Approche crédibiliste pour la fusion multi capteurs décentralisée
Cyrille André

HAL Id: tel-00976761
https://tel.archives-ouvertes.fr/tel-00976761
Submitted on 10 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Approche crédibiliste pour la fusion multi capteurs décentralisée
## Table des matières

1 Introduction générale ........................................ 5  
1.1 Fusion de données ........................................ 6  
1.2 Systèmes multi capteurs ................................ 6  
1.3 Contexte applicatif ........................................ 8  
1.4 Objectifs de la fusion de données .......................... 9  
  1.4.1 Suivi des cibles ....................................... 9  
  1.4.2 Synthèse et précision de l’information ................. 9  
  1.4.3 Réduction des fausses alarmes .......................... 9  
  1.4.4 Détectio et dysfonctionnement de capteurs ............. 10  
1.5 Taxonomie et organisation fonctionnelle ................... 10  
  1.5.1 Hiérarchie des niveaux d’abstraction .................. 10  
  1.5.2 Découpage en blocs fonctionnels ....................... 11  
1.6 Évolution des architectures ................................. 12  
  1.6.1 Architectures hiérarchiques ............................ 13  
  1.6.2 Architectures distribuées .............................. 14  
  1.6.3 Architectures décentralisées ............................ 14  
1.7 Motivations pour de nouvelles approches .................. 15  
  1.7.1 Limitations du filtre de Kalman ....................... 15  
  1.7.2 Alternatives ......................................... 16  
1.8 Organisation du document .................................. 16  

2 Le modèle des croyances transférables (TBM) .............. 19  
2.1 Introduction ................................................ 20  
2.2 Un modèle non probabiliste ............................... 20  
  2.2.1 Fonctions de masse (BBAs) ........................... 20  
  2.2.2 Croyance, implicabilité, plausibilité et communalité ... 22  
2.3 Combinaisons d’informations ............................... 23  
  2.3.1 Principe de moindre engagement ....................... 24  
  2.3.2 Conditionnement ..................................... 25  
  2.3.3 Combinaisons .......................................... 26  
  2.3.4 Fiabilité des sources et affaiblissement ............... 28  
  2.3.5 Modifications du cadre de discernement ............... 28  
2.4 Décomposition canonique ................................. 30  
  2.4.1 Décomposition canonique conjonctive .................. 30  
  2.4.2 $w$ ordonnancement ................................... 32
2.4.3 Règle conjonctive prudente .......................... 32
2.4.4 Règle disjonctive hardie ............................ 33
2.5 Prise de décision ....................................... 34
   2.5.1 Transformation pignistique .......................... 35
   2.5.2 Minimisation du risque ............................. 36
   2.5.3 Décisions alternatives .............................. 36
2.6 Comparaison de BBA s .................................. 37
   2.6.1 Similarité ......................................... 37
   2.6.2 Conflit ............................................ 39
2.7 Interprétations de la théorie des croyances ............... 39

3 Cadre crédibiliste pour le pistage .......................... 43
   3.1 Définition du problème ............................... 44
   3.2 Cadres de discernement ............................... 45
      3.2.1 Modélisation de la localisation des pistes ...... 46
      3.2.2 Spécificités de l’application ...................... 46
      3.2.3 Arguments en faveur du TBM ....................... 47
   3.3 Construction des BBA s ............................... 48
      3.3.1 Représentation des détections ...................... 48
      3.3.2 Informations topographiques ....................... 51
      3.3.3 Conditionnement ................................... 52
      3.3.4 Changement d’échelle ............................ 53
   3.4 Fusion des pistes et des détections .................... 54
      3.4.1 Règle de combinaison .............................. 54
      3.4.2 Prédiction ........................................ 56
      3.4.3 Ré troprojection des pistes ......................... 57
      3.4.4 Mise à jour des paramètres ....................... 58
   3.5 Analyse de sensibilité ................................ 59
      3.5.1 Localisation mono capteur ......................... 60
      3.5.2 Localisation multi capteurs ....................... 63
      3.5.3 Synthèse de l’analyse ............................ 65
   3.6 Conclusions ......................................... 65

4 Suivi multi pistes ........................................ 67
   4.1 Association détections-pistes : généralités ............ 68
      4.1.1 Stratégies d’association ........................... 68
      4.1.2 Notations et hypothèses de travail ................. 68
      4.1.3 Méthodes de résolution ........................... 70
   4.2 Traitements dans le cadre du TBM ...................... 72
      4.2.1 BBA s d’association ................................ 72
      4.2.2 Coûts crédibilistes ............................... 73
      4.2.3 Plausibilité globale ................................ 75
      4.2.4 Prise en compte de l’engagement ................... 77
   4.3 Résultats ............................................. 80
   4.4 Conclusion .......................................... 84
5 Simplification des fonctions de masse

5.1 Introduction ........................................... 86
5.2 Méthodes de simplification ............................... 86
  5.2.1 Summerization .................................. 87
  5.2.2 Agrégation itérative .............................. 87
  5.2.3 Minimisation d’une distance ....................... 88
  5.2.4 Simplification de SSFs ............................ 89
5.3 Expérimentations .................................... 91
    5.3.1 Taux de bonne classification après combinaison ... 91
    5.3.2 Distance de Jousselme .......................... 93
5.4 Conclusion ........................................... 95

6 Fusion décentralisée

6.1 Définition du problème .................................. 98
    6.1.1 Éléments du déploiement ......................... 98
    6.1.2 Objectifs et difficultés ........................... 99
6.2 Principes de la méthode ................................ 100
    6.2.1 Application locale de la fusion .................. 100
    6.2.2 Unicité des pistes ................................ 100
    6.2.3 Unicité des associations .......................... 100
6.3 Communications entre agents ............................ 100
    6.3.1 Attribution des détections ....................... 100
    6.3.2 Création d’une piste ............................. 101
    6.3.3 Transfert d’une piste ............................ 102
    6.3.4 Disparition d’une piste ........................... 103
6.4 Conclusion ........................................... 103

7 Système de surveillance Smartmesh

7.1 Description générale du projet .......................... 106
    7.1.1 Capteurs et sources de données .................. 106
    7.1.2 Découpage fonctionnel du projet ................ 107
    7.1.3 Composants logiciels en interaction avec la fusion ... 109
7.2 Agent de fusion SmartMesh .............................. 112
    7.2.1 Plan de déploiement ............................. 112
    7.2.2 Construction des BBAs ........................... 114
    7.2.3 Gestion des communications ..................... 115
    7.2.4 Module de fusion ................................ 116
7.3 Installation et configuration du système ............... 116
    7.3.1 Préparation du déploiement ...................... 117
    7.3.2 Initialisation de la fusion ....................... 117
    7.3.3 Configuration dynamique ......................... 118
7.4 Conclusions ........................................... 119
# Table des matières

## 8 Essais et expérimentations

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Plan d’intégration et d’expérimentation</td>
<td>122</td>
</tr>
<tr>
<td>8.2 Cadre expérimental</td>
<td>122</td>
</tr>
<tr>
<td>8.2.1 Outils d’analyse</td>
<td>122</td>
</tr>
<tr>
<td>8.2.2 Présentation des sites de déploiements</td>
<td>124</td>
</tr>
<tr>
<td>8.3 Scénarios de déplacement de cibles</td>
<td>126</td>
</tr>
<tr>
<td>8.3.1 Suivi d’une piste</td>
<td>126</td>
</tr>
<tr>
<td>8.3.2 Croisement de deux pistes</td>
<td>127</td>
</tr>
<tr>
<td>8.3.3 Séparation d’un groupe</td>
<td>130</td>
</tr>
<tr>
<td>8.4 Étude de sensibilité</td>
<td>132</td>
</tr>
<tr>
<td>8.5 Conclusions</td>
<td>135</td>
</tr>
</tbody>
</table>

