UNIVERSITÉ LIBRE DE BRUXELLES
FACULTÉ DES SCIENCES
DÉPARTEMENT D'INFORMATIQUE

Analyse vidéo pour la détection, suivi et reconnaissance de poissons

DELLA MONICA Simon, OOMS Aurélien, SONNET Jean-Baptiste

Superviseur : Yann-Aël Le Borgne

Année académique 2012 - 2013

Table des matières

Chapitre 1

Introduction

1.1 Motivation

L'analyse logicielle vient de plus en plus appuyer le travail de ceux pour qui la récolte d'informations provient essentiellement d'une source numérique, i.e des vidéos enregistrées lors d'observations menées par des biologistes ou statisticiens, etc.
L'usage de procédés automatisés permet le filtrage et l'analyse massive de données qui était jusqu'alors une tâche fastidieuse à traiter. Données qui une fois traitées permettent d'inférer ou d'appuyer des thèses selon les besoins.
Par exemple, l'analyse des déplacements individuels au sein d'un banc de poissons filmé pendant un longue période est une tâche pour laquelle le passage par un traitement informatisé apparaît comme nécessaire.

Le but de ce projet est d'ébaucher une solution logicielle d'analyse vidéo permettant l'automatisation du travail de détection et de suivi de poissons dans un milieu donné.

1.2 Approche et solution

Dans le cadre de ce projet, l'algorithme *condensation* a été implémenté comme solution pour le suivi de cibles.
La réalisation de l'application comprenait plusieurs étapes :

1. Segmentation
2. Détection de traits d'intérêts
3. Observations
4. Prédictions
5. Évaluation et Mise en correspondance

Chacune de ces étapes a nécessité la compréhension de ses enjeux et des différentes réponses qui y ont été apportés. Suite à quoi, une implémentation a été choisie pour chacune d'elles. La concaténation de ces étapes a abouti en une implémentation de l'algorithme *condensation*.

Dans la recherche de solutions, d'autres approches ont été explorées, particulièrement l'algorithme SURF. Ce dernier sera exposé dans la section éponyme.

Chapitre 2

État de l'art

Il existe un grand nombre d'approche pour traiter la détection et le suivi d'objet au travers de flux d'images.
Il est nécessaire de prendre connaissance de ces différentes approches et leur évolution, ainsi que des bénéfices ou contraintes qu'elles induisent. D'autre part le sujet étant prolixe, nous limiterons ici l'exposé des méthodes de détection et de suivi à celles que nous avons utilisée ou celles qui nous sont, à un moment ou l'autre, apparues importantes pour traiter de la problématique qui nous occupe.

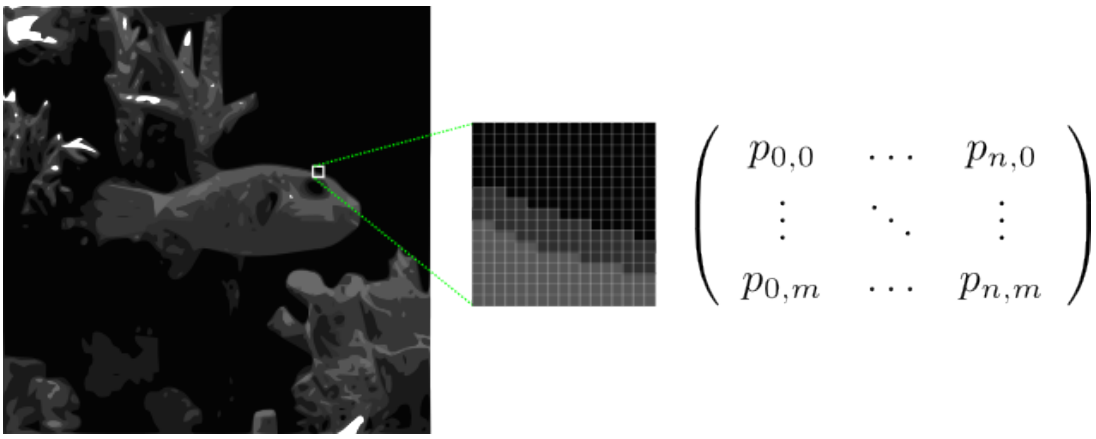
2.1 Contexte, matériaux et questions

Une vidéo est un flux, une succession d'images, dites *frames*. Le flux est dépendant de la fréquence de lecture/d'affichage, soit un taux en fps (*frames per second*).
Suivre une cible au long d'une séquence d'images, c'est définir et reconnaître une partie de l'image courante comme identique à la précédente, tout en lui autorisant des modifications (taille, position, couleur,...). Rétroactivement, c'est aussi reconstruire la trajectoire d'une cible dans la séquence d'images, cette approche se montrera pertinente pour la suite.

Le *video tracking* ou suivi de cibles au sein d'une vidéo est le suivi de morceaux ciblés d'images au travers du flux duquel elles proviennent. C'est la tâche automatisée qui consiste en la localisation, d'images en images, d'un groupe de pixels généralement en mouvement.

Si une cible apparait à l'oeil humain comme consistante alors qu'elle se meut dans son champs de vision, il en va tout autrement en *computer vision*. En effet, à tout instant la consistance conférant son unité à la cible doit être déterminée, calculée et éprouvée pour qu'elle soit reconnue.
Les parties traquées sont en fait chacune une quantité donnée de pixels, dans une espace restreint. Quantité à laquelle on concédera une identité en lui définissant/reconnaissant des attributs propres (variants et invariants). La cible sera alors contenue dans une *région d'intérêt*, c'est-à-dire une sous-matrice d'une matrice plus grande que représente l'image toute entière.

On perçoit plusieurs difficultés qu'il conviendra de maîtriser : l'unité de la cible et sa différenciation du fond, l'identité de celle-ci, malgré des modifications intrinsèques dans le flux d'images, la caractérisation et détermination du mouvement de la cible.



$$\begin{pmatrix} p_{0,0} & \cdots & p_{n,0} \\ \vdots & \ddots & \vdots \\ p_{0,m} & \cdots & p_{n,m} \end{pmatrix}$$

FIGURE 2.1 – Matrice de pixels

Deux problématiques fortement liées peuvent être dégagées [1] : le *mouvement* et la *correspondance*.

2.1.1 La question du *mouvement*

La question du mouvement comporte deux aspects : la détermination d'une cible comme mouvante et la caractérisation de ce mouvement.

Une cible mouvante

L'approche qui semble la plus évidente pour cerner une cible en mouvement est sa différenciation avec ce qui apparait comme statique.
On supposera les objets d'intérêt et mouvant, comme appartenant à l'avant-plan (*foreground*) et le reste, statique, comme appartenant au fond (*background*). On parlera alors de *soustraction du fond*.

$$F(\lambda_t) = \lambda_t - B_t$$

où,
λ_t est une frame à l'instant t
Le *foreground* : $F(\lambda_t)$
La *background*, la matrice B_t

La soustraction du fond peut se réaliser de différentes manières, mais elle doit pouvoir résister aux changements de luminosité, aux bruits et aux différentes variations de vitesse. De façon générale, l'extraction s'obtient en soustrayant de l'image courante une image de "référence", le fond. L'image de référence est actualisé à chaque étape, par accumulation pondérée et relativement à l'avant-plan extrait :

$$\texttt{dst}(x,y) \leftarrow (1-\alpha) \cdot \texttt{dst}(x,y) + \alpha \cdot \texttt{src}(x,y)$$
$$\texttt{if} \quad \texttt{mask}(x,y) \neq 0$$

où,

1. Video Tracking : A Concise Survey, E. Trucco, K. Plakas, IEEE Journal of Oceanic Engineering, Avril 2006

α est le *facteur d'oubli*
$\texttt{dst}(x, y)$ est le pixels de destination et $\texttt{src}(x, y)$ celui de l'image source.
$\texttt{mask}(x, y)$ est une image binaire générée à partir de l'avant-plan extrait en $t - 1$.

Un mouvement déterminé

Le mouvement est caractérisé par une position d'origine (vecteur position), une direction (orientation vers un point de destination relativement a un référentiel) et une vitesse (obtenu en dérivant les coordonnées par rapport au temps). L'enjeu est de parvenir déterminer que la position d'une cible à un instant t est la résultante d'un mouvement, de cette même cible, initié en $t - 1$.

Plusieurs approche sont envisageables, notamment la définition du mouvement d'image de J.Shi et C.Tomasi .
Le *mouvement d'image* :

$$I(x, y, t + \tau) = I(x - \xi(x, y, t, \tau), y - \eta(x, y, t, \tau))$$

où

$I_{t+\tau}$ est une image obtenue à partir d'un déplacement des points à l'instant t.
Le vecteur de *déplacement* $\delta = (\xi, \eta)$ est la quantité de mouvement. On perçoit l'insuffisance de la définition du déplacement δ quant aux multiples déplacements internes à la cible.

Le déplacement comme *champs de mouvement affine* :

$$\delta = D\texttt{x} + \texttt{d}$$

où D est la matrice de déformation et d une translation sont donné comme :

$$D = \begin{pmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{pmatrix}$$

Ce qui permet de poser pour un point x, centre d'une région d'intérêt de l'image J_t son mouvement équivaudra au point $Ax + \texttt{d}$ de l'image suivante $J_{t+\tau}$, où $A = I + \texttt{D}$ et I est la matrice identité donnée par :

$$J_{t+\tau}(Ax + \texttt{d}) = J_t(x)$$

La prédiction du mouvement sera donnée par l'estimation des paramètres de la matrice D et du vecteur de déplacement $\texttt{d}$.

Un autre approche plus courante est celle s'appuyant sur le concept de flux optique dont l'introduction est attribuée au psychologue James J. Gibson [2] .
Le flux optique (*Optical Flow*) est décrit comme le modèle sous-jacent au mouvement visible d'un objet dans son contexte et relativement à l'observateur. Ce mouvement peut être estimé à partir d'une séquence d'images comme une suite de vitesses instantanées ou de déplacements discrets d'images.

2. http://fr.wikipedia.org/wiki/Flux_optique

2.1.2 La question de la *correspondance*

L'objectif est de reconnaître et identifier une cible d'une frame à l'autre.

Critères d'intérêt

Pour suivre une cible, on la compare au travers du flux d'images selon certaines *unités de mesure*. Ces unités de mesure sont façonnées au regard de la problématique traitée. Elles peuvent être complexes et sont généralement hautement dépendantes des paramètres qui les composent.
Une unité de mesure est un ensemble des caractéristiques paramétrant l'objet cible, telles la position du centre de masse, l'aire, les coins, les contours ou encore l'historique des mouvements antérieurs.

