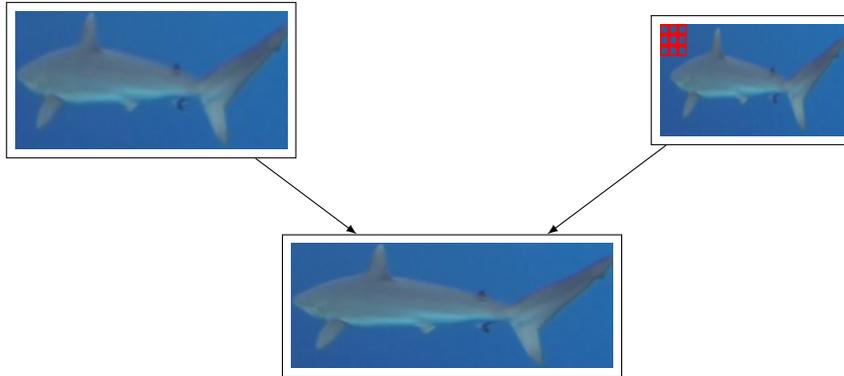


FIGURE 4.4 – Poids de la particule à partir du motif à l’instant $t-1$ et d’une vignette de référence.

4.3 Amélioration par regroupement d’images

Dans la version naïve de l’algorithme de condensation nous traitons image par image jusqu’à la fin de la vidéo. Cependant, l’utilisation de groupement d’images ou GOP* permet d’avoir un algorithme plus robuste. Le GOP* consiste à couper la vidéo en plusieurs ensemble d’images successives. Nous effectuons ensuite un traitement non plus sur l’ensemble de la vidéo, mais sur les GOP* successifs. Pour cela nous effectuons le suivi de différentes manières sur le même GOP* de manière à soit affiner le suivi soit pour corriger les éventuelles erreurs.

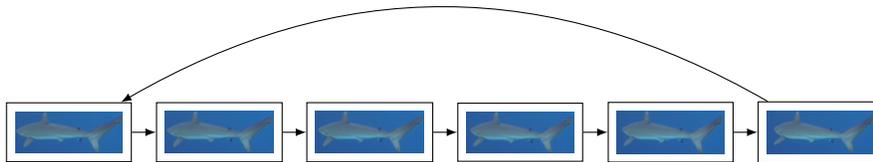


FIGURE 4.5 – Groupement d’image.

La première passe sera un suivi classique dans le but d’obtenir une estimation de la trajectoire de la cible. Lors du deuxième passage, nous affinerons la trajectoire obtenue en effectuant une seconde prédiction sur la position des particules. Nous calculerons ensuite la moyenne de ces deux trajectoires pondérées par leur poids.

Le troisième passage est un suivi sans a priori. À chaque image t , nous effectuons une dispersion des particules sans prise en compte des antécédents de

la cible pour tenter de corriger les erreurs dues à un changement de trajectoire non pris en compte.

Dans le cas d'une perte de la cible, si le poids descend sous un certain seuil, nous reprendrons une phase de suivi à partir de la dernière position ayant un poids jugé satisfaisant.

Algorithme 4 : Filtre à particule final

Données : Le descripteur de la cible X

Résultat : La liste des positions de la cible dans la vidéo

```
1 pour tous les Images  $t$  faire
2   si Première image alors
3     Calcul du descripteur de référence :  $D$ 
4   sinon
5     pour tous les Particules  $p$  faire
6       Prédiction de la position de  $p$  dans  $t$  en fonction de ses
7       positions précédentes.
8       Calcul du descripteur de  $p$  :  $D_p^t$ 
9       Calcul du poids de  $p$  dans  $t$  :
10
11         
$$w_p^t = d_{BHATTACHARYYA}(D_p^t, D_p^{t-1})$$

12
13       Calcul du poids de la particule en fonction des images de
14       référence :
15
16         
$$wr_p^t = \max(d_{BHATTACHARYYA}(D_p^t, ref))$$

17
18       Calcul du poids final de la particule :
19
20         
$$W_p^t = \alpha * wr_p^t + (1 - \alpha) * w_p^t$$

21
22     fin
23     Ré-échantillonnage des particules pondérées par leur poids
24     Estimation de la position de la cible dans l'image  $t$  :
25
26         
$$X^t = \frac{1}{NB_{particule}} * \sum_{p=0}^{NB_{particule}} x_p^t * W_p^t$$

27
28     Correction de l'estimation
29   fin
30   Mise a jour du modèle de référence
31 fin
```