## 9 Conclusions et perspectives

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1 Détection d’intrusions par caméra visible</td>
<td>142</td>
</tr>
<tr>
<td>10.1.1 Objectifs et hypothèses de travail</td>
<td>142</td>
</tr>
<tr>
<td>10.1.2 Détection du changement</td>
<td>142</td>
</tr>
<tr>
<td>10.1.3 Pistage</td>
<td>148</td>
</tr>
<tr>
<td>10.1.4 Organisation et fonctionnalités de l’application</td>
<td>151</td>
</tr>
<tr>
<td>10.2 Spécification des plans de déploiement</td>
<td>152</td>
</tr>
<tr>
<td>10.2.1 Format KML</td>
<td>152</td>
</tr>
<tr>
<td>10.2.2 Contenu d’un fichier de déploiement</td>
<td>158</td>
</tr>
<tr>
<td>10.2.3 Données topographiques</td>
<td>160</td>
</tr>
</tbody>
</table>
Chapitre 1

Introduction générale

Sommaire

1.1 Fusion de données ........................................... 6
1.2 Systèmes multi capteurs ................................. 6
1.3 Contexte applicatif ....................................... 8
1.4 Objectifs de la fusion de données ....................... 9
1.4.1 Suivi des cibles ....................................... 9
1.4.2 Synthèse et précision de l’information ............... 9
1.4.3 Réduction des fausses alarmes ....................... 9
1.4.4 Détection du dysfonctionnement de capteurs ....... 10
1.5 Taxonomie et organisation fonctionnelle ............... 10
1.5.1 Hiérarchie des niveaux d’abstraction ............... 10
1.5.2 Découpage en blocs fonctionnels .................... 11
1.6 Évolution des architectures .............................. 12
1.6.1 Architectures hiérarchiques .......................... 13
1.6.2 Architectures distribuées .............................. 14
1.6.3 Architectures décentralisées .......................... 14
1.7 Motivations pour de nouvelles approches ............... 15
1.7.1 Limitations du filtre de Kalman ...................... 15
1.7.2 Alternatives ........................................... 16
1.8 Organisation du document ............................... 16


1.1 Fusion de données

La fusion de données consiste à combiner plusieurs observations d’un environnement ou d’un phénomène afin de produire une description plus robuste, plus précise ou plus complète. L’idée sous-jacente est de remédier à l’imperfection des informations en exploitant la redondance ou la complémentarité des données.

Le concept de fusion d’informations est naturel. Par exemple, les êtres humains peuvent combiner les perceptions des différents sens pour construire une image mentale unifiée de l’environnement. Il a toutefois fallu attendre l’émergence de l’informatique pour transposer aux capteurs artificiels la capacité naturelle des êtres vivants à fusionner des informations. Actuellement, la profusion des capteurs, l’amélioration des processseurs et les progrès en modélisation formelle tendent à favoriser un usage de plus en plus répandu de la fusion de données.

Historiquement, les premières techniques de fusion de données ont été développées dans le cadre d’applications militaires [31]. Dans ce domaine, nous pouvons citer la détection et le suivi d’intrus, la surveillance de champs de bataille ou la détection de mines. Depuis lors, l’usage de la fusion s’est largement répandu dans d’autres domaines. Parmi les nouveaux champs d’applications, citons la télédétection, l’aide aux diagnostiques médicaux, la robotique ou encore l’assistance aux personnes et aux opérateurs [32][8] et [76]. De manière générale, la fusion de données s’est développée pour répondre à un besoin de synthèse et de fiabilité de l’information, car dans nombre d’applications, la quantité des données à manipuler devient de plus en plus importante et difficile à appréhender par des opérateurs humains.

1.2 Systèmes multi capteurs

Les objectifs et les données traitées par l’ensemble de la communauté ayant recours à la fusion sont de natures très différentes. Les techniques mises en œuvre sont par conséquent assez hétérogènes et il n’existe pas de formalisme faisant l’unanimité de la discipline. Certaines caractéristiques sont toutefois communes à bon nombre d’applications.

En règle générale, les mécanismes de fusion de données cherchent à estimer une grandeur, une information ou un état inconnu à partir d’une ou plusieurs sources d’informations imparfaites. Tout comme en théorie de la décision, la variable \( x \) que l’on cherche à estimer est d’ordinaire appelée l’état [7][21]. Cette quantité peut aussi bien être une valeur numérique qu’un processus ou une assertion. Le cadre de discernement \( X \) désigne alors l’ensemble des états admissibles et mutuellement exclusifs (\( x \in X \)).

La fusion de données propose d’estimer cet état inconnu à partir d’observations \( z \in Z \). Il est donc nécessaire de disposer d’un modèle \( F(x) \) explicitant la relation entre l’état inconnu \( x \) et les observations \( z \). Notons que ce modèle comprend généralement une composante aléatoire représentant le bruit de mesure, si bien que malgré la redondance des observations, deux états différents peuvent donner lieu à une même observation. C’est cette ambiguïté que va tenter de lever la fusion de
Systèmes multi capteurs

D’un point de vue formel, l’objectif de la fusion de données consiste alors à inférer l’état $x$ d’un système à partir d’observations $z$. Cette inférence est réalisée en établissant une fonction de décision de la forme $\delta(z) \rightarrow x$ qui, outre le modèle d’observation, peut également prendre en compte l’incertitude des mesures ainsi que des connaissances a priori de l’état. La définition de ce mécanisme décisionnel constitue le cœur de la fusion de données.

Dans le suite de ce travail, nous allons tout particulièremment nous intéresser à la fusion de données appliquée aux systèmes de détection multi capteurs. Dans ce contexte spécifique, précisons le sens donné aux termes suivants [33].

Cible

C’est l’objet ou l’intrus dont la fusion cherche à détecter la présence. Le cas échéant, les objectifs visés par le système vont concerner le nombre de cibles (dénombrement), leurs positions respectives (localisations), le suivi des déplacement (pistage) ainsi qu’éventuellement l’estimation de certaines caractéristiques (identifications).

Capteur

Dans le cadre de ce travail, un capteur sera défini comme une combinaison de moyens matériels et logiciels formant un système capable de détecter la présence de cibles. Selon leur nature, les capteurs auront des caractéristiques différentes. Certains capteurs seront par exemple capables de dénombrer les cibles, d’estimer la distance ou la direction de leurs positions ou encore de fournir des informations d’identification. Notons qu’afin d’être moins vulnérables face aux leurres et aux conditions climatiques, la plupart des systèmes de surveillance sont constitués de capteurs de natures différentes.

Scène

Il s’agit de la région spatiale à l’intérieur de laquelle la fusion va chercher à inférer le nombre et la localisation des cibles. Cette tâche est réalisée au moyen de capteurs mais il faut préciser que la totalité de la scène ne sera pas nécessairement couverte par ceux-ci.

Détection

C’est l’information provenant d’un capteur signalant la présence d’une ou plusieurs cibles dans la scène à un moment donné. Selon la nature des capteurs, la précision et la fiabilité des détections peuvent varier. Une détection qui ne correspond à aucune cible, sera appelée fausse détection et si au contraire, une cible n’est pas détectée lorsqu’elle se trouve dans une zone surveillée, on parlera de non détection.
Introduction générale

Etat d'une cible

Il s'agit de la position et de l'identification d'une cible à un moment donné. La fusion a donc pour objet l'estimation de cet état à partir des détections provenant des capteurs et éventuellement des estimations des états antérieurs. Par abus de langage, l'état pourra également désigner son estimation inférée par la fusion.

Piste

Une piste est un historique des estimations successives de l'état d'une cible au cours du temps.

Localisation

La localisation est l'estimation de la position d'une cible. Le résultat de cette tâche, produit de la fusion, est un des paramètres clés constituant l'état d'une piste. Dans ce travail, le cas tridimensionnel ne sera pas abordé. La localisation sera donc plane.