La paramétrisation de la cible est capitale, car elle doit offrir des garanties sur l'identité de l'objet traqué. D'une image à l'autre, on doit pouvoir s'appuyer sur des critères robustes à l'évolution de la cible dans le flux d'image.
Il faut trouver des caractéristiques présentant des propriétés locales remarquables, c'est-à-dire des traits d'intérêts, stables ou *invariants*, on parlera de *features* de la cible.
Il existe diverses méthodes de détection de zones d'intérêts, chacune relative aux types de zones d'intérêts sur lesquels on souhaite baser l'analyse de la cible. Citons parmi les plus connu, l'algorithme de détection de coins de C. Harris et M. Stephens ou encore celui de J. Shi et C. Tomasi sur l'estimation qualitative des traits d'intérêts, mais aussi l'algorithme de détection de contours, *Canny edge detection*, présenté par l'australien J. Canny en 1986.

Fonction de mérite

Une fois la caractérisation choisie, il faut établir une méthode de comparaison débouchant sur un coefficient de qualité sanctionnant l'estimation ou la prédiction de correspondance.

Une fonction de mérite, *figure-of-merit*, est une fonction qui mesure la concordance entre les données et le *modèle*, tout en considérant un choix particulier de paramètres.

En statistique fréquentielle, la fonction de mérite est généralement agencée de sorte que de petites valeurs obtenues représentent une concordance étroite. Tandis qu'une approche bayésienne choisirait une fonction de mérite de sorte à ce que des valeurs élevées représentent une meilleure concordance [Press et al., 2007].
La fonction de mérite devra être telle qu'elle offre la meilleur façon de trouver l'extremum désiré en fonction des caractéristiques prédéfinies de la cible.

Elle devra aussi considérer que les données récoltées sont généralement bruitées. Du fait que l'objet cible se modifie tout au long du flux d'images, la reconnaissance des critères comme correspondant comprend aussi leur différenciation du contexte/bruit.
Par exemple, l'environnement où évolue la cible pourrait présenter l'une ou l'autre parcelle d'images assez ressemblante que pour passer comme identique au regard de l'unité de mesure définie. De façon générale, pour outrepasser la pollution issue d'éléments parasites valides, il sera souvent nécessaire de mettre en œuvre plusieurs approches.

2.1.3 Luma, luminance et niveau de gris

La plupart des algorithmes de traitement d'images fonctionnent à partir d'images couleurs transformées en niveau de gris, c'est-à-dire ne considérant que l'intensité lumineuse de l'image.

La lumière dénuée de couleurs est dite monochromatique ou achromatique. L'unique attribut d'une telle lumière est son intensité. L'image en *niveau de gris* représente alors l'intensité achromatique de l'image.

D'une image en niveau de gris, on peut d'une part travailler sur l'intensité des objets décrits dans l'image, et d'autre part réduire le coût computationnel du traitement en ne considérant qu'une valeur par pixels.

La luminance est définie de différentes façons, ce qui peut porter à confusion. Globalement, la luminance, définie en *lumens*, mesure la quantité d'énergie qu'un observateur perçoit d'une source de lumière donnée.

Luma ou luminance

On parlera de luma pour décrire la luminance calculée d'un appareil visuel et de luminance lorsqu'il s'agit de la grandeur physique ou de la luminance relative. Les deux termes désignent un même phénomène physique, leur distinction provient de l'usage qui en est fait.

1. Luma est utilisé dans le domaine vidéo. C'est la brillance de l'image, soit la partie du signal vidéo correspondant à l'intensité lumineuse.

   Elle représente donc la part achromatique de la vidéo, mais précisément la "quantité de lumière" présente dans l'image résultante de l'application de la loi de puissance *gamma*.

2. La luminance relative, est l'intensité de l'image calculée selon un espace colorimétrique précis, cette appellation est couramment utilisée en imagerie numérique.

Dans le domaine vidéo, gamma est le facteur qui caractérise le contraste d'un signal visuel. C'est une relation qui approche une loi de puissance liant amplitude de la luminance du signal et luminance réelle de l'image à l'écran.
Ce facteur était initialement dû au fait que les tubes cathodiques avaient une courbe de réponses qui ne réagissait peu ou pas pour des amplitudes de signal trop faible. Le facteur gamma était alors calculé tel qu'il corrigeait le problème.
Le facteur gamma est utilisé pour la calibration du support d'affichage des images afin que celle-ci apparaissent normale à l'œil humain.

La luminance relative et la valeur luma sont calculées identiquement comme une somme pondérée des composantes RGB et suivant le cas après *correction gamma* des images vidéos ou pas :

$$Y' = 0.299 * R' + 0.587 * G' + 0.114 * B'$$

où

Y' est la valeur de luma.

R',G',B' sont les composantes RGB après correction gamma.

Les coefficients permettent de se rapprocher au mieux de la perception humaine des couleurs. On voit en particulier que, dans la perception humaine de l'intensité lumineuse, le vert intervient sensiblement plus, contrairement au bleu.
Ces coefficients sont déterminés relativement au support d'affichage et sont précisés dans des recommandations.
Les plus courantes sont :

1. la recommandation Rec. 709 pour les HDTV donne pour luma :

$$Y = 0.2126 * R + 0.7152 * G + 0.0722 * B$$

2. la recommandation CCIR 601 donne pour luma :

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

Ces recommandations sont faites de telle façons que les couleurs fondamentales et le blanc soient correctes et donc que la représentation de la luminance soit maintenue.

Niveau de gris

L'image en niveau de gris est obtenue en attribuant aux composantes de chaque pixel la valeur en luma ou luminance de ce point.

$$\forall p \in I_{m*n}, \begin{cases} Y'_p = 0.299 * R'_p + 0.587 * G'_p + 0.114 * B'_p \\ p_{[r,g,b]} = [Y'_p, Y'_p, Y'_p] \end{cases}$$

Une autre approche plus simpliste est de calculer le niveau de gris d'un pixels p à partir de sa coloration moyenne :

$$p_{\{r,g,b\}} = \frac{p_r + p_g + p_b}{3}$$

2.2 Caneva

Même s'il n'est pas de structure commune aux algorithmes existants pour la détection et le suivi de cible en mouvement, les étapes suivantes semble prévaloir pour beaucoup de ceux-ci.

Segmentation → Extraction → Analyse → Suivi

2.2.1 Segmentation

A cette étape, il s'agit de dégrossir le matériau brut, de simplifier l'image en ne gardant que ce qui fait sens pour les opérations suivantes. L'objectif sous-tendant à toutes méthodes de traitement d'images étant de minimiser l'usage inutile de ressources computationnelles, la segmentation de l'image est une première approche pour ne plus focaliser que sur le signifiant.
La segmentation est une opération de partitionnement de l'image en un certain nombre de segments. Cette opération est utilisée pour dégager des zones d'intérêts de l'image. Elle assigne une étiquette à chaque pixel de sorte que tous pixels identiquement étiquetés partagent des caractéristiques visuelles données (couleur, intensité, texture). Aussi tous les segments adjacents sont sensiblement différents suivant ces caractéristiques [3].

Il existe différentes méthodes de segmentation, certaines travaillent sur base de régions qu'elles accroissent, décomposent ou fusionnent, d'autres sur les contours, la classification ou le seuillage des pixels en fonction de leur intensité.

Image binaire

La méthode de segmentation la plus simple et la plus rapide est la création d'une image monochrome (aussi appelée *binaire*) par seuillage : À partir de l'image originale convertie en niveaux de gris, on la transforme comparativement à une valeur seuil prédéterminée en une image binaire où chaque pixel prendra une des deux valeurs possibles.

La transformation suivra la règle suivante :
soit une image I de taille $m * n$, un seuil T *global* et $g(x, y)$ le niveau de gris du point (x, y),

$$\forall (x, y) \in I_{\{m,n\}}, (x, y) = \begin{cases} 1 & \text{si } g(x, y) > T \\ 0 & \text{sinon} \end{cases}$$

Où les pixels appartenant à un objet de l'avant-plan sont étiquetés 1 et ceux provenant du fond sont étiquetés 0.

Cette approche grossière de seuillage peut être affinée par l'usage d'un histogramme de niveau de gris, offrant notamment la possibilité de segmenter l'image selon de multiples seuils.
Ou encore, plutôt que d'utiliser un seuil *global*, il est possible de faire dépendre T de propriétés locales du point évalué, comme par exemple de la valeur moyenne du niveau de gris de l'entourage du point considéré. On parlera d'un seuillage *adaptatif* ou *dynamique*.

3. `http://en.wikipedia.org/wiki/Segmentation_(image_processing)`

L'image binaire peut, ensuite, être utilisée comme un masque permettant d'isoler des régions potentiellement intéressantes.

Watershed

L'algorithme de segmentation *Watershed*, traduit par "ligne de partage des eaux", se base sur une interprétation tridimensionnelle de l'image, où un point est caractérisé par ses deux composantes spatiales et son niveau de gris.
De cette image, perçue comme un relief topographique, sera calculée la ligne de partage des eaux pour délimiter le bassin-versant, c'est-à-dire l'aire à l'intérieur de laquelle convergerait de l'eau hypothétiquement tombée.
Trois types de points sont définis par cette interprétation :
- ceux appartenant à un minimum local.
- ceux à partir desquels une goutte d'eau s'écoulerait inévitablement vers une minimum local précis. L'ensemble des points relatés à un même minimum constitueront un bassin-versant de ce minimum.
- ceux à partir desquels toute goutte d'eau ruissellerait équitablement vers l'un ou l'autre minimum. L'ensemble des ces points forment topologiquement une crête, ils constituent la ligne de partage des eaux.

Il existe plusieurs d'implémenter cet algorithme, parmi les plus communes :
- selon la distance topographique d'un point au minimum le plus proche, à partir de chaque pixel de l'image, on suit le gradient jusqu'à atteindre un minimum, à l'image d'un ruissellement.
- par inondation, où est simulé une montée progressive du niveau d'eau à partir des minima du relief.

La segmentation par ligne de partage des eaux donne de bons résultats dans l'extraction d'objet presque uniforme, mais conduit souvent à une sur-segmentation dû aux bruits et irrégularités locales. Une façon de pallier à ce désavantage est d'utiliser des marqueurs. Les marqueurs sont définis comme des composantes connexes appartenant soit à un objet d'avant-plan, soit au fond. La sélection des marqueurs à garder pourra être effectuée par simple estimation du niveau de gris et de la connectivité.