Chapitre 5

Réseau de neurones convolutif

Les réseaux de neurones convolutifs sont utilisés dans le cadre de l'apprentissage automatique. Les neurones sont arrangés de sorte à ce qu'ils correspondent à des régions qui se chevauchent lors du pavage du champ visuel. Les CNN* consistent en un empilage multicouche de "perceptrons", dont le but est de pré-traiter de petites quantités d'informations. Les réseaux neuronaux convolutifs ont de larges applications dans les systèmes de recommandation, le traitement du langage naturel et le cas intéressant dans le cadre de ce stage la reconnaissance d'image. L'objectif était de coupler les CNN* avec le filtre particulier dans le but d'augmenter son efficacité. Nous nous appuyons sur l'article [6] qui traite de la reconnaissance de visage à l'aide des CNN*.

5.1 Définition

Les principales applications des réseaux de neurones sont l'optimisation et l'apprentissage. Il existe plusieurs types d'apprentissage utilisant les réseaux de neurones :

- l'apprentissage supervisé,
- l'apprentissage non supervisé,
- l'apprentissage par renforcement.

Dans le cadre de ce stage, le plus intéressant est le réseau à apprentissage supervisé, du fait que nous l'entraînons à détecter des motifs voulus. Nous nous concentrerons donc uniquement sur celui-ci.

Un réseau de neurones est formé d'un ensemble d'éléments de base interconnectés. Il existe plusieurs variantes qui sont définies selon le choix de la base, de l'architecture du réseau et ou la dynamique de celui-ci. Les éléments de base appelés des "perceptrons" ou "neurones".

La figure 5.2 est un exemple de neurone. Il reçoit les entrées définies comme $\{x_1, \dots, x_i, \dots, x_n\}$. Le potentiel d'activation du neurone p est calculé comme la

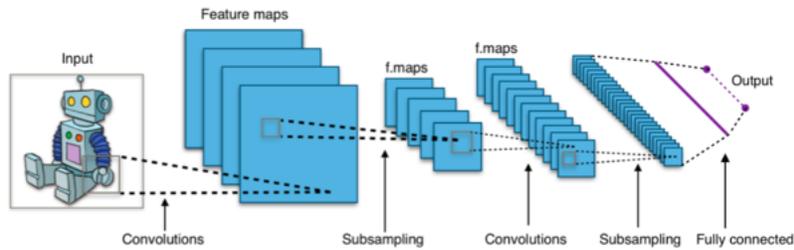


FIGURE 5.1 – Architecture standard d'un réseau à convolutions.

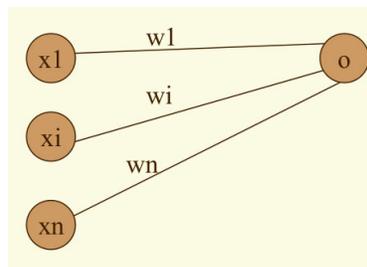


FIGURE 5.2 – Exemple de neurone.

somme pondérée par les coefficients synaptiques w_i des entrées. La sortie o est alors calculée en fonction du seuil θ . Soit $p = x.w = x_1.w_1 + \dots + x_i.w_i + \dots + x_n.w_n$. Alors : $o = 1$ si $p < 0$ et $o = 0$ si $p \geq 0$

5.2 Apprentissage supervisé

Dans le cadre de l'apprentissage supervisé, déterminer un réseau de neurones revient à chercher les coefficients synaptiques défini précédemment par w_i . Nous parlons alors de phase d'apprentissage. Nous allons modifier les caractéristiques du réseau jusqu'à que le comportement désiré soit obtenu. Pour obtenir un entraînement intéressant, nous devons pouvoir nous appuyer sur une base d'apprentissage suffisamment conséquente. Lors apprentissage supervisé, les coefficients synaptiques sont évalués en minimisant l'erreur (entre la sortie souhaitée et celle obtenue) sur une base d'apprentissage. Les coefficients synaptiques sont déterminés par rapport à des critères de conformité ou des spécifications générales.