Identification

D'une manière générale, l'identification consiste à déterminer la nature d'une cible. Selon le contexte, il peut s'agir de l'évaluation d'une valeur statique (par exemple, la classe d'une piste) ou dynamique (par exemple, l'estimation de la vitesse ou du danger). Tout comme la localisation, l'identification fait partie de l'état d'une piste.

1.3 Contexte applicatif

L'application que nous proposons d'étudier concerne la surveillance de zone. Plus précisément, le système auquel nous allons appliquer la fusion de données est constitué d'un ensemble de capteurs détectant de manière indépendante la présence de cibles sur une zone d'intérêt (la scène). Le système est composé de capteurs de natures différentes et chacun de ceux-ci ne couvre généralement qu'une partie de la scène.

Les détections fournissent des informations sur les localisations possibles de cibles. Elles sont toutefois entachées d'une imprécision variable qui dépend non seulement des propriétés des objets détectés (par exemple, leur couleur ou leur taille), mais aussi des spécificités de la scène tel que les éléments masquants ou les conditions météorologiques, et des caractéristiques intrinsèques aux capteurs. Certains capteurs ne fournissent en effet aucune information sur la position relative de la cible alors que d'autres sont capables d'évaluer la direction ou la distance des objets.

Ces détections hétérogènes sont transmises à fréquence régulière à la fusion qui produit une représentation unifiée à partir de toutes les informations disponibles. Les principaux résultats attendus sont l'évaluation du nombre total de cibles, l'affinement de leur localisation par recoupement d'informations, leur suivi par pistage dans
la zone surveillée et l’agrégation cohérente des informations d’identification provenant des différents capteurs.

1.4 Objectifs de la fusion de données

Comme nous l’avons déjà mentionné, les objectifs auxquels la fusion de données va tenter de répondre concerneront le dénombrement des intrus, leurs localisations respectives ainsi que leur classification ou l’identification de leurs caractéristiques. L’intérêt qui en découle est quadruple.

1.4.1 Suivi des cibles

Le premier objectif de la fusion est le pistage des cibles. Il s’agit d’associer les détections provenant des différents capteurs et qui se rapportent à un même objet physique.

Cette association permet de confirmer une détection incertaine et d’y adjoindre des informations plus précises. Les informations fournies par différents capteurs mais relatives à la même cible pourront dès lors être combinées.

La connaissance des trajectoires des cibles est une information intéressante pour une analyse de plus haut niveau. Elle permet par exemple d’identifier des comportements anormaux, ce qui est particulièrement utile pour la surveillance de sites ouverts.

1.4.2 Synthèse et précision de l’information

La fusion d’informations partiellement redondantes induit naturellement une synthèse dans la mesure où le traitement réduit la quantité de données. Idéalement, la fusion ambitionne de ne pas générer de pertes d’informations utiles. Aussi, le terme *synthèse* a-t-il été préféré à celui de *compression*.

L’objectif poursuivi par la synthèse est de produire un état plus précis que chacune des détections isolées. Il ne s’agit donc pas seulement d’une suppression des redondances mais aussi d’une optimisation par recoupement d’informations.

1.4.3 Réduction des fausses alarmes

Si les capteurs du système de détection génèrent de fausses détections et que ce manque de fiabilité est pris en compte dans la fusion, nous pouvons espérer réduire le nombre de fausses alarmes. Le principe général est alors d’adopter une stratégie prudente en attendant la confirmation d’une détection incertaine par plusieurs capteurs avant d’inérer l’existence d’une cible.

Notons que dans la plupart des applications, la notion de fausses alarmes n’est pas la même du point de vue de l’utilisateur et des capteurs. Par exemple, un système de surveillance de site vise généralement à détecter la présence d’intrus humains dans une zone. La détection d’un animal peut donc, de ce point de vue, être considérée comme une fausse alarme. Cependant, un capteur ne peut pas présumer de cet
objectif. S’il n’est pas doté de fonctions d’identifications, il ne peut (et ne doit) pas distinguer ces deux cibles. De son point de vue, il ne s’agit donc pas d’une fausse détection.

Le filtrage de ces fausses alarmes subjectives dépend de l’application. Il est le plus souvent réalisé au moyen d’un système expert qui génère des alarmes à partir des caractéristiques des cibles tout en s’appuyant sur des règles spécifiques. La fusion contribue à la suppression de ce deuxième type de fausses alarmes en fournissant au système expert l’ensemble des informations pertinentes relatives à chaque piste.

1.4.4 Détection du dysfonctionnement de capteurs

Grâce à la fusion de données, nous pouvons enfin envisager de mettre en évidence certains dysfonctionnements de capteurs. Par exemple, si un capteur est systématiquement en désaccord avec les autres capteurs du système, nous pouvons nous interroger sur sa fiabilité.

De façon plus générale, l’étude de la redondance attendue permet de valider le bon paramétrage des capteurs. Si besoin, des informations peuvent alors être retournées afin de corriger la configuration du détecteur. Cet aspect ne sera toutefois pas abordé dans ce travail.

1.5 Taxonomie et organisation fonctionnelle

L’ensemble des techniques relatives à la fusion de données est usuellement divisée en quatre catégories se distinguant essentiellement par le niveau d’abstraction des données manipulées. Cette subdivision a initialement été introduite par le Joint Directors of Laboratories (JDL) lors d’une tentative de standardisation de la terminologie utilisée par différents laboratoires de l’armée. Il s’agit d’un modèle de type hiérarchique et fonctionnel qui appréhende la fusion en terme de couches : signal, menace et analyse de la situation. Ce découpage relativement arbitraire et spécifiquement adapté aux applications militaires n’a depuis lors pas été adopté par l’ensemble de la communauté civile. D’autres cadres d’analyse ont donc été proposés [39][48], mais le travail du JDL a le mérite d’être le premier du genre [75][77] et [72].

1.5.1 Hiérarchie des niveaux d’abstraction

Le premier niveau du modèle conceptuel du JDL regroupe les opérations sur les détections. Il concerne tous les traitements permettant d’extraire, d’identifier et de caractériser dans un référentiel commun les objets d’intérêt. Le deuxième niveau concerne la description et l’analyse des interactions entre ces objets. L’ensemble des traitements numériques des données est donc réuni dans ces deux premiers niveaux. Les troisième et quatrième niveaux exploitent les données provenant des traitements précédents afin d’en extraire des informations plus abstraites ou qualitatives. Les outils d’aide à la décision se situent à ces niveaux.

Dans cette étude, nous nous focaliserons sur des méthodes de fusion de niveau un et deux, pour répondre à la question suivante : “Étant données les détections
partiellement incertaines et imprécises provenant des capteurs, combien y a-t-il de cibles dans la scène, où sont-elles localisées et de quel type sont-elles ?”.

1.5.2 Découpage en blocs fonctionnels

L’estimation dynamique de l’état des pistes est généralement réalisée au moyen d’un filtrage récursif de type prédiction-correction. L’architecture de tels systèmes dépend alors de la gestion synchrone ou asynchrone des données, de l’organisation centralisée, distribuée ou décentralisée de la fusion ou encore d’éventuelles contraintes liées aux communications. Cependant, les principaux blocs fonctionnels présentés sur la figure 1.1 et détaillés dans cette section sont presque toujours identiques [32].

Prétraitements des détections

Les détections transmises à la fusion sont entachées d’une part d’incertitude. Cette dernière dépend de la nature du capteur, des conditions de détection ou des caractéristiques des cibles. Avant la fusion des données, un prétraitement est donc nécessaire pour représenter les détections et leurs incertitudes respectives de manière homogène. L’approche classique, de type probabiliste, consiste à représenter cette incertitude par une ellipse dont la taille varie en fonction de la qualité et de la fréquence des détections. Il existe bien entendu des alternatives à cette approche, dont la solution qui sera proposée dans ce travail. Soulignons cependant que les ellipses d’incertitude constituent incontestablement un standard largement utilisé.