Détection de contours

Intuitivement, un contour est défini comme une suite de pixels contigus reflétant la frontière entre deux régions. La détection des contours permet alors de découper ou fusionner l'image en sous-régions.
En pratique, les principaux algorithmes de détection de contours, *edge detection*, se basent sur l'étude des dérivées de la fonction d'intensité de l'image : le gradient, les extremums locaux et le passage par zéro du Laplacien. Le contour est obtenu par détection d'une discontinuité (changement abrupte d'intensité) et des similarités (selon des critères prédéfinis).

Dû notamment aux différentes méthodes d'acquisition, la frontière entre deux régions n'est pas toujours très contrastée ni exempte d'aucun bruit. De fait, elle apparaîtra généralement floutée et bruitée.
À partir d'une image en niveaux de gris, la frontière d'une région à une autre, représenté en *a*

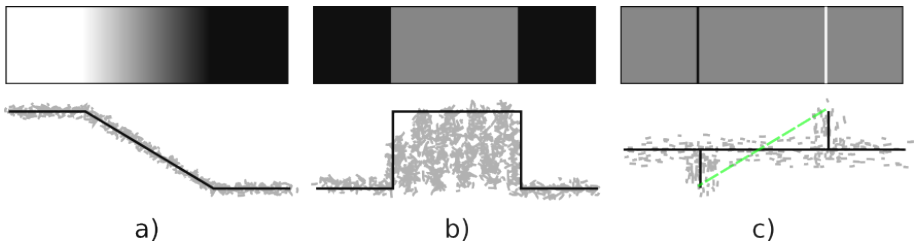


FIGURE 2.2 – Détection de contours

dans la figure 2.2, peut être *idéalement* définie par une fonction rampe dont la longueur sera caractérisé comme le niveau de floutage de la frontière.
Par l'étude de la dérivée première calculée en utilisant le gradient d'un point considéré (2.2 *b*), on peut déterminer si comparativement à un voisinage, on se trouve potentiellement sur un point du contour.
De même, en évaluant la dérivée seconde par application du Laplacien (2.2 *c*), on peut :
- en étudiant son signe, caractériser le point du contour comme appartenant à l'un ou l'autre coté de la frontière.
- déterminer le milieu exacte de la frontière floutée. En calculant la droite imaginaire (2.2 *c*, ligne pointillée) joignant les extremums de la dérivée seconde, on obtient au passage à zéro le point médian.

Le filtre Canny est un des algorithmes les plus utilisés pour la détection des contours, il se base sur l'intensité et la direction du gradient.

2.2.2 Extraction

Des régions brutes délimitées par segmentation, il faut extraire les spécificités en vue d'effectuer les calculs souhaités.

L'idée est de réduire les cibles à leurs caractéristiques internes (pixels compris dans la région) et/ou externes (contours, frontières, coins) ayant une forte signifiance, puis de rassembler ces caractéristiques en descripteurs propre à chaque région. On parlera de *descripteur* pour désigner l'ensemble des traits intéressants (*features*) décrivant l'objet cible.
Par exemple, une région pourrait être représentée par ses frontières et celles-ci serait décrites par des traits spécifiques (position relative des points du bord, périmètre, concavité, etc).

Cette étape met en place un système de représentation des données en vue de les analyser. Autrement dit, on crée un système qui va compacter les données en des représentations utiles pour effectuer des calculs sur les descripteurs.

Le choix de ces traits est laissé libre, ils seront choisis relativement à la façon dont sera entrevue l'analyse des données. Par contre il est impératif que les traits caractéristiques forment des descripteur insensibles aux variations géométriques (homothétie, translation, rotation) ou photométriques (intensité).

Good features to track

Les «bons» traits à suivre sont ceux dont le mouvement peut être estimé de manière fiable.

J. Shi et C. Tomasi ont présenté [4] en 1994 leurs travaux sous l'intitulé *Good Features to Track*, l'article fait toujours autorité sur le sujet.
Ils partent de l'évidence simple qu'aucun système de vision basée sur des traits d'intérêts ne peut fonctionner sans que des bons traits caractéristiques et robustes n'aient pu être préalablement identifiés.
Ils proposent alors un critère de sélection de traits intéressants qu'ils précisent "optimal par construction" car basé sur la façon dont un système de suivi fonctionne.

Du constat qu'il n'existe de traits robustes à toutes épreuves et que même les bons traits peuvent se retrouver caché derrière un obstacle, cet article explique comment contrôler la qualité des traits caractéristiques de l'image.
Ce contrôle s'effectue pendant l'opération de suivi de la cible, par évaluation de la *dissemblance*.
La fonction de dissemblance, définie comme moyenne quadratique des traits, a pour rôle la quantification du changement d'apparence d'un trait entre l'image originale et l'image actuelle.
Lorsque la dissemblance devient trop importante, le trait est abandonné.

Hypothèses concernant les traits intéressants à suivre :
- Luminosité constante : la projection d'un point conserve son apparence d'une frame à la suivante.
- Mouvement court : un point ne bouge de beaucoup.
- Cohérence spatiale : un point se déplace dans son voisinage.

Detecteur de coins

Un coin est un point intersection d'au moins deux arêtes de sens opposé.

Le *Harris corner detector* est une méthode de détection de coins (traits d'intérêt) reconnu pour sa relative robustesse face aux bruits, aux variations de luminosité et aux variations géométriques.

L'algorithme, décrit par Harris et Stephens, fonctionne sur l'évaluation d'une fonction d'*auto-corrélation* appliquée localement. L'auto-corrélation c'est la corrélation croisée du signal par lui-même, cela permet de détecter des régularités, des motifs répétés au sein d'un signal.
La fonction d'auto-corrélation, décrite par [Harris and Stephens, 1988] comme l'erreur quadratique moyenne (*sum of squared differences*), mesure les changements locaux dûs à l'application d'un léger décalage dans chaque direction.

$$E(x,y) = \sum_{u,v} \mathtt{w}(u,v)\ (I(x+u, y+v) - I(u,v))^2$$

où

I est une image en niveau de gris.

$\mathtt{w}(u,v)$ spécifie la taille de la fenêtre considérée, vaudra 1 quand on se trouve dans la région d'intérêt, 0 sinon.

4. IEEE Conference on Computer Vision and Pattern Recognition (CVPR94) Seattle, June 1994

x,y est la quantité de déplacement dans une direction.
Une grande variation de E dans la direction de (x, y) dénote un trait d'intérêt.

Flux Optique

Une caméra filme des objets en mouvement dans un espace à trois dimensions, leur mouvement relatif est un champ de vecteurs à trois composantes.
La scène filmée est projetée sur le plan, en deux dimensions, du flux vidéo. Dès lors, il est possible de définir un champ de vecteurs, le *champ de vitesses projeté* [Bernard, 1999].

Tout point $\mathbf{X}$, réel filmé, possède un point x d'une image dans le flux vidéo qui est le *projeté* $p(\mathbf{X})$ de vitesse $\mathbf{V}$.
Le flux optique est le vecteur $\vec{v} = dp(\mathbf{X})\mathbf{V}$.

L'estimation du mouvement est effectuée à partir de variations temporelles des intensités (niveau de gris) dans le flux d'images. Pour obtenir analytiquement le flux optique, on fait l'hypothèse que chaque point filmé a une intensité constante. On peut alors écrire cette dérivée comme

$$\frac{d}{dt}I(t, x(t)) \equiv \frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} \equiv \vec{v} \cdot \nabla I + \frac{\partial I}{\partial t}$$

où
$I(t, x, y)$ est une image en niveau de gris.

On a, sous l'hypothèse d'intensité constante, l'équation dite du *flux optique* :

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

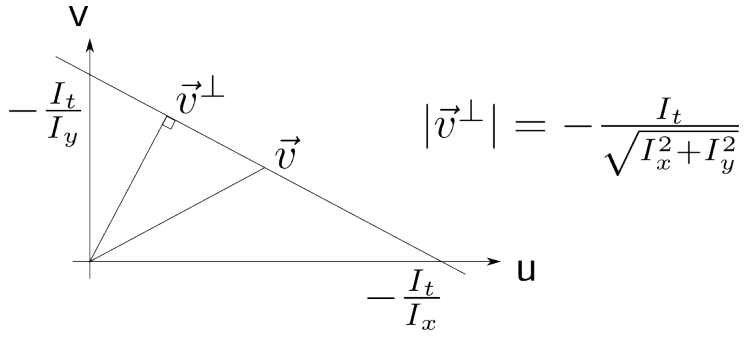


FIGURE 2.3 – Contrainte du flux optique

2.2.3 Analyses et suivi

Cette étape a pour objectif la comparaison des zones d'intérêt et la reconnaissance de celles-ci comme consistantes.
Le suivi de cibles s'apparente à l'étiquetage d'objets comme uniques, au travers du flux vidéo, c'est la reconnaissance des cibles dans la séquence d'image.

Pour en arriver au suivi d'objets, on aura dû effectuer la localisation de cibles par segmentation et leur représentation par extraction de traits d'intérêt. Les images y ont été réduites aux traits essentiels permettant une caractérisation relativement robuste des cibles.
Le procédé de suivi sera alors caractérisé par les opérations de *filtrage* et d'*association de données*.

Le processus de reconnaissance est opéré à partir des descripteurs précédemment extraits dont on aura estimé l'évolution suivant notamment leur dynamique. Il y a donc la création d'une prédiction à partir de laquelle sera évaluée la correspondance avec l'observation de l'image au moment présent.

Évaluation et filtrage Parmi les filtres les plus courant, il y a le filtre de Kalman et les filtres à particules.
Le filtre de Kalman est décrit comme ayant de bons résultats pour des systèmes à dynamique linéaire ou unimodale, typiquement gaussienne. L'approche par filtres à particules semble plus indiquée dans le cadre du suivi de cibles, car la prédiction de l'état courant doit prendre en compte toutes les alternatives possibles simultanément, en ce sens, la dynamique du système y est multimodale. Le filtre *condensation* sera décrit plus en détails dans le chapitre "Méthodes implémentées".

Il existe de nombreuses façon d'évaluer la correspondance de données. Pour évaluer la correspondance, on compare les données dans une zone probable et on retient comme potentiellement correspondant les données pour lesquels on obtient la différence minimale.
Parmi les méthodes les plus connues, pouvant faire office de fonctions de mérite ou de corrélation : La distance de Mahalanobis et la distance euclidienne. Du fait de la simplicité d'implémentation pour un résultat satisfaisant, le calcul de la distance euclidienne a été utilisé.