Une base d'apprentissage est un ensemble d'exemples représentatifs du comportement ou de la fonction à modéliser. Ces exemples sont sous la forme de couples (entrée ; sortie) connus.

Lorsque la phase d'apprentissage du réseau est terminée, nous l'appliquons sur une base d'essai différente de la base d'apprentissage. Celle-ci est un ensemble d'entrée dont nous allons calculer la sortie à l'aide du réseau. Nous pouvons alors évaluer la performance de celui-ci.

La figure 5.3 est un exemple d'application d'un CNN*. Nous cherchons dans cette figure à détecter l'objet dans l'image donnée en entrée. Cette image passera à travers plusieurs couches définies ci-dessous :

- Les couches CONV calculerons le produit scalaire entre leurs poids et une partie de l'image,
- Les couches RELU appliquerons des opérations rationnelle telles que max, seuillage sur la sortie des couches CONV,
- Les couches POOL, vont effectuer une opération de sous-échantillonnage le long des dimensions spatiales (largeur, hauteur).

Au final nous obtiendrons en sortie la probabilité que l'image appartienne à telle ou telle classe.

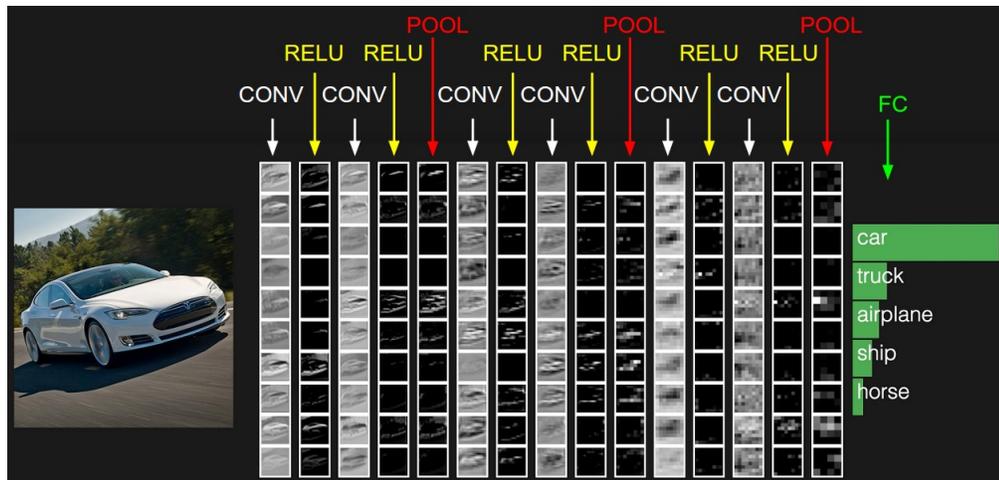


FIGURE 5.3 – Application d'un CNN*.

Pour un exemple plus en lien avec ce stage, la figure 5.4 représente la phase d'apprentissage d'un réseau de neurone développé par Sébastien VILLON, ingénieur d'étude dans l'équipe ICAR et travaillant avec une équipe du laboratoire MARBEC, située à Montpellier sur la reconnaissance de poisson en milieu corallien.

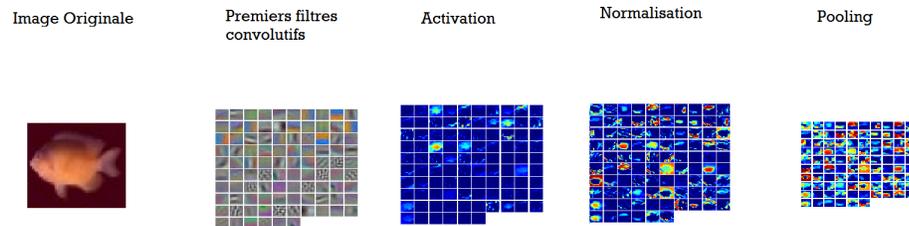


FIGURE 5.4 – Exemple d'un CNN* pour la reconnaissance de poisson en milieu corallien.