Alignement des données

Lorsque le système est composé de plusieurs capteurs, chacun d’eux opère en général dans un repère relatif qui lui est propre. Avant de combiner les détections, il faut donc les représenter dans un référentiel commun. Si cette transformation est connue, autrement dit si les capteurs sont calibrés, ce traitement peut s’avérer une
tâche relativement simple. Cette dernière est toutefois la source d’une imprécision supplémentaire sur les données.

Représentation des états de pistes

L’estimation des états des pistes est construite par combinaison des détecteurs et de l’état antérieur de la piste. La représentation d’un état et de son incertitude dépend donc de la théorie adoptée pour fusionner les détecteurs.

Prédiction

Avant de combiner l’état d’une piste existante avec une ou plusieurs nouvelles détecteurs, il est nécessaire d’en prédire la position ainsi que l’incertitude associée. Cette étape de prédiction permet de synchroniser les données et de ne pas sous-estimer l’incertitude de localisation de la piste. En fonction de la fréquence des observations et de la nature des cibles observées, la prise en compte d’un modèle de déplacement peut améliorer significativement la qualité de cette prédiction.

Mise à jour

L’état des pistes est mis à jour en combinant les nouvelles informations disponibles. Cette étape va généralement de pair avec la prédiction. Elles forment alors à elles deux un filtre récursif alternant les étapes de prédiction et de correction.

Association

Lorsque plusieurs pistes sont simultanément présentes dans la scène, l’étape d’association entre les détections et les pistes permet de déterminer à quelle piste se rapporte une nouvelle détection avant d’effectuer la mise à jour.

Gestion des pistes

Ce module est chargé de supprimer les pistes qui ne sont plus détectées ou qui sortent de la scène. Il gère également la création de nouvelles pistes lorsque certaines détections n’ont pu être associées à une piste existante.

1.6 Évolution des architectures

Les systèmes de surveillance exploitent habituellement des capteurs physiquement répartis sur l’ensemble de la scène. Une organisation centralisée des traitements impose que toutes ces données soient rapatriées vers un seul processeur. Celui-ci se charge alors de représenter et de fusionner toutes les détections dans un référentiel commun.

Bien que cette architecture soit pertinente pour nombre d’applications, la taille croissante des systèmes de fusion et la baisse de coût des calculateurs embarqués
encouragent le développement d’architectures alternatives réalisant une part de plus en plus importante des traitements en dehors de ce processeur central.

Un système est dit distribué à partir du moment où chaque détecteur est équipé d’un processeur qui traite les données brutes avant de transmettre des informations. En considérant le système de surveillance dans son ensemble, sans distinguer les tâches de détection et de fusion, des capteurs constitués de caméras et de logiciels de détection embarqués constituent donc déjà une forme d’architecture distribuée.

Cette organisation ne constitue que la forme élémentaire d’un traitement réparti et il est clair qu’une part plus importante des calculs peut être déportée hors de l’unité centrale. Parmi les nombreuses solutions existantes, trois types d’architectures sont souvent cités dans la littérature. Ils se distinguent par une répartition de plus en plus marquée des traitements.

1.6.1 Architectures hiérarchiques

L’approche la plus naturelle pour organiser le traitement des données de manière répartie consiste à diviser la tâche en plusieurs niveaux hiérarchiques. Selon ce principe illustré sur la figure 1.2, une fusion locale est réalisée au niveau de chaque capteur. Les pistes résultantes, constituées des détections provenant d’un seul détecteur, sont ensuite envoyées au processeur central. Celui-ci fusionne alors les données collectées afin de constituer des pistes regroupant de manière cohérente toutes les informations [21].

![Architecture hiérarchique](image)

Outre la réduction de la charge imposée au processeur central, la répartition du pistage permet également de mieux gérer les problèmes de réseau. Cet argument est important car les difficultés liées aux délais et aux pertes de données lors des communications sont parmi les plus critiques dans beaucoup de cas pratiques.

En dépit de leurs avantages indéniables, les architectures hiérarchiques souffrent de plusieurs inconvénients. Tout d’abord, la charge de calcul imposée au processeur central reste importante et proportionnelle à la taille du système. De plus, le système est entièrement dépendant du bon fonctionnement de l’unité centrale.
1.6.2 Architectures distribuées

La volonté de faire évoluer les systèmes vers des solutions de plus en plus réparties a d’abord été motivée par un besoin croissant de flexibilité et de modularité. Pour répondre à ces nouvelles exigences, les premières solutions proposées ont directement été inspirées des travaux réalisés dans le domaine de l’intelligence artificielle.

Dans cette discipline, l’architecture Blackboard (cf. figure 1.3) est une de celles qui ont rencontré le plus de succès [36][57]. Le système est constitué d’agents autonomes représentant une source d’informations ou une ressource de traitement. Tous ces agents échangent des données au moyen d’un espace partagé fonctionnant comme un tableau noir. Chaque agent peut ainsi écrire ou lire des données sur ce support sans contrainte de formalisme ou de contenu.

Cette approche est donc très souple et en principe n’importe quel agent peut être supprimé ou ajouté. Cependant, bien qu’aucun agent ne s’impose en maître, le tableau noir (Blackboard) occupe une position centrale et constitue donc en pratique un point faible et un obstacle à la croissance des systèmes.

1.6.3 Architectures décentralisées

Un système de fusion de données décentralisé est un réseau de nœuds constitués de capteurs et de processeurs qui ne nécessitent l’utilisation d’aucune ressource centrale. La fusion est alors réalisée sur chaque nœud, à partir des observations locales et des informations communiquées par les voisins. Un tel système, dont l’organisation générale est illustrée sur la figure 1.4, est alors caractérisé par trois contraintes [22] :

- aucun composant n’occupe de position centrale,
- les communications se font uniquement entre deux nœuds, sans que l’information ne soit visible par tous les agents,
- aucun nœud ne connait la topologie complète du réseau. Ils ne peuvent donc communiquer qu’avec leurs voisins.
Lorsque ces trois contraintes sont respectées, les systèmes décentralisés bénéficient de trois caractéristiques qui en font leur intérêt :

- ce sont des systèmes scalables. Aucun goulot d’étranglement ne limite en théorie leur taille,
- aucun composant n’est nécessaire au fonctionnement des autres éléments. Par conséquent, aucun d’eux ne constitue un point faible dont l’arrêt paralyse le reste du système,
- aucune connaissance globale n’est nécessaire et tous les traitements sont réalisés localement. Le système est donc modulaire et le nombre comme la nature des capteurs peuvent être modifiés en fonction des besoins.

Ces spécificités assurent aux approches décentralisées un avantage considérable, notamment dans le contexte des applications de surveillance. Cette tendance est toutefois relativement récente et les systèmes opérationnels fidèles à ces recommandations font encore l’objet de recherches.

1.7 Motivations pour de nouvelles approches


1.7.1 Limitations du filtre de Kalman

Bien que le filtre de Kalman *classique* soit en théorie optimal, il souffre de plusieurs inconvénients qui rendent sa mise en œuvre délicate dans notre cas d’étude.
Introduction générale

Incertitude gaussienne

Les observations $z$ sont supposées être des mesures entachées d’un bruit blanc gaussien. Or, en pratique, le bruit d’observation est non seulement très souvent corréle dans le temps, mais l’hypothèse gaussienne est également souvent contestable. Beaucoup de capteurs sont par exemple capables de renseigner la direction d’une cible sans pouvoir estimer la distance. D’autres détecteurs ne peuvent fournir aucune information de localisation plus précise que la portée du capteur. Dans ces conditions il semble hasardeux de caractériser l’incertitude de localisation par une gaussienne.