Distance euclidienne

La correspondance par calcul de la distance euclidienne entre 2 pixels n'est rien d'autre que la racine carrée de la somme du carré des différences en niveau de gris dans une zone $n * n$ autour des 2 pixels.

$$\sqrt{\sum_{i=0}^{n}\sum_{j=0}^{n}(a_{i,j} - b_{i,j})^2}$$

2.3 SURF

2.3.1 Speeded Up Robust Features

Le *Speeded Up Robust Features* (SURF) est un algorithme de *détection de caractéristique* présenté par des chercheurs de l'*École Polytechnique Fédérale* de Zurich et de l'*Université Catholique de Louvain* pour la première fois en 2006 [5], puis dans une version révisée en 2008 [6]. L'objectif de l'algorithme est de rechercher les correspondances entre objets ou scènes présentes dans deux ou plusieurs images distinctes mises en confrontation deux à deux. Il comporte 3 phases.
Dans un premier temps, l'algorithme recherche des points intéressants de l'image, comme par exemple les points de bords ou les jointures en «T». Ces points doivent être faciles à déterminer, c'est-à-dire, que leurs caractéristiques sont telles que leur détermination doit être toujours reproductible sans ambigüité.
Ensuite, l'algorithme associe à chaque point d'intérêt un vecteur caractéristique qui représente un descripteur. Celui-ci, doit être distinct et surtout robuste par rapport au bruit, aux erreurs d'individuation et aux déformations géométriques et photométriques.
Enfin, l'algorithme compare les descripteurs des deux images. Cette comparaison se base généralement sur la distance entre les vecteurs, par exemple, la distance euclidienne.

2.3.2 Recherche des points d'intérêt

Image Intégrale

L'efficacité de l'algorithme SURF est dû principalement à l'utilisation d'une représentation intermédiaire de l'image connue sous le nom d'*image intégrale*.
L'image intégrale est une image numérique qui est calculée rapidement à partir de l'image originale. On l'utilise pour accélérer le calcul de chaque zone rectangulaire dont le sommet supérieur gauche est à l'origine de l'image.
Proposée en 1984 [7] comme méthode d'*infographie*, c'est en 2001 qu'elle a été reformulée par la méthode de Viola et Jones [8], dans le cadre de la *vision par ordinateur*.
Étant donné une image I en *input* et un point de coordonnées (x, y), l'image intégrale $I(x, y)$ est calculée en prenant la somme des valeurs de l'intensité des pixels compris entre le point et l'origine. Formellement, la formule est :

$$I_{\Sigma}(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(x, y)$$

En utilisant l'image intégrale, le calcul de la somme des intensités d'une région rectangulaire quelconque se réduit à quatre opérations. En effet, si l'on considère un rectangle définit par les sommets A, B, C, D (figure 2.4), la somme qui donne la valeur de l'image intégrale vaut :

$$\Sigma = A + D - (C + B)$$

5. Herbert Bay, Tinne Tuytelaars et Luc Van Gool, "*SURF : Speeded Up Robust Features*", dans 9th *European Conference on Computer Vision*, Graz, Autriche, 7-13 mai 2006

6. Herbert Bay, Andreas Ess, Tinne Tuytelaars et Luc Van Gool, "*SURF : Speeded Up Robust Features*", *Computer Vision and Image Understanding*, vol. 110, no 3, 2008, p. 346-359

7. Crow, Franklin (1984). "*Summed-area tables for texture mapping*". SIGGRAPH '84 : *Proceedings of the 11th annual conference on Computer graphics and interactive techniques* : 207–212

8. Paul Viola et Michael Jones, *Robust Real-time Object Detection* IJCV 2001

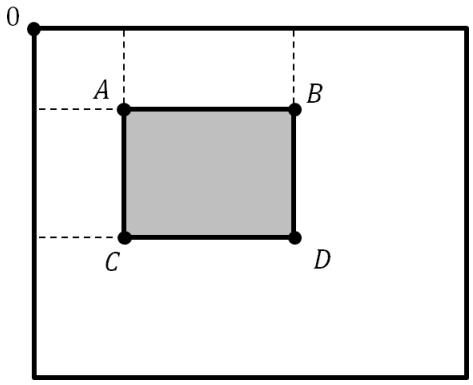


FIGURE 2.4 – Calcul d'une aire en utilisant une image intégrale.

où A, B, C, D sont les intégrales correspondantes aux coordonnées des sommets. Ce calcul est invariant par rapport aux dimensions de la zone considéré et SURF utilise cette propriété pour effectuer de façon efficace la *convolution* lorsque les dimensions des filtres appliqués à l'image varient.
La convolution est l'outil qui permet la construction de filtres linéaires ou de filtres de déplacements invariants. L'équation de convolution, notée $g(x)$, de la séquence $f(x)$ avec une fonction $h(x)$ est :

$$g(x) = f(x) * h(x) = \sum_{\forall k} h(x-k)f(k)$$

$f(x)$ est la fonction d'origine, $g(x)$ est la fonction qui résulte de la convolution et $h(x)$ est le noyau de convolution.

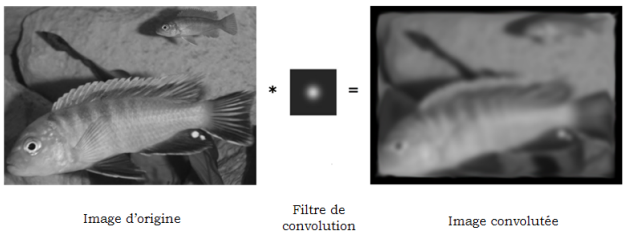


FIGURE 2.5 – Exemple de convolution 2D.

Points d'intérêt basés sur la matrice Hessienne

Le *détecteur* SURF se base sur le déterminant de la *matrice hessienne*. Pour comprendre son fonctionnement, on considère une fonction continue f de deux variables telle que :

$$f : x, y \mapsto f(x, y)$$

La matrice hessienne, H, est la matrice des dérivées partielles de la fonction f :

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial x \partial y} \\ \frac{\partial^2 f(x,y)}{\partial x \partial y} & \frac{\partial^2 f(x,y)}{\partial y^2} \end{bmatrix}$$

Le déterminant de cette matrice est calculé de la façon suivante :

$$|H(f(x,y))| = \frac{\partial^2 f(x,y)}{\partial x^2}\frac{\partial^2 f(x,y)}{\partial y^2} - \left(\frac{\partial^2 f(x,y)}{\partial x \partial y}\right)^2$$

Ce déterminant est utilisé pour trouver le maximum et le minimum de la fonction à travers le test du hessien des dérivées secondes. En appliquant le test, il est alors possible de savoir si un point (x, y) est un extremum local pour la fonction f.
Les dérivées partielles secondes sont calculées en utilisant un *filtre gaussien normalisé du second ordre*, qui permet une analyse à plusieurs échelles et dans l'espace. Grâce à ce calcul, il est possible de déterminer les points du filtre en x, y et xy et de calculer les quatre termes de la matrice. L'utilisation de la gaussienne permet en outre, de faire varier l'effet de lissage durant la phase de convolution de façon à permettre le calcul du déterminant à différentes échelles.
Étant donné que la gaussienne est une fonction *isotrope* (c'est-à-dire à symétrie circulaire), la convolution avec le point est invariant à la rotation. Il est alors possible de calculer la matrice hessienne comme fonction du point $\mathsf{x} = (x, y)$ et de l'échelle σ.

$$H(\mathsf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathsf{x}, \sigma) & L_{xy}(\mathsf{x}, \sigma) \\ L_{xy}(\mathsf{x}, \sigma) & L_{yy}(\mathsf{x}, \sigma) \end{bmatrix}$$

Ici, $L_{xx}(\mathsf{x}, \sigma)$ se réfère à la convolution de la dérivée gaussienne de second ordre $\frac{\partial^2 g(\sigma)}{\partial x^2}$ avec l'image I au point $\mathsf{x} = (x, y)$ et de façon analogue on définit $L_{yy}(\mathsf{x}, \sigma)$ et $L_{xy}(\mathsf{x}, \sigma)$. Ces dérivées sont connues comme LoG (*Laplacian of Gaussian*).

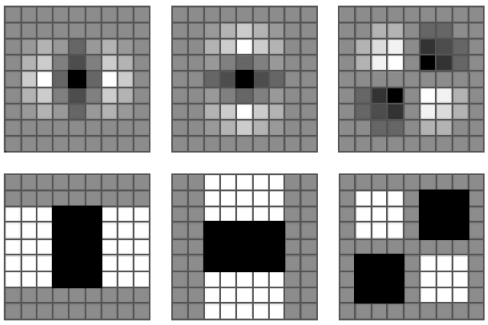


FIGURE 2.6 – Approximation du laplacien de gaussienne.

L'approximation des LoG se fait à travers les approximations des points respectifs. La figure 2.6 montre les similitudes entre les filtres originaux et ceux obtenus par approximation et discrétisation.
Une amélioration des performances peut s'obtenir en utilisant conjointement les filtres avec les images intégrales. Et pour une approximation du determinant de la hessienne plus précise, on utilise une approximation de gaussienne.

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2$$

La perte de precision est amplement compensée par une augmentation d'efficacité et de vitesse. La recherche des maximum locaux dans l'espace à travers les différentes échelles conduit à l'identification des points d'intérêt de l'image.

Espace d'échelle

La théorie des *espaces d'échelle* (*Scale space theory*) est un model qui est apparu progressivement [9] [10] [11] [12] dans le domaine de la vision par ordinateur, pour prendre en compte la nature résolument multi-échelles des données images. L'espace d'échelle (*scale-space*) est donc une fonction qui est utilisée pour trouver des points à travers toute les échelles possibles de l'image. Cette fonction est implémentée comme une pyramide dans laquelle l'image en *input* est iterativement convolutée avec un point gaussien, et à maintes reprises redimensionnée.
Étant donné que les coûts de computation des points utilisés par SURF sont invariants par rapport à leurs dimensions, l'espace d'échelle est créé en appliquant des points toujours plus grands à l'image originale. Ceci permet aux différents niveaux de la pyramide de l'espace d'échelle d'être calculés simultanément dans une éventuelle implémentation *multithread*.