Dans cet exemple, le réseau prend en entrée l'image d'un poisson et retourne en sortie l'espèce la plus probable à laquelle il appartient. En amont, le réseau a été entraîné à identifier les caractéristiques de plusieurs poissons dont l'espèce est connue. À la fin de la phase d'entraînement, le réseau pourra reconnaître un poisson, à condition que celui-ci appartienne à l'une des espèces sur lesquels le réseau ait été entraîné, en se basant sur les caractéristiques qui ont été définies.

Bien que des tests aient été effectués dans le but de combiner les CNN* à l'algorithme développé, la base d'apprentissage trop faible fit que l'entraînement du réseau fut insuffisant pour pouvoir reconnaître les cibles. L'algorithme final n'utilise donc pas l'apprentissage supervisé.

Chapitre 6

Résultats

6.1 Méthodologie

La vérification de la qualité du suivi fut faite sur trois vidéos n'ayant pas servies lors des tests du filtre. Cependant, pour pouvoir avoir une métrique définissant la qualité de notre suivi, une étape manuelle fut nécessaire. En effet, nous ne disposons pas d'information sur la position des requins. Il a donc fallu, effectuer un suivi manuel sur les futures cibles afin de pouvoir comparer la position retourner lors du suivi.

Pour quantifier la qualité du suivi, nous allons nous appuyer sur l'aire de la position retournée par le suivi et celles définies lors de la vérité terrain. Nous comparerons à la fois leur position dans l'image et la taille des boites englobantes de la manière suivante :

Soit P_v et P_t respectivement la position définie lors de la vérité terrain et celle trouvée par le suivi. Nous utiliserons l' $Aire(P_v)$ et l' $Aire(P_t)$ obtenir une mesure quantitative du suivi de la manière suivante :

$$m = \frac{Aire(P_v) \cap Aire(P_t)}{Aire(P_v) \cup Aire(P_t)}$$

Où m , qui représente la correspondance entre les deux aires, est compris entre $[0 ; 1]$, 0 si les deux aires sont totalement disjointes et 1 dans le cas où elles sont confondues.

6.2 Résultats

Les résultats suivants ont donc été obtenu en comparant le suivi obtenu par l'algorithme à celui défini manuellement grâce à la métrique définie précédemment sur les trois vidéos sur lesquelles nous n'avons jamais appliqué l'algorithme.

À noter que les résultats ne sont pas absolus. En effet ils dépendent également de la vérité terrain qui peut contenir des imprécisions. Nous obtenons donc les résultats suivants :

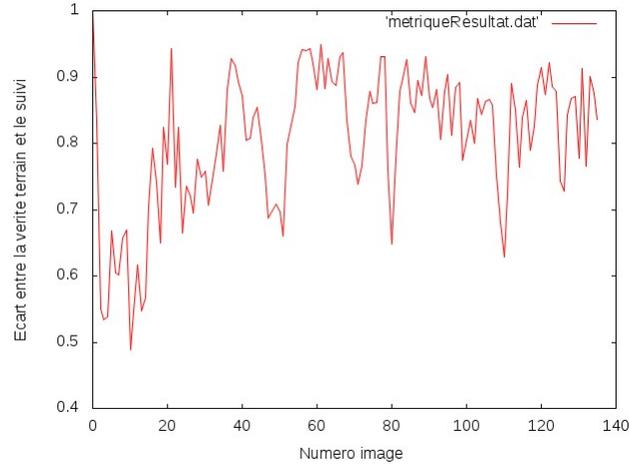


FIGURE 6.1 – Écart entre la vérité terrain et le suivi retourné d’un requin de profil avec un fond uniforme.