Modèle de transition linéaire

La mise en œuvre d’un filtrage de Kalman suppose la connaissance de la matrice de transition $F$ traduisant l’évolution d’état entre deux observations successives. En fonction du type de cibles surveillées et de la fréquence des observations, ce modèle de transition peut se résumer à une matrice de bruit. Dans ce cas, l’incertitude de localisation est entièrement déterminée par l’incertitude des observations et l’intérêt du filtrage devient limité.

Informations contextuelles

Le modèle d’incertitude adopté par le filtre de Kalman rend également difficile la prise en compte d’informations contextuelles telles que la connaissance d’éléments topographiques. Ces données ont pourtant intérêt à être exploitées pour améliorer le pistage. Il y a, par exemple, peu de chance qu’un piéton passe à travers les murs d’un bâtiment et inversément, il est probable qu’un véhicule se déplace en suivant le tracé de la route.

1.7.2 Alternatives

Il existe bien entendu plusieurs alternatives au filtrage de Kalman. Parmi celles-ci, nous pouvons citer les filtres particulaires [2], l’analyse par intervalles [53] [28], les grilles probabilistes [9][24], les approches possibilistes [80][19] ou encore crédibiliste [1].

Dans ce travail, nous proposons de représenter l’état des pistes ainsi que les détections au moyen de fonctions de croyance. Le choix de cette représentation non probabiliste est motivé par la possibilité de distinguer explicitement l’incertitude et l’imprécision dans la représentation de la localisation. Cette représentation très riche de la connaissance de l’état des pistes nous permettra en outre de manipuler et d’interpréter des informations partiellement conflictuelles.

1.8 Organisation du document

La suite de ce document est organisée comme suit. Le premier chapitre est consacrée à une brève présentation du modèle des croyances transférables ($TBM$) qui
constitue le cadre théorique que nous avons choisi pour représenter les pistes et fusionner les informations. Le problème du pistage multi capteurs sera ensuite reconsidéré dans le contexte spécifique du TBM au cours du deuxième chapitre. À la fin de ce chapitre, la représentation crédibiliste (BBA) des détections et des pistes ainsi que toutes les opérations nécessaires pour le suivi multi capteurs d’une piste auront été présentées. L’association entre détections et pistes, étape clé lors du suivi simultané de plusieurs pistes sera abordée dans le troisième chapitre et la simplification des BBAs, opération particulièrement critique dans notre cas sera étudiée dans le quatrième chapitre. Le cinquième chapitre sera consacré à l’élaboration d’une stratégie originale permettant de réaliser les traitements de fusion selon une architecture décentralisée. Enfin, le démonstrateur d’un système de détections décentralisé, dont le module de fusion a été réalisé au cours de ce travail, ainsi que quelques essais effectués sur le terrain seront présentés dans les deux derniers chapitres.
Chapitre 2

Le modèle des croyances transférables (TBM)

Sommaire

2.1 Introduction ........................................... 20
2.2 Un modèle non probabiliste ............................ 20
   2.2.1 Fonctions de masse (BBAs) ..................... 20
   2.2.2 Croyance, implicabilité, plausibilité et communalité . . . . 22
2.3 Combinaisons d’informations ........................... 23
   2.3.1 Principe de moindre engagement .................. 24
   2.3.2 Conditionnement .................................... 25
   2.3.3 Combinaisons ....................................... 26
   2.3.4 Fiabilité des sources et affaiblissement ............ 28
   2.3.5 Modifications du cadre de discernement ............. 28
2.4 Décomposition canonique .............................. 30
   2.4.1 Décomposition canonique conjonctive ............... 30
   2.4.2 w ordonnancement .................................. 32
   2.4.3 Règle conjonctive prudente ....................... 32
   2.4.4 Règle disjonctive hardie .......................... 33
2.5 Prise de décision ...................................... 34
   2.5.1 Transformation pignistique ....................... 35
   2.5.2 Minimisation du risque ........................... 36
   2.5.3 Décisions alternatives ............................ 36
2.6 Comparaison de BBAs .................................. 37
   2.6.1 Similarité .......................................... 37
   2.6.2 Conflit ............................................. 39
2.7 Interprétations de la théorie des croyances .............. 39
2.1 Introduction

L’exploitation de connaissances partielles lors d’une prise de décision a été abordée dans de nombreuses théories. Parmi celles-ci, la théorie des fonctions de croyance (Tranferable Belief Model : TBM) propose un formalisme permettant de distinguer ce qui est incertain de ce qui est imprécis. Cette approche développée à partir des travaux de Dempster [11] et Shafer [64], a initialement été introduite sous le nom de théorie des preuves (evidence theory). La démarche a ensuite été reprise et réinterprétée par Smets dans la théorie des croyances transférables [70]. Cette dernière, bien que gardant la même représentation à l’aide de fonctions de croyance, se distingue par la partie dynamique (combinatoire) et par son interprétation non probabiliste.

Avant d’étudier l’application de cette théorie aux systèmes multi capteurs, nous allons en faire une brève présentation au cours de ce chapitre. L’objectif n’est pas d’en faire une description exhaustive, mais d’exposer les principes fondamentaux et d’introduire les notions utilisées dans la suite du travail.

2.2 Un modèle non probabiliste

Lors de l’étude d’un phénomène aléatoire dans le cadre de la théorie bayésienne classique, les probabilités combinées doivent être connues pour chacun des événements que l’on cherche à discriminer. À défaut de mieux, il faut donc considérer a priori que deux hypothèses incompatibles dont on ne connaît que la somme des probabilités, sont équiprobables. Partant du constat de l’aspect arbitraire de cette répartition équiprobable, la théorie des fonctions de croyance (TBM) propose d’attribuer des croyances non pas à des singletons, mais à des sous-ensembles d’hypothèses. L’idée sous-jacente est que si une source d’information considère indistinctement un regroupement d’hypothèses, il n’y a aucune raison de diviser arbitrairement la confiance accordée à cet ensemble.

2.2.1 Fonctions de masse (BBAs)

Le cadre d’étude sur lequel la théorie des fonctions de croyance représente un état incertain est appelé le cadre de discernement.

Définition 2.1. Cadre de discernement

Le cadre de discernement Ω est l’ensemble des hypothèses ωi mutuellement exclusives envisagées comme solution d’un problème posé, de sorte que :

Ω = {ω0, ...ωN}  (2.1)

Le cadre de discernement peut, en toute généralité, ne pas contenir toutes les solutions possibles. Dans ce cas, on admet que la solution peut ne pas avoir été envisagée. C’est l’hypothèse de monde ouvert dont l’hypothèse contraire, celle d’un monde fermé peut être considérée comme un cas particulier.
Un modèle non probabiliste

L’ensemble des parties de \( \Omega \) est désigné par \( 2^\Omega \). Cette notation quelque peu abusive souligne le fait que si \( \Omega \) contient \( N \) singletons (\( |\Omega| = N \)), il existe alors \( 2^N \) parties différentes de \( \Omega \).

\[
2^\Omega = \{ A | A \subseteq \Omega \} \tag{2.2}
\]

La théorie des fonctions de croyance propose alors d’alloquer des *parts de croyance* non pas uniquement aux singletons \( \omega_i \) (comme dans la théorie des probabilités) mais à n’importe quel ensemble de \( 2^\Omega \). Cette croyance peut notamment être représentée par une *fonction de masse*.