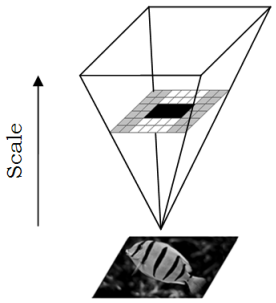


FIGURE 2.7 – Pyramide du filtre.

La figure 2.7 montre l'espace d'échelle utilisée par SURF. Seul le filtre varie, contrairement à l'image originale qui reste invariante.
Dans SURF, le niveau le plus bas d'espace d'échelle est obtenu en appliquant des filtre 9×9 comme ceux de la figure 2.6. Ces filtres correspondent à une gaussienne avec un écart type σ qui vaut 1,2. Les niveaux successifs sont obtenus en incrémentant les filtres de bases et en conservant les proportions (figure 2.8).
Étant donnée que les proportions sont conservées lorsque les dimensions des filtres et l'échelle augmentent, il est possible de calculer la formule suivante :

$$\sigma_{approx} = CurrentFilterSize.\frac{BaseFilterScale}{BaseFilterSize} = CurrentFilterSize.\frac{1,2}{9}$$

Localisation des points d'intérêts

Le processus de détermination de l'échelle qui permet de trouver les points d'intérêt d'une image se divise en 3 partie. Dans un premier temps, on applique un filtre avec un seuil pour faire en sorte d'éliminer les valeurs trop petites (en augmentant ce seuil le nombre de points diminue).
Successivement, on applique une *suppression non-maximale* pour pouvoir trouver un ensemble de points candidats. Chaque pixel de l'espace d'échelle est confronté avec ses 26 points voisins,

9. Witkin, A. P. "*Scale-space filtering*", Proc. 8th Int. Joint Conf. Art. Intell., Karlsruhe, Germany,1019–1022, 1983
10. Koenderink, Jan "*The structure of images*", Biological Cybernetics, 50 :363–370, 1984
11. Florack, Luc, "*Image Structure*", Kluwer Academic Publishers, 1997
12. Romeny, Bart ter Haar, "*Front-End Vision and Multi-Scale Image Analysis*", Kluwer Academic Publishers, 2003

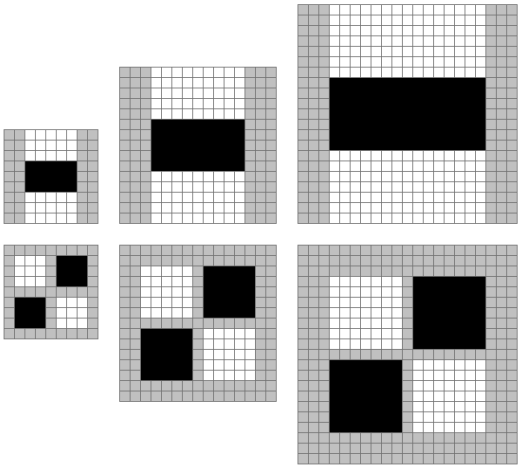


FIGURE 2.8 – Structure du filtre.

y compris les 8 points de l'échelle native et les 9 de chaque échelle supérieure et inférieure (figure 2.9). Le pixel est un maximum s'il a une valeur supérieure à celle des pixels qui l'entourent à son échelle, à l'échelle supérieure et à celle inférieure. Si cette valeur est inférieure, il s'agit d'un minimum. À cette étape on dispose donc d'un ensemble de points d'intérêt filtrés qui sont soit un minimum, soit un maximum dans l'espace d'échelle.

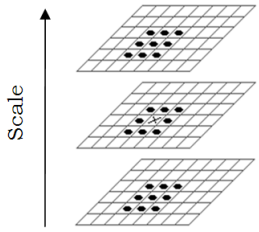


FIGURE 2.9 – Suppression non maximale.

L'étape finale consiste à interpoler les données dans le but de localiser de façon précise les points d'intérêt (les maximums).

2.3.3 Descripteur des points d'intérêt

Les *descripteurs* SURF décrivent la distribution des intensités des pixels autours des points d'intérêt sélectionnés. Ce résultat est possible grâce à l'utilisation des *filtres Haar*, qui servent à déterminer les gradients dans les directions x et y.

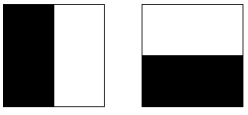


FIGURE 2.10 – *Haar Wavelets.*

Le filtre à gauche de la figure 2.10 calcule la réponse dans la direction x, tandis que celui à droite effectue le calcul par rapport à y. Les poids valent 1 pour les régions noires et -1 pour les régions blanches. Quand ces filtres sont utilisés avec les images intégrales, seules 6 opérations

sont nécessaires pour obtenir un résultat.
L'extraction des descripteurs se divise en 2 phases. Tout d'abord pour chaque point d'intérêt on détermine une orientation, ensuite on construit une fenêtre centrée sur le point, dont la dimension dépend de l'échelle à laquelle le point d'intérêt a été trouvé. À partir de cette zone, et avec l'utilisation conjointe des filtres Haar et de l'image intégrale, on extrait un vecteur de 64 composantes.

Composantes du descripteur

Le premier pas dans l'extraction des descripteurs SURF consiste à construire une fenêtre carrée centrée au point d'intérêt. Cette fenêtre contient les pixels qui produiront le descripteur. Sa dimension est de 20σ, avec σ l'échelle à laquelle le point d'intérêt a été trouvé. La fenêtre est orientée de façon à ce qu'elle concorde à l'orientation du point.

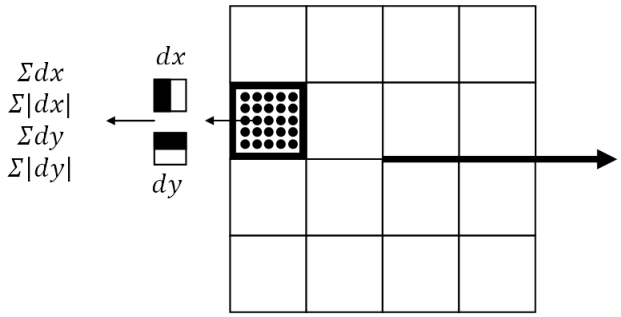


FIGURE 2.11 – Composants du descripteur.

La fenêtre du descripteur est divisée en sous-régions 4×4. À l'intérieur de chaque sous-région les filtres de Haar, de dimension 2σ, sont appliqués sur 25 points uniformément distribués. Pour chaque région (pour les 25 points) on a :

$$v_{sous-region} = \left[\sum dx, \sum dy, \sum |dx|, \sum |dy|\right]$$

À la figure 2.11, le carré à la bordure épaisse est une des 16 sous-régions et les points internes représentent les échantillons sur lesquelles est calculé le résultat donné par les *Haar Wavelets*.

2.4 Exigences

Suite à ce qui a été énoncé, il est intéressant de dégager quelques exigences auxquels se devrait de répondre un système de tracking robuste.
- *Faux positifs, faux négatifs et résistance à la pollution d'éléments parasites*, il convient de ne suivre que ce qui doit l'être.
- *Fiabilité quand à une possible occlusion*, il est fort probable qu'à un moment ou l'autre, la cible sera occulté par un autre élément et réapparaîtra ensuite. Le tracking doit alors rester consistant.
- *Souplesse du tracking*, celui-ci doit pouvoir suivre des éléments aux vitesses variables.
- *Stabilité*, malgré tout, le suivi de la cible doit perdurer.

Chapitre 3

Méthodes implémentées

3.1 Segmentation, traits d'intérêt et comparaison

En liaison à l'implémentation de l'algorithme *condensation*, différentes procédures, telles que décrites dans le chapitre précédent, ont été réalisées.

L'algorithme *condensation*, qui effectue la principale tâche du suivi de cibles, reste ouvert sur les traitements annexes possibles quant au contexte dans lequel il s'effectue.
Entre autres, les choix et les procédés d'extraction des traits d'intérêts, ainsi que la méthode d'estimation de la correspondance entre prédiction et observations, peuvent être réalisés de différentes façons.

L'extraction de l'avant-plan, la transformation en image binaire et la détection de contours ont été implémentées en vue de discerner les zones potentiellement intéressantes.
L'extraction de traits d'intérêts et l'estimation du mouvement (vitesse et direction) ont été utilisées pour établir les descripteurs de chaque cible.
La maximisation de la correspondance, c'est-à-dire la recherche de la différence minimale, a majoritairement été assurée par le calcul de la distance euclidienne.

Les descriptions de ces procédures ont été explicité dans le chapitre précédent.

3.2 Conditionnal Density Propagation

3.2.1 Introduction

Vulgairement, l'algorithme disperse, suivant une certaine probabilité, une suite de particules autour d'un trait d'intérêt.
La correspondance de chacune des particules avec les observations est évaluée et donne un poids.
Il en résulte une pondération des particules qui vient, après ré-échantillonnage, affiner la prédiction pour l'étape suivante.
Le ré-échantillonnage évince les particules à faible probabilité.
Le processus récursif fait converger les particules vers le trait de correspondance maximale.

Concrètement, afin de suivre un objet en mouvement au travers d'un flux vidéo, on doit être en mesure de déterminer à chaque *frame* la position de la cible.

C'est un processus d'anticipation sur le futur, où on peut soit évaluer toutes les possibilités, soit inférer à partir de ce qu'on connait et réduire au maximum le nombre de possibilités à tester.
La détermination est rarement exacte car d'une part, la situation à évaluer est généralement d'une grande complexité et d'autre part, le processus de détermination s'appuie sur de nombreuses mesures, pour la plupart instables.
La déformation de la cible, son occlusion, des changements de luminosité, etc, sont autant de mesures dont la propension à varier aléatoirement contribue à la génération de *bruit* dans la détermination de la cible.

Un des objectifs de la méthode est d'approcher l'hypothétique détermination réelle qui aurait été obtenue aux moyens de mesures idéales.
Autrement dit, la méthode devrait être en mesure de mettre en exergue le tout ou une partie de la cible comme n'appartenant pas au *bruit*.

Le processus de détermination se base sur un mouvement cyclique. Partant d'un modèle donné dont la paramétrisation est issue d'observations antérieures, on consolide ce modèle en évaluant sa pertinence à l'état présent. Ce mouvement conduit à une prédiction sur l'état probable à l'étape suivante.

On distingue alors deux phases, la *prédiction* et l'*observation*.

La prédiction est basée sur un modèle affiné par les informations passées, en vue d'*estimer* l'état a posteriori.

L'observation ou phase de mesure, est la récolte d'informations sur l'état courant du système en vue de *corriger* la prédiction basée sur les mesures précédentes.