Le premier suivi, représenté par le résultat 6.1, est réalisé sur une vidéo propre ayant peu de parasite, avec un fond relativement homogène excepté quelque instant comprenant le filet. Nous obtenons lors de ce suivi un indice de correspondance moyenne d’environ 80,45 % par rapport à notre vérité terrain. L’indice médian étant à 83,43 %. Ce test sert à valider le filtre à particule dans lors d’un cas présentant des conditions optimales. La vidéo est nette et avec des mouvements de caméra limités. Le suivi reste centré sur la cible les raisons de la différence entre les

Le deuxième suivi, résultat 6.2, est réalisé sur une vidéo propre, mais avec un banc de thons situé à l’arrière-plan. De plus, la cible est prise en contre-plongée nous avons donc également la surface ainsi que les reflets de la lumière qui peuvent interférer lors du calcul du HOG*. Les résultats obtenus sont en moyenne de 79,55 % avec une médiane de 82,92 %. Ce test permet de tester la robustesse de l’algorithme en présence d’éléments parasites, dans ce cas un banc de thons et les reflets de la surface. Des trois vidéos utilisées pour tester le filtre mis en œuvre, nous pouvons remarquer que cette vidéo est celle dont le taux chute le plus fortement, le taux minimal étant de 40,60 % pour celle-ci alors que la première et troisième vidéos ont respectivement pour taux minimal 48,93 % et 58,49 %. Les éléments parasites en trop grand nombre peuvent donc impacter la qualité du suivi.

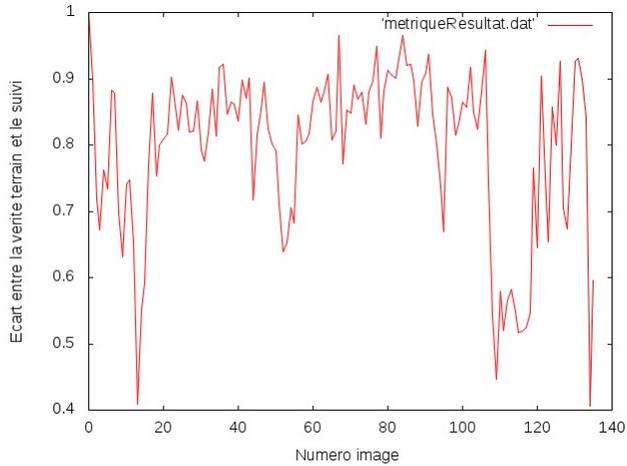


FIGURE 6.2 – Écart entre la vérité terrain et le suivi retourné d’un requin en contre-plongée avec un banc de thons en arrière-plan.

Le dernier suivi, résultat 6.3, possède cette fois plusieurs thons en premier plan. La cible est cette fois de face et deux autres requins sont également proches de notre cible avec un croisement de la trajectoire pour l’un d’eux. Le taux de correspondance du suivi, par rapport à la vérité terrain, est en moyenne de 80.86 %. Le rapport médian s’établissant à 81.89 %. Bien que d’autres cibles potentielles soient présentes et proches de notre cible, le suivi reste centré sur celle que nous avons définie, validant l’utilisant de la trajectoire pour augmenter la robustesse du filtre particulaire.

Nous pouvons donc voir que l’algorithme mis en œuvre a été efficace lors de nos tests. L’écart entre le résultat retourné et celui de la vérité terrain dans les trois vidéos, est en moyenne aux alentours de 80 %. Cependant, celle-ci étant faites manuellement et étant également imprécise, ces résultats peuvent être jugés satisfaisant. La principale différence entre la vérité terrain et le suivi est, plus que la position exacte du centre de la cible, les échelles des boîtes englobantes. En effet, celles définies lors de la vérité terrain, bien que soumises à une imprécision, sont théoriquement les boîtes optimales. Celles retournées par le suivi ont donc une marge de manœuvre pour être améliorées.