**Définition 2.2. Fonction de masse**

Une fonction de masse \( m^\Omega \) est une fonction de \( 2^\Omega \) vers \([0, 1]\) définie par :

\[
m^\Omega : 2^\Omega \to [0, 1] \\
\sum_{A \subseteq \Omega} m^\Omega(A) = 1
\]

La masse \( m^\Omega(A) \) représente la part de croyance allouée indistinctement à l’ensemble des hypothèses incluses dans \( A \). La distinction entre ces dernières ne pourra se faire qu’à l’aide d’informations supplémentaires. Afin d’alléger la notation, l’exposant indiquant le cadre de discernement ne sera mentionné que s’il y a ambiguïté. Ainsi, lorsqu’un seul cadre de discernement \( \Omega \) est considéré, \( m^\Omega(A) \) peut s’écritre \( m(A) \).

La distribution des masses modélisant une proposition est appelée une *BBA* (*Basic Belief Assignment*) et tout ensemble \( A \subseteq \Omega \) tel que \( m(A) > 0 \) est un élément focal de cette BBA.

Lorsqu’ils existent, deux éléments focaux méritent une attention particulière : \( m(\Omega) \) représente le degré d’ignorance globale et \( m(\emptyset) \) est appelé le conflit. Ce conflit, inexistant sous hypothèse d’un monde fermé (approche de Dempster-Shafer), peut être interprété comme une croyance en faveur d’une hypothèse en dehors du cadre de discernement ou comme le signe d’un dysfonctionnement soit du modèle, soit d’une des sources.

Plusieurs BBAs bénéficient également de dénominations spécifiques.

**Définition 2.3. BBA normale**

Une BBA est dite normale si l’ensemble vide n’est pas un élément focal (\( m(\emptyset) = 0 \)). Dans le cas contraire, la BBA est dite sous-normale.

**Définition 2.4. BBA catégorique**

Une BBA catégorique est une BBA n’ayant qu’un seul élément focal \( A \subseteq \Omega \). Cela implique que \( m(A) = 1 \) et \( m(B) = 0 \), \( \forall B \subseteq \Omega, B \neq A \). Une telle fonction est notée \( m^\Omega_A \).

**Définition 2.5. BBA vide**

Une BBA vide est une BBA catégorique pour laquelle l’élément focal est \( \Omega \). Par conséquent, cette fonction vérifie : \( m(\Omega) = 1 \).
Définition 2.6. **BBA dogmatique**

Une BBA dogmatique est caractérisée par le fait que $\Omega$ n’est pas un élément focal ($m(\Omega) = 0$).

Définition 2.7. **BBA simple**

Une BBA est dite simple si elle contient au plus deux éléments focaux, dont $\Omega$.

Définition 2.8. **BBA bayésienne**

Une BBA bayésienne est une BBA dont les éléments focaux sont tous des singletons.

Définition 2.9. **BBA consonante**

Une BBA est dite consonante si tous les éléments focaux sont emboîtés. Par conséquent, si $m(A) \neq 0$ et $m(B) \neq 0$ alors $A \subseteq B$ ou $B \subseteq A$, $\forall A, B \subseteq \Omega$. Cette spécificité implique que l’opérateur d’inclusion définit une relation d’ordre total sur les éléments focaux de la BBA consonante.

### 2.2.2 Croyance, implicabilité, plausibilité et communalité

Outre la fonction de masse $m$, il existe d’autres fonctions permettant de représenter la répartition de la croyance sur $\Omega$. Il s’agit de la crédibilité $\text{bel}$, de l’implicabilité $b$, de la plausibilité $pl$ et de la communalité $q$. Les fonctions $m, \text{bel}, b, pl$ et $q$ sont des fonctions biunivoques et il est possible de passer de l’une à l’autre par transformation bijective. Ces cinq représentations sont donc équivalentes mais mettent en évidence des aspects différents de la même structure de croyance.

Ces nouvelles fonctions, de $2^\Omega$ vers $[0,1]$, peuvent être définies à partir de la fonction de masse.

Définition 2.10. **Crédibilité**

La crédibilité $\text{bel}$ d’un ensemble $A$ est définie comme la somme des masses de tous les sous-ensembles de $A$ différents de $\emptyset$ :

$$\text{bel}(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B), \quad \forall A \subseteq \Omega \quad (2.3)$$

$\text{bel}(A)$ peut être interprétée comme le degré de croyance accordé spécifiquement à $A$ plutôt qu’à son complémentaire $\bar{A}$.

Définition 2.11. **Implicabilité**

L’implicabilité $b$ d’un ensemble $A$ est définie comme la somme de tous les sous-ensembles de $A$ ($\emptyset$ compris) :

$$b(A) = \sum_{B \subseteq A} m(B), \quad \forall A \subseteq \Omega \quad (2.4)$$

de sorte que $b$ est égale à la crédibilité à laquelle est ajoutée la masse de l’ensemble vide $m(\emptyset)$ :

$$b(A) = \text{bel}(A) + m(\emptyset), \quad \forall A \subseteq \Omega \quad (2.5)$$
Définition 2.12. Plausibilité

La plausibilité $pl$ d’un ensemble $A$ est la somme des masses de tous les ensembles d’intersection non nulle avec $A$ :

$$pl(A) = \sum_{A \cap B \neq \emptyset} m(B), \quad \forall A \subseteq \Omega$$ (2.6)

La plausibilité peut être vue comme la part de croyance compatible avec $A$. En d’autres termes, il s’agit de la somme des masses des sous-ensembles de $\Omega$ qui ne sont pas en contradiction avec $A$.

Définition 2.13. Communalité

La communalité $q$ d’un ensemble $A$ est définie comme la somme des masses des ensembles dans lesquels $A$ est inclus :

$$q(A) = \sum_{B \supseteq A} m(B), \quad \forall A \subseteq \Omega$$ (2.7)

Il s’agit du degré d’ignorance après conditionnement par $A$.

Comme cela a déjà été mentionné, chacune des représentations peut être définie à partir d’une autre. Par exemple :

$$m(A) = \sum_{B \subseteq A} (-1)^{|A| - |B|} pl(\overline{B}), \quad \forall A \subseteq \Omega$$ (2.8)

$$m(A) = \sum_{B \supseteq A} (-1)^{|B| - |A|} q(B), \quad \forall A \subseteq \Omega$$ (2.9)

$$m(A) = \sum_{B \subseteq A} (-1)^{|A| - |B|} b(B), \quad \forall A \subseteq \Omega$$ (2.10)

$$pl(A) = bel(\Omega) - bel(\overline{A}) = 1 - b(\overline{A}), \quad \forall A \subseteq \Omega$$ (2.11)

2.3 Combinaisons d’informations

Le formalisme introduit dans la section précédente permet de représenter des croyances provenant de différentes sources d’informations. Ces structures constituent la partie statique du modèle. Elles doivent à présent être combinées afin de construire une vue synthétique de l’ensemble des données. Il s’agit de la partie dynamique du modèle. De manière un peu plus générale, la partie dynamique regroupe l’ensemble des manipulations de BBA s.
2.3.1 Principe de moindre engagement

Les fonctions de croyance permettent de laisser de la masse sur des unions d’hypothèses singletons tant que l’on ignore comment la répartir. Ainsi, l’information contenue dans une BBA vide

\[ m(\Omega) = 1 \]

est très différente de l’information contenue dans une BBA bayésienne équiprobable

\[ m(\omega_i) = \frac{1}{|\Omega|} \]

Cette distinction, fondamentale dans la théorie des fonctions de croyance, justifie l’introduction du principe de moindre engagement [70] [71].

Définition 2.14. Principe de moindre engagement

Lorsque deux fonctions de croyance sont compatibles avec l’ensemble des contraintes, il faut privilégier celle qui est la moins engagée.

Le respect de ce principe permet d’éviter d’ajouter des informations non contenues dans les contraintes et donc arbitraires. Son application suppose toutefois l’existence d’une relation d’ordre, au moins partielle, permettant de comparer la quantité d’information (l’engagement) des BBAs. Plusieurs relations d’ordre partiel permettant de comparer certaines fonctions de croyance définies sur un même cadre de discernement \( \Omega \) ont été proposées à cette fin.