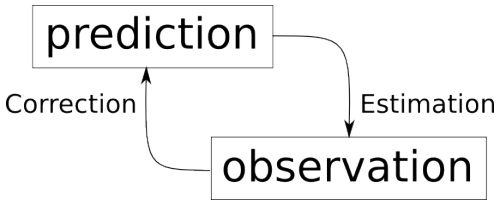


FIGURE 3.1 – Cycle de prédiction - observation

3.2.2 Notation

L'image globale au moment k est notée I_k, son historique est l'ensemble

$$I_{0:n} = \{I_k, k = 0, \cdots, n\}$$

L'objet est la zone d'intérêt dans l'image dont on souhaite faire le suivi.

l'état de l'objet $\mathbf{x}$ au temps t est noté $\mathbf{x}_t$ et son historique est l'ensemble

$$\mathcal{X}_{0:t} = \{\mathbf{x}_0, ..., \mathbf{x}_t\}$$

Les observations sont l'ensemble des *features* (traits *invariants*) aussi dit *descripteur* ou vecteur de *features* $\mathbf{z}_t$ et son historique

$$\mathcal{Z}_{0:t} = \{\mathbf{z}_0, ..., \mathbf{z}_t\}$$

La dynamique stochastique de l'objet est entièrement donnée par

$$P(\mathbf{x}_t|\mathbf{x}_{t-1})$$

(processus Markovien)

Les invariants sont les caractéristiques locales de luminance ou géométrique.

3.2.3 Filtre à particules et Monte-Carlo

L'algorithme *condensation* appartient à la classe des filtres particulaires.
Les filtres à particules, aussi appelé méthodes de Monte-Carlo séquentielles, s'appuient sur des méthodes statistiques. Ils ont pour dessein l'estimation de l'état d'un système donné et qui n'est pas toujours directement observable. *Condensation* contient en plus une approche bayésienne dans son processus d'approximation de l'état à calculer.

L'intérêt des méthodes de Monte-Carlo survient lorsque les méthodes numériques nécessitent une discrétisation de l'espace du problème. Cette discrétisation a pour effet d'augmenter la complexité de la recherche de solutions. Les techniques et optimisations probabilistes simplifient cette recherche de solutions.
Les méthodes Monte-Carlo dites séquentielles sont fondées sur des simulations numériques facilitant aussi le calcul d'une distribution a posteriori. La discrétisation de l'espace des possibles est alors comme borné par une densité donnée.

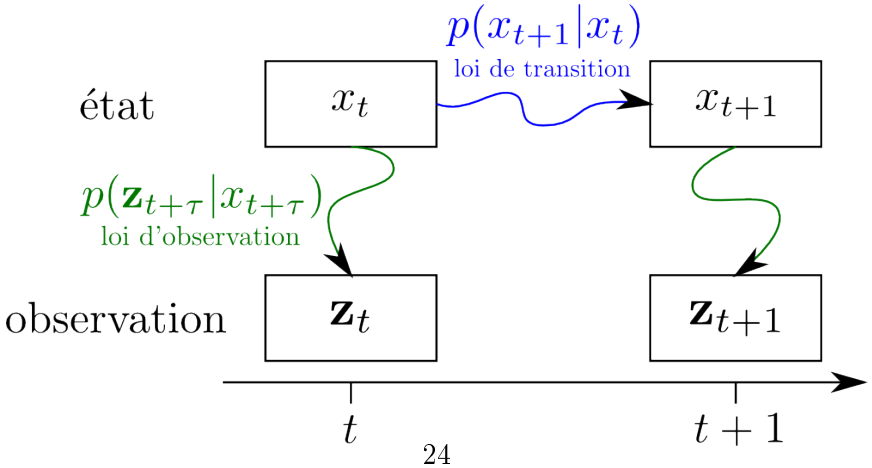
Les composantes du filtre particulaire apposé au suivi d'objets sont :

1. La séquence d'image $I_0, \cdots, I_n$ à des instants $t_0 < \cdots < t_n$ est une suite d'observations $\mathbf{z}_t$ appartenant à la séquence observée $\mathcal{Z}_{0:t} = \{\mathbf{z}_0, ..., \mathbf{z}_t\}$.

   L'image entière est définie comme un système complexe dont l'évolution se manifeste de façon discrète au travers de la séquence vidéo.

2. La cible que l'on souhaite suivre est un état caché $\mathbf{x}_k$ duquel on peut postuler l'existence :
   - d'une suite d'états $\mathbf{x}_1, \cdots, \mathbf{x}_n$ corrélée à une suite d'instants $t_0, \cdots, t_n$
   - d'un état initial $\mathbf{x}_0$ muni d'une *loi initiale* $P(\mathbf{x}_0)$
   - d'une *loi de transition* ou d'*évolution* d'un état $\mathbf{x}_{t-1}$ à $\mathbf{x}_t$, soit $P(\mathbf{x}_t|\mathbf{x}_{t-1})$

3. Une loi d'*observation* $P(\mathbf{z}_{t+\tau}|\mathbf{x}_{t+\tau})$



À partir de ces postulats, il est possible de :

1. *Estimer*, c'est-à-dire établir une *loi de proposition* caractérisant la relation entre l'état de l'objet conditionnellement aux observations : $P(\mathbf{x}_t|\mathcal{Z}_{1:t})$
2. *Filtrer*, soit connaître $P(\mathbf{x}_t|\mathcal{Z}_{1:t+\tau})$
3. *Prédire*, soit déterminer $P(\mathbf{x}_t|\mathcal{Z}_{1:t-1})$

L'algorithme *condensation* propose donc d'approcher l'état du système a posteriori en utilisant une connaissance a priori de celui-ci.
À partir d'une connaissance a priori du système et d'observations desquelles peut être caractérisée la variabilité statistique des données z d'une image relativement à l'état de l'objet x, la distribution a posteriori $P(\mathcal{X}_{0:t}|\mathcal{Z}_{1:t})$ peut être estimée pour chaque x_t étant donné z_t au temps t.

3.2.4 Chaîne Markov

L'hypothèse est faite que la dynamique des objets forme une chaîne de Markov telle que

$$P(\mathbf{x}_t|\mathcal{X}_{0:t}) = P(\mathbf{x}_t|\mathbf{x}_{t-1})$$

Le processus stochastique possède la propriété de Markov si la distribution conditionnelle de probabilités des états futurs, étant donné les états passés et l'état présent, ne dépend que de l'état présent. On parle d'absence de mémoire au sein du processus. La dynamique future de l'objet ne dépend pas de la séquence passée, mais uniquement de son état au temps présent.

3.2.5 Approche bayésienne

Le théorème de Bayes est :

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)}$$

où transposée à la problématique du suivi de cible, elle s'écrit :

$$P(\mathcal{X}_{0:t}|\mathcal{Z}_{1:t}) = \frac{P(\mathcal{Z}_{1:t}|\mathcal{X}_{0:t})P(\mathcal{X}_{0:t})}{\int_{\mathcal{X}_{t+1}} P(\mathcal{Z}_{1:t}|\mathcal{X}_{0:t})P(\mathcal{X}_{0:t})d\mathcal{X}_{0:t}}$$

Cette formulation, que le coût computationnel rend impraticable comme telle, devra être résolue de façon récursive. La formulation récursive fait apparaître la notion de vraisemblance et d'a priori sur l'état prédit.

$$P(\mathcal{X}_{0:t+1}|\mathcal{Z}_{1:t+1}) = P(\mathcal{X}_{0:t}|\mathcal{Z}_{1:t})\frac{P(\mathbf{z}_{t+1}|\mathbf{x}_{t+1})P(\mathbf{x}_{t+1}|\mathbf{x}_t)}{P(\mathbf{z}_{t+1}|\mathcal{Z}_{1:t})}$$

où,
la vraisemblance est donnée comme $P(\mathbf{z}_{t+1}|\mathbf{x}_{t+1})$
l'a priori est $P(\mathbf{x}_{t+1}|\mathbf{x}_t)$

La densité a posteriori est exprimable comme fonction de l'*observation* (évaluation de vraisemblance) et de la loi de *transition*.

3.2.6 Observation

Les observations $\mathbf{z}_t$ sont indépendantes entre-elles et vis-à-vis du processus dynamique :

$$P(\mathcal{Z}_{1:t-1}, \mathbf{x}_t | \mathcal{X}_{0:t-1}) = P(\mathbf{x}_t | \mathcal{X}_{0:t-1}) \prod_{t=1}^{t-1} P(\mathbf{z}_t | \mathbf{x}_t)$$

ce qui se réduit, considérant la condition mutuelle d'indépendance des observations, en

$$P(\mathcal{Z}_{1:t} | \mathcal{X}_{0:t}) = \prod_{i=1}^{t} P(\mathbf{z}_i | \mathbf{x}_i)$$

Le processus d'observation est alors défini en spécifiant la densité conditionnelle $P(\mathbf{z}_t|\mathbf{x}_t)$ pour chaque instant t

3.2.7 Propagation

À partir des observations, la densité conditionnelle de l'état au moment t est $P_t(\mathbf{x}_t) \equiv P(\mathbf{x}_t|\mathcal{Z}_{1:t})$. Elle représente toute l'information de l'état pouvant être déduite de l'entièreté du flux de données. Selon le théorème de Bayes, on déduit la règle de propagation de la densité de l'état dans le temps comme

$$P(\mathbf{x}_t|\mathcal{Z}_t) = k_t P(\mathbf{z}_t|\mathbf{x}_t) P(\mathbf{x}_t|\mathcal{Z}_{1:t-1})$$

où k représente une constante de normalisation ne dépendant pas de $\mathbf{x}_t$.
La densité *a priori* $P(\mathbf{x}_t|\mathcal{Z}_{1:t-1})$ est une prédiction issue de la densité *a posteriori* $P(\mathbf{x}_{t-1}|\mathcal{Z}_{1:t-1})$ à laquelle a été surimposé un pas de temps du modèle dynamique (temporalité indépendante du flux vidéo). La règle de propagation est alors

$$P(\mathbf{x}_t|\mathbf{z}_t) = kP(\mathbf{z}_t|\mathbf{x}_t)P(\mathbf{x}_t)$$

Pour atteindre cette densité *a priori* tout en évitant un coût computationnel conséquent, celle-ci est approchée de façon récursive par échantillonnage pondéré.