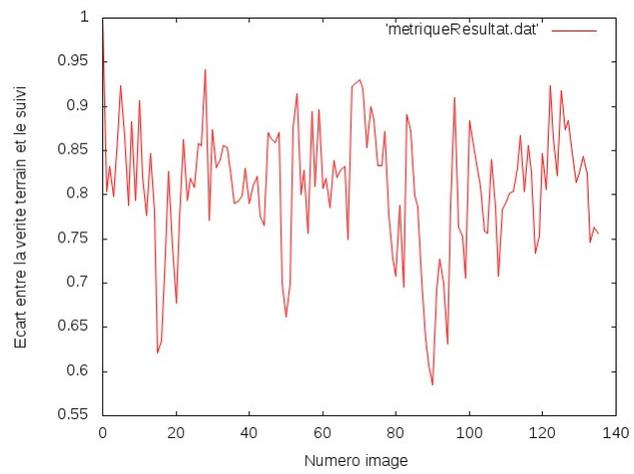


FIGURE 6.3 – Écart entre la vérité terrain et le suivi retourné d'un requin de face avec d'autres cibles potentielles proches.

Chapitre 7

Conclusion

Au terme du stage, nous avons disposons d'un logiciel de suivi multi-cible utilisant le descripteur HOG*. En effet, le suivi obtenu lors des phases de tests tend à être relativement proche de celui définie lors de la vérité terrain, l'écart moyen lors des tests s'élevant à 80,29 %. Cependant, notre méthode reste soumise à différents paramètres tels que la qualité des vidéos ou encore le nombre et la représentativité des vignettes de référence. En effet, les vidéos fortement bruité ou avec des mouvements très brusques de la caméra mettent l'algorithme en difficulté. De même dans le cas d'un nombre de vignette insuffisant ou non représentatif. Dans les autres cas, nous arrivons à obtenir un suivi acceptable même dans les cas détectés comme problématique lors du début de ce stage comme par exemple le croisement de cibles similaires ou encore les mouvements liés à la caméra.

Les perspectives d'amélioration tournent principalement autour des réseaux de neurones. Il serait d'expérimenter un réseau entraîné à la détection des espèces sur lesquelles nous voulons appliquer le suivi, soit pour remplacer totalement le descripteur HOG* soit en combinaison de celui-ci pour améliorer la qualité du suivi et notamment optimiser les échelles trouvées.

Nous pourrions également envisager d'effectuer le suivi en trois dimensions si nous possédons les vidéos adéquates tels que des vidéos stéréoscopiques. Cela permettant d'obtenir des informations complémentaires sur la cible telle que la taille des cibles ou sa distance par rapport à des points d'intérêts.

Bibliographie

- [1] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, VOL. 50, NO. 2, February 2002.
- [2] A. Bhattachary. On a measure of divergence between two multinomial populations. *The Indian Journal of Statistics*, pages 401–406, July 1946.
- [3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2005.
- [4] Robert B. Fisher, Yun-Heh Chen-Burger, Daniela Giordano, Lynda Hardman, and Fang-Pang Lin. *Fish4Knowledge : Collecting and Analyzing Massive Coral Reef Fish Video Data*. Springer, March 2016.
- [5] Kushsairy Kadir, Mohd Khairi Kamaruddin, Haidawati Nasir, Sairul I Saifie, and Zulkifli Abdul Kadir Bakti. A comparative study between lbp and haar-like features for face detection using opencv. *Engineering Technology and Technopreneuship (ICE2T)*, August 2014.
- [6] S. Lawrence, C. Lee Giles, Ah Chung Tsoi, and Andrew D. Back. Face recognition : a convolutional neural-network approach. *The Indian Journal of Statistics*, pages 98 – 113, January 1997.
- [7] Xiongbiao Luo, Ying Wan, Xiangjian He, Jie Yang, and Kensaku Mori. Diversity-enhanced condensation algorithm and its application for robust and accurate endoscope three-dimensional motion tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, June 2014.
- [8] Pablo Augusto Negri, Lionel Prévost, and Xavier Clady. Cascade de classifieurs génératifs et discriminants pour la détection de véhicules. *Université Pierre et Marie Curie*.
- [9] Carl Vondrick, Aditya Khosla, Hamed Pirsiavash, Tomasz Malisiewicz, and Antonio Torralba. Visualizing object detection features. *IEEE International Conference on Computer Vision*, December 2013.
- [10] Qiang Zhu, Shai Avidan, Mei-Chen Yeh, and Kwang-Ting Cheng. Fast human detection using a cascade of histograms of oriented gradients. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, June 2006.