Définition 2.15. pl-ordonnancement

Une fonction de croyance \( m_1 \) est plus engagée que \( m_2 \) au sens de pl si et seulement si la plausibilité de \( m_1 \) est inférieure à la plausibilité de \( m_2 \) sur tous les sous-ensembles de \( \Omega \).

\[ m_1 \sqsubseteq_{pl} m_2 \quad \text{ssi} \quad pl_1(A) \leq pl_2(A), \ \forall A \subseteq \Omega \]

(2.14)

Définition 2.16. q-ordonnancement

Une fonction de croyance \( m_1 \) est plus engagée que \( m_2 \) au sens de q si et seulement si la communalité de \( m_1 \) est inférieure à la communalité de \( m_2 \) sur tous les sous-ensembles de \( \Omega \).

\[ m_1 \sqsubseteq q m_2 \quad \text{ssi} \quad q_1(A) \leq q_2(A), \ \forall A \subseteq \Omega \]

(2.15)

Définition 2.17. S-ordonnancement

Une fonction de croyance \( m_1 \) est plus engagée que \( m_2 \) au sens de s si et seulement si \( m_1 \) peut être construite en répartissant la masse de tous les éléments focaux \( B \) de \( m_2 \) sur des sous-ensembles \( A \) de \( B \).

\[ m_1 \sqsubseteq_S m_2 \quad \text{ssi} \quad \exists S \text{ tq } S(A, B) > 0 \Rightarrow A \subseteq B, \sum_A S(A, B) = 1 \ \forall A \subseteq \Omega \]

(2.16)
Dans ce cas $m_1$ est une spécialisation de $m_2$. Chacun des termes non nul de cette matrice de transformation $S(A, B)$ représente la part de $m_2(B)$ qui est transférée sur $m_1(A)$. Étant donné que la transformation respecte la propriété $S(A, B) > 0 \Rightarrow A \subseteq B$, la masse des éléments focaux est répartie sur des sous-ensembles inclus dans ceux-ci. Il s’agit donc bien d’une spécialisation au sens intuitif du terme. À l’inverse, $m_2$ est une généralisation de $m_1$.

Bien que ces relations d’ordre ne permettent pas de comparer deux fonctions de croyances quelconques, elles expriment bien la notion intuitive d’engagement.

Notons également que toutes ces relations d’ordre partiel acceptent la BBA vide comme unique plus grand élément. Elles sont en outre liées par la relation :

$$m_1 \sqsubseteq_S m_2 \Rightarrow \begin{cases} m_1 \sqsubseteq_{pl} m_2 \\ m_1 \sqsubseteq_q m_2 \end{cases}$$ (2.17)

### 2.3.2 Conditionnement

L’opération de conditionnement correspond au raisonnement suivant. Supposons qu’au cours de notre analyse, nous disposions d’une BBA définie sur le cadre de discernement $\Omega$. Si par ailleurs une nouvelle information nous permet de savoir de manière incontestable que la solution de notre problème (l’hypothèse vraie) ne se trouve pas dans $\bar{A}$, quelle transformation faut-il appliquer à la BBA pour tenir compte de cette nouvelle information ?

La transformation appelée conditionnement $m^\Omega[A]$ (ou lorsque le contexte l’autorise $m[A]$) peut être justifiée à l’aide du principe de moindre engagement appliqué aux spécialisations [69]. Le conditionnement est alors défini à partir des axiomes suivants :

- $m[A]$ est une spécialisation de $m$. Cette contrainte, bien que pouvant sembler arbitraire, est toutefois assez intuitive. Le conditionnement sur $A$ est en effet une nouvelle donnée qui ne doit ni remettre en cause ni contredire les autres sources d’informations. La transformation doit donc se traduire par un transfert de masses vers des sous-ensembles. Il s’agit bien d’une spécialisation.
- Après conditionnement sur $A$, le complémentaire $\bar{A}$ ne peut plus être plausible. Autrement dit :

$$pl[A](\bar{A}) = 0$$ (2.18)

Cette propriété traduit le fait que si la solution se trouve avec certitude dans $A$, elle ne se trouve nécessairement pas dans $\bar{A}$. Ce principe est justifié par le fait que les hypothèses $A$ et $\bar{A}$ sont par définition mutuellement exclusives.
- Parmi toutes les spécialisations vérifiant la contrainte sur la plausibilité, $m[A]$ est la moins engagée.

**Théorème 2.18.** [69] Soit une BBA $m^\Omega_1$ et $S(., A)$ l’ensemble des spécialisations $m^\Omega_2$ de $m^\Omega_1$ vérifiant $pl^\Omega_2(A) = 0$ pour un ensemble $A \subseteq \Omega$. Alors $m^\Omega_1[A]$ l’élément le moins engagé de $S(., A)$ vérifie :

$$m^\Omega_1[A](B) = \begin{cases} \sum_{C \subseteq \bar{A}} m^\Omega_1(B \cup C), & \forall B \subseteq A \\ 0 & \forall B \not\subseteq A \end{cases}$$ (2.19)
Cette spécialisation, appelée conditionnement de $m_1^\Omega$ sur $A$, a pour effet de transférer la masse de tous les éléments focaux $B$ vers $A \cap B$. Ce principe a donné naissance à l’appellation de modèle des croyances transférables (TBM), alternative aux interprétations des fonctions de croyance proposée par Dempster [11] ou Shafer [64].

Notons enfin que cette règle de conditionnement est justifiée par les principes fondateurs du TBM. D’autres interprétations de la théorie des fonctions de croyance existent et celles-ci peuvent impliquer d’autres règles de conditionnement. Ce point sera évoqué dans la section 2.7.

### 2.3.3 Combinaisons

L’opération de combinaison consiste à fusionner deux BBAs définies sur un même cadre de discernement en une seule structure de croyance.

Lors de l’introduction des fonctions de croyance, la règle de combinaison initialement proposée par Dempster était la suivante.

**Définition 2.19. Règle de Dempster [11]**

$$m_{1 \oplus 2}(A) = m_1 \oplus m_2(A) = \frac{1}{1 - K} \sum_{B \cap C = A \neq \emptyset} m_1(B)m_2(C) \quad \forall A \subseteq \Omega$$  

(2.20)

où $K = \sum_{B \cap C = \emptyset} m_1(B)m_2(C)$

Si nous adoptons l’interprétation des fonctions de croyance par des ensembles aléatoires, cette règle se justifie entièrement par la théorie probabiliste en s’appuyant sur le principe d’indépendance stochastique des sources [69]. Dans ce contexte, elle peut donc être considérée comme la seule valide. Cependant, d’autres interprétations du modèle ont depuis lors été proposées. Un grand nombre de variantes à la règle de Dempster ont donc été introduites [19] [49] [37] [44]. Parmi celles-ci, nous pouvons citer :

**Définition 2.20. Règle conjonctive TBM [70]**

$$m_{1 \cap^\bullet 2}(A) = m_1 \cap^\bullet m_2(A) = \sum_{B \cap C = A} m_1(B)m_2(C) \quad \forall A \subseteq \Omega$$  

(2.21)

**Définition 2.21. Règle disjonctive [66]**

$$m_{1 \cup^\bullet 2}(A) = m_1 \cup^\bullet m_2(A) = \sum_{B \cup C = A} m_1(B)m_2(C) \quad \forall A \subseteq \Omega$$  

(2.22)

**Définition 2.22. Règle de Yager [78]**

$$m_{12}(A) = \begin{cases} 
\sum_{B \cap C = A} m_1(B)m_2(C) & \forall A \not\subset \Omega, \quad A \neq \emptyset \\
\sum_{B \cap C = \emptyset} m_1(B)m_2(C) + m_1(\Omega)m_2(\Omega) & A = \Omega \\
0 & A = \emptyset 
\end{cases}$$  