3.2.8 Algorithme

L'hypothèse du processus markovien et l'approche bayésienne posées, on défini récursivement l'estimation du système comme :

$$\texttt{Est}(t) = P(\mathbf{z}_t|\mathbf{x}_t) * P(\mathbf{x}_t|\mathbf{x}_{t-1}) * \texttt{Est}(t-1)$$

L'approximation récursive se déroule suivant le canevas :

Prédiction → Mesure → Échantillonnage

Prédiction

La prédiction de la position d'une nouvelle particule $\mathbf{s}'^{(k)}_t$ est obtenue à partir de la densité a priori $P(\mathbf{x}_t|\mathbf{x}_{t-1})$
où
$P(\mathbf{x}_t|\mathbf{x}_{t-1} = \mathbf{s}^{(k)}_{t-1})$, pour $k = 1, \cdots, N$
$\mathbf{s}^{(k)}_{t-1}$ est issu du précédent échantillonnage.

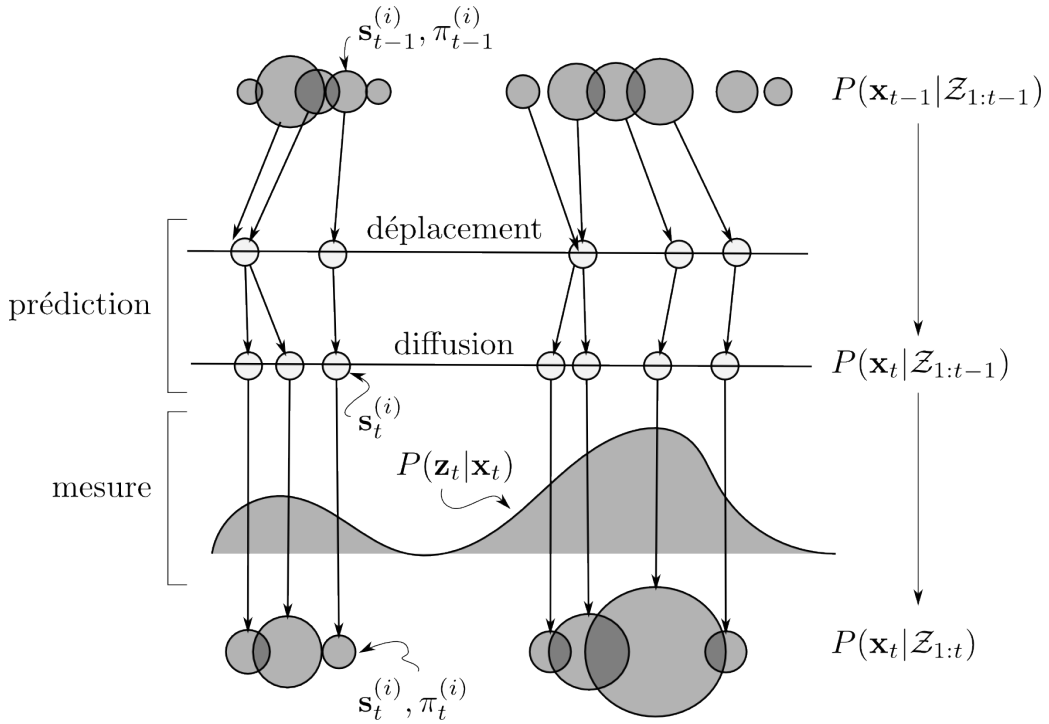


FIGURE 3.2 – Visualisation d'un pas de l'algorithme

Mesure

La pondération des particules déterminant la probabilité π_t est obtenue à partir de la loi d'observation $P(\mathbf{z}_{t+}|\mathbf{x}_t)$
où
$P(\mathbf{z}_{t+}|\mathbf{x}_t = \mathbf{s}'^{(k)}_t)$, pour $k = 1, \cdots, N$
$\mathbf{s}'^{(k)}_t$ est issu de la prédiction

Échantillonnage

Il s'agit de retrouver un objet $\mathbf{x}$ de densité *a priori* $P(\mathbf{x})$ en utilisant les données observées $\mathbf{z}$ d'une image. La densité *a posteriori* est calculée récursivement par échantillonnage pondéré dont le canevas est le suivant :

1. Soit un vecteur S_{t-1} de N échantillons pris sur l'image I_{t-1},

$$S_{t-1} = \{\mathbf{s}_{t-1}^{(i)}, \pi_{t-1}^{(i)}, i = 1, \cdots, N\}$$

   Initialement, les poids $\pi_0^{(i)}$ valent $\frac{1}{N}$

2. S_t est obtenu :

   `for` $k = 1$ à N,

   On sélectionne un échantillon $\mathbf{s}_t^{(k)}$ du vecteur S_{t-1} relativement à son poids $\pi_{t-1}^{(i)}$ issu la probabilité π_{t-1}

On obtient un vecteur

$$S_t = \{\mathbf{s}_t^{(i)}, \pi_t^{(i)}, i = 1, \cdots, N\}$$

où les faibles particules ont été évincées.

Chapitre 4

Résultats expérimentaux

4.1 Différences

La réalisation présentée à l'issue du projet se différencie de l'approche proposée par les auteurs M. Isard et B. Andrew, en un certain nombre de points :

1. *Condensation* est présenté avec l'utilisation de *splines* pour descripteurs [Isard, 1998]. Dans ce travail, le choix a été fait de baser le suivi de cible sur des points robustes non mis en relation, à la manière d'un *blob*.
2. La problématique du suivi de poissons à orienté l'implémentation vers une solution potentiellement multi-cibles.

4.2 Scénario de solution logicielle

La trame de la solution esquissée par le présent travail se décrit comme suit :

1. Détection de cible
   (a) Segmentation
   (b) Sélection des points intéressants
   (c) Mémorisation de la cible
2. Suivi de cible
   (a) Identification des traits d'intérêt
      i. Choix de candidats selon une distribution
      ii. Élection du meilleur candidat : première passe et ré-échantillonnage
   (b) Élimination des traits non robustes ou perdus

L'application commence par segmenter l'image et y détecter des cibles, elle ne retournera à cette étape de segmentation que sous la condition d'avoir perdu un certain nombre de traits d'intérêt (seuil).
Les cibles détectées, commence la phase de suivi dans laquelle la procédure cycle autant que possible.
De cette façon, l'application alterne des phases de recherche de cibles et de suivi de cibles.

4.3 Le logiciel

4.3.1 Aperçu

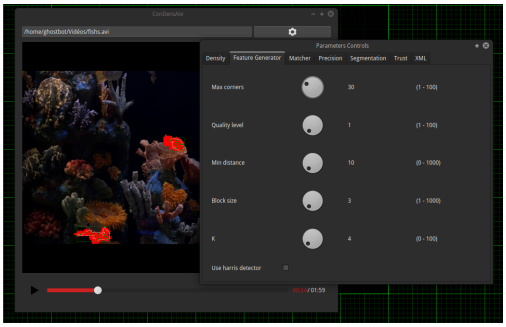


FIGURE 4.1 – aperçu de l'application ConDensAte

Le logiciel (4.1) se compose d'une fenêtre principale au sein de laquelle peut être lu un flux vidéo provenant :
– d'un fichier.
– d'une caméra locale ou usb.
– d'une caméra en réseau (4.2).

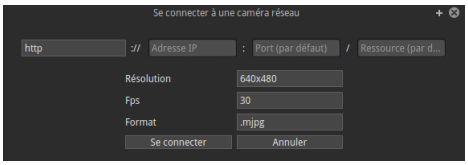


FIGURE 4.2 – Connection à une caméra en réseau

En vue de répondre adéquatement, au diverses qualités et résolutions de vidéo, l'application propose de pouvoir paramétrer (4.3) les différentes étapes de l'algorithme et de sauvegarder ces réglages au format XML.

FIGURE 4.3 – Paramétrisation

4.3.2 En action

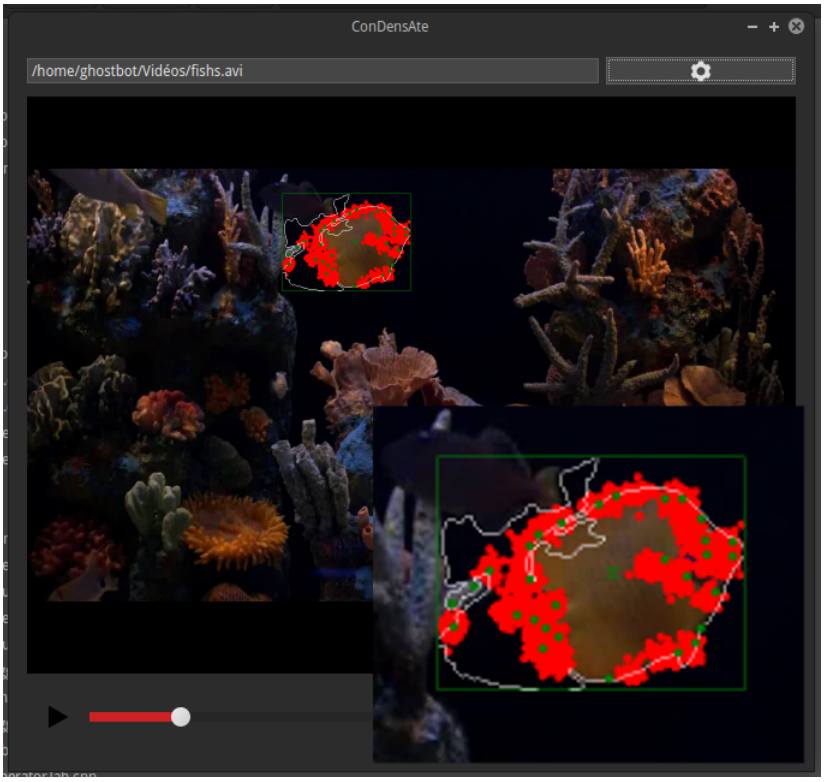


FIGURE 4.4 – ConDensAte : Recherche de cibles

Détection de cibles L'application segmente l'image afin de cerner les régions potentiellement intéressantes (figure 4.4 cadre vert).
Cette segmentation, par différentiation des objets en mouvement et réalisée en utilisant un masque binaire, débouche sur une détection des contours de la cible éventuelle (figure 4.4, trait blanc).
Au sein des contours, la recherche et l'extraction des traits d'intérêt, selon les recommandations de Shi et Tomasi [Shi and Tomasi, 1994], est effectuée. Un vecteur contenant les traits d'intérêt est alors créé.

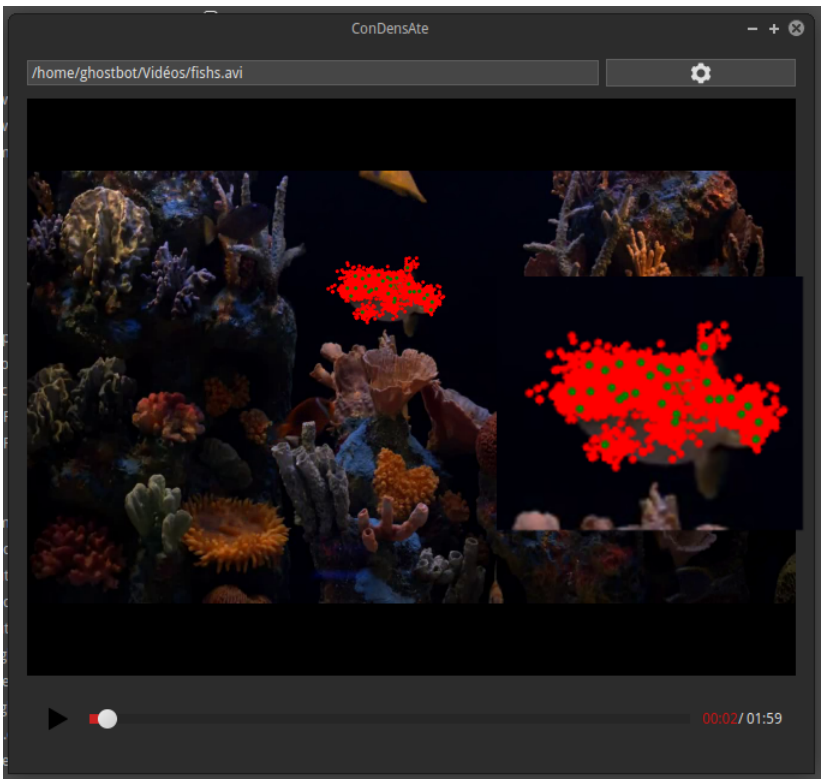


FIGURE 4.5 – ConDensAte : Suivi de cibles

Suivi de cibles À partir du vecteur de traits d'intérêts, pour chacun d'eux, l'application va établir des prédictions suivant une distribution donnée.
Sur la figure 4.5, les points rouges représentent les particules de la prédiction courante, les points verts les traits de la prédiction précédente qui ont satisfont au mieux la correspondance avec les traits d'intérêts initiaux.

4.3.3 Code source

L'application et le code source sont téléchargeables à l'adresse :

`www.webresearch.be/fishtube`

Chapitre 5

Conclusion et perspectives

L'application donne des résultats, dans une certaine mesure, satisfaisants. Mais l'implémentation telle que réalisée peut être largement améliorée.

Le algorithmes annexes (segmentation, vraisemblance, détection de *features*) qui ont été utilisé ne sont pas forcément ceux qui donnent les résultats les plus précis.

Par exemple, il aurait été intéressant d'intégrer la puissance de l'algorithme SURF, ce qui aurait permis d'ajouter à la détection et au suivi de cibles, la reconnaissance du type de cibles selon des *templates* données.

Bibliographie

[Arnaud et al., 2004] Arnaud, E., Memin, E., Cernuschi-Frias, B., et al. (2004). Filtrage conditionnel pour la trajectographie dans des sequences d'images-application au suivi de points. In *14e congrès francophone de Reconnaissance des formes et d'Intelligence artificielle (RFIA'04)*.

[Arulampalam et al., 2002] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2) :174–188.

[Bardet, 2009] Bardet, F. (2009). *Suivi et catégorisation multi-objets par vision artificielle. Applications au suivi de personnes et de véhicules.* PhD thesis, Université Blaise Pascal-Clermont-Ferrand II.

[Bernard, 1999] Bernard, C. (1999). Ondelettes et problèmes mal posés : la mesure du flot optique et l'interpolation irrégulière. *Thèse de l'École Polytechnique.*

[Bréhard and Le Cadre,] Bréhard, T. and Le Cadre, J.-P. Estimation distribuée pour un probleme de filtrage non linéaire.

[Brethes et al., 2004] Brethes, L., Menezes, P., Lerasle, F., and Briot, M. (2004). Segmentation couleur et condensation pour le suivi et la reconnaissance de gestes humains. In *Reconnaissance des Formes et Intelligence Artificielle*, volume 2, pages 967–975.

[Breton, 2004] Breton, M. (2004). Suivi temporel d'objets en 2d et 3d : revue de littérature sur le filtre de kalman et autres méthodes par propagation de densité de probabilité conditionnelle. *Report for the course, Laval University.*

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (8) :679–714.

[Crow, 1984] Crow, F. (1984). Summed-area tables for texture mapping. In '84, S., editor, *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 207–212.

[E. and K., 2006] E., T. and K., P. (2006). Video tracking : a concise survey. *Oceanic Engineering, IEEE Journal*, 31(2) :520–529.

[Faro et al.,] Faro, A., Giordano, D., Palazzo, S., and Spampinato, C. Fish detection and tracking. *IST – 257024 – Fish4Knowledge.*

[Florack, 1997] Florack, L. (1997). *Image Structure.* Kluwer Academic Publishers.

[Fontaine et al.,] Fontaine, E., Barr, A., and Burdick, J. Tracking of multiple worms and fish for biological studies. *http ://www.cvl.iis.u-tokyo.ac.jp/mva/proceedings/2007CD/papers/11-02.pdf.*

[Fontmarty, 2008] Fontmarty, M. (2008). *Vision et filtrage particulaire pour le suivi tridimensionnel de mouvements humains : applications à la robotique.* PhD thesis, Université Paul Sabatier-Toulouse III.

[Gonzalez and Woods, 2002] Gonzalez, R. C. and Woods, R. E. (2002). *Digital image processing.* Number 2. Prentice-Hall Inc.

[Gyaourova et al., 2003] Gyaourova, A., Kamath, C., and Cheung, S.-C. (2003). Block matching for object tracking. *Lawrence LiverMore National Laboratory.*

[Hamlaoui and Davoine, 2005] Hamlaoui, S. and Davoine, F. (2005). Suivi des mouvements faciaux et de la pose 2d d'un visage. In *20° Colloque sur le traitement du signal et des images, FRA, 2005.* GRETSI, Groupe d'Etudes du Traitement du Signal et des Images.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

[Herbert Bay and Gool, 2006] Herbert Bay, T. T. and Gool, L. V. (2006). Surf : Speeded up robust features. In *9th European Conference on Computer Vision.* Graz, Austria.

[Herbert Bay and Gool, 2008] Herbert Bay, Andreas Ess, T. T. and Gool, L. V. (2008). Surf : Speeded up robust features. *Computer Vision and Image Understanding*, 110(3) :346–359.

[Hue et al., 2001] Hue, C., Le Cadre, J.-P., and Pérez, P. (2001). Trajectographie multi-objets par filtrage particulaire. *Actes des journes francophones des jeunes chercheurs eni analyse dirnages et perception visuelle, ORASIS*, pages 511–520.

[Isard and Blake, 1998] Isard, M. and Blake, A. (1998). Condensation—conditional density propagation for visual tracking. *International journal of computer vision*, 29(1) :5–28.

[Isard, 1998] Isard, M. A. (1998). *Visual Motion Analysis by Probabilistic Propagation of Conditional Density.* Departement of engineering science, University of Oxford.

[Jain et al., 1996] Jain, A., IEEE, F., Zhong, Y., and Lakshmanan, S. (1996). Object matching using deformable templates. *Oceanic Engineering, IEEE Journal.*

[Kim,] Kim, Y. M. Object tracking in a video sequence, cs 229 final project report. *CS 229, Stanford University.*

[Koenderink, 1984] Koenderink, J. (1984). The structure of images. *Biological Cybernetics*, 50 :363–370.

[Koller-Meier and Ade, 2001] Koller-Meier, E. B. and Ade, F. (2001). Tracking multiple objects using the condensation algorithm. *Robotics and Autonomous Systems*, 34(2) :93–105.

[Legland, 2003] Legland, F. (2003). Filtrage particulaire. In *Proceedings 19eme Colloque GRETSI sur le Traitement du Signal et des Images*, volume 1, pages 1–8.

[Maheo and Colas,] Maheo, A.-C. and Colas, R. M. Méthodes de suivi d'un objet en mouvement sur une vidéo. *Département d'ingénierie et des sciences informatiques, Institut Supérieur de l'Électronique et du Numérique.*

[Moeslund, 2012] Moeslund, T. (2012). Introduction to video and image processing. *Undergraduate Topics in Computer Science.*

[Press et al., 2007] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes the art of scientific computing.* Number 3. Cambridge University Press.

[Romeny, 2003] Romeny, B. t. H. (2003). *Front-End Vision and Multi-Scale Image Analysis.* Kluwer Academic Publishers.

[Rova et al.,] Rova, A., Mori, G., and Dill, L. One fish, two fish, butterfish, trumpeter : Recognizing fish in underwater video. *http ://www.cvl.iis.u-tokyo.ac.jp/mva/proceedings/2007CD/papers/11-02.pdf.*

[Schmid and Mohr, 1997] Schmid, C. and Mohr, R. (1997). Local grayvalue invariants for image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(5) :530–535.

[Sellent et al.,] Sellent, A., Eisemann, M., and Magnor, M. Two algorithms for motion estimation from alternate exposure images. *Institut fur Computergraphik, TU Braunschweig, Germany.*

[Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. *9th IEEE Conference on Computer Vision and Pattern Recognition.*

[Spampinato et al., 2008] Spampinato, C., Chen-Burger, Y., Nadarajan, G., and Fisher, R. (2008). Detecting, tracking and counting fish in low quality unconstrained underwater videos. *VISAPP (2)*, pages 514–519.

[Suzuki and Abe, 1985] Suzuki, S. and Abe, K. (1985). Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics and Image Processing.*

[Tay and Sung, 2001] Tay, T. and Sung, K. K. (2001). Probabilistic learning and modelling of object dynamics for tracking. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 648–653. IEEE.

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Robust real-time object detection. *International Journal of Computer Vision.*

[Wang et al.,] Wang, Y., Doherty, J., and Dyck, R. V. Moving object tracking in video. *Department of Electrical Engineering, The Pennsylvania State University, National Institute of Standards and Technology.*

[Witkin, 1983] Witkin, A. P. (1983). Scale-space filtering. In *Proceedings of the 8th International joint conference in artificial intelligence*, pages 1019–1022. Karlsruhe, Germany.

Table des figures