(2.23)
Définition 2.23. Règle de Dubois & Prade [18]

\[
m_{12}(A) = \begin{cases} 
\frac{1}{\lambda} \sum_{B \cap C = A} m_1(B)m_2(C) + \frac{1}{\mu} \sum_{B \cap C = \emptyset, B \cup C = A} m_1(B)m_2(C) & \forall A \subseteq \Omega, \ A \neq \emptyset \\
0 & \text{si } A = \emptyset 
\end{cases}
\]  
\begin{equation}
(2.24)
\end{equation}

Parmi les règles présentées ci-dessus, la règle disjonctive fait systématiquement l’hypothèse qu’une des deux sources d’information n’est peut-être pas fiable. En ce sens, elle se démarque donc des autres règles. Les combinaisons restantes se distinguent par la manière dont elles redistribuent la masse provenant du conflit :

\[
\sum_{B \cap C = \emptyset} m_1(B)m_2(C)
\]  
\begin{equation}
(2.25)
\end{equation}

Cette masse est transférée sur l’ensemble vide par la règle conjonctive TBM, placée sur \(\Omega\) par la règle de Yager, répartie sur tous les éléments focaux par la règle de Dempster et transférée vers chaque union d’ensembles conflictuels par la règle de Dubois & Prade. Cette dernière est donc une règle hybride, se comportant de manière conjonctive lorsque les éléments focaux ont une intersection non vide et de manière disjonctive en cas de conflit.

Notons également qu’à l’instar de la règle de Dempster justifiée dans le contexte des ensembles aléatoires, la règle conjonctive TBM bénéficie aussi d’une justification axiomatique solide dans le cadre des croyances transférables [65] [69] :

**Theorème 2.24.** Soit trois BBAs \(m_i^\Omega\), \(m_j^\Omega\) et \(m_k^\Omega\). Si une règle de combinaison satisfait les propriétés suivantes :

- la combinaison \(m_{ij}^\Omega\) est une spécialisation de \(m_i^\Omega\) et de \(m_j^\Omega\) (\(m_{ij}^\Omega \subseteq m_i^\Omega\) et \(m_{ij}^\Omega \subseteq m_j^\Omega\)),
- commutativité : \(m_{ij}^\Omega = m_{ji}^\Omega\),
- associativité : \(m_{(ij)k}^\Omega = m_{ij(k)}^\Omega\),
- la combinaison avec une BBA catégorique \(m_A^\Omega\) est équivalente au conditionnement sur l’ensemble \(A \subseteq \Omega\) tel que défini dans le TBM : \(m_i^\Omega\) = \(m_i^{\Omega \setminus A}\)

alors la combinaison entre deux BBAs est définie par :

\[
m_{ij}^\Omega(A) = m_i^\Omega \otimes m_j^\Omega(A) = \sum_{A = B \cap C} m_i^\Omega(B)m_j^\Omega(C) \quad \forall A \subseteq \Omega
\]  
\begin{equation}
(2.26)
\end{equation}

Cette règle correspond à une version non normalisée de la combinaison de Dempster. De ce point de vue, toutes les règles de combinaison proposées peuvent alors être définies en justifiant l’abandon d’au moins un des quatre axiomes. Par exemple, la règle de Dempster ne respecte pas l’axiome de spécialisation. De même, ni la règle de Yager ni la règle de Dubois & Prade ne sont associatives.
2.3.4 Fiabilité des sources et affaiblissement

Le troisième type d’opération souvent appliqué aux fonctions de croyance est l’affaiblissement. Ce traitement intervient lorsque l’on manipule des BBAs provenant de sources considérées comme peu fiables. L’affaiblissement traduit alors la confiance accordée à la source K par un expert E. Si dans un cadre de discernement relatif à la fiabilité de la source $\Theta = \{\text{Fiable, Non Fiable}\}$, les masses valent $m^\Theta_E(\text{Fiable}) = \alpha$ et $m^\Theta_E(\text{Non Fiable}) = 1 - \alpha$ et si l’on suppose également que $m^\Omega_E[\text{Fiable}](A) = m^K_\Omega(A)$ et $m^\Omega_E[\text{Non Fiable}](\Omega) = 1$ alors les croyances $m^K_\Omega$ fournies par la source deviennent :

$$m^\Omega_E[m^K_E, m^K_\Omega](A) = \begin{cases} 
\alpha m^K_\Omega(A) & \forall A \subsetneq \Omega \\
\alpha m^K_\Omega(A) + 1 - \alpha & \text{si } A = \Omega
\end{cases}$$ (2.27)

Cette transformation est non seulement intuitive, mais elle est également justifiée par l’application du théorème de Bayes généralisé [69]. L’affaiblissement d’une fonction $m^K_\Omega$ par $\alpha$ s’écrit $m^{\Omega,\alpha}$ ou plus simplement $m^\alpha$.

Notons deux propriétés intéressantes de l’affaiblissement :

**Propriété 2.25.** L’affaiblissement d’une fonction de croyance vide est une fonction de croyance vide :

$$m^\Omega_{\Omega}^{\Omega,\alpha} = m^\Omega_{\Omega}$$ (2.28)

**Propriété 2.26.** Deux affaiblissements successifs $\alpha_1$ et $\alpha_2$ d’une fonction sont égaux à l’affaiblissement par le produit $\alpha_1 \alpha_2$ :

$$(m^{\alpha_1})^{\alpha_2} = m^{\alpha_1 \alpha_2}$$ (2.29)

2.3.5 Modifications du cadre de discernement

La dernière catégorie de manipulations de BBAs introduite dans ce chapitre bibliographique concerne les changements de cadre de discernement. Ce type de transformations, très fréquemment utilisé, intervient notamment lorsque l’on souhaite combiner des fonctions de croyance définies sur des cadres de discernement différents. Il est alors nécessaire de les ramener dans un référentiel commun avant toute opération de fusion. Ces transformations sont réalisées au moyen de raffinements et de grossissements dont le principe est illustré sur la figure 2.1.

![Fig. 2.1: Raffinement et grossissement](image)

De manière intuitive, un raffinement définit un cadre de discernement $\Omega$ plus fin que le cadre original $\Theta$. 
Définition 2.27. Raffinement

Si $\Omega$ et $\Theta$ sont deux cadres de discernement, un raffinement de $\Theta$ vers $\Omega$ est une fonction $\rho : 2^\Theta \to 2^\Omega$ telle que :
- l’ensemble $\{\rho(\theta), \theta \in \Theta\} \subseteq 2^\Omega$ est une partition de $\Omega$,
- $\rho(A) = \bigcup_{b \in A} \rho(\theta)$ $\forall A \subseteq \Theta$

Par extension, $\Omega$ est appelé un raffinement de $\Theta$ et la tranformation $\rho(A)$ d’un ensemble $A \subseteq \Theta$ peut s’écrire $A^\Omega$.

Le problème se pose alors d’exprimer une fonction de croyance $m^\Theta$ dans ce nouveau référentiel $\Omega$. La transformation, notée $m^\Theta \uparrow \Omega$, et justifiée par le principe de moindre engagement, respecte la règle suivante :

$$m^\Theta \uparrow \Omega(\rho(A)) = m^\Theta \uparrow \Omega(A^\Omega) = m^\Theta(A), \forall A \subseteq \Theta$$  \hspace{1cm} (2.30)

A l’inverse, $\Theta$ est appelé un grossissement de $\Omega$. Malheureusement, la fonction $\rho$ définissant le raffinement n’étant pas surjective, la fonction inverse $\rho^{-1}$ n’est pas définie pour tout élément de $\Omega$ :

$$\exists B \subseteq \Omega \text{ tq } \rho(A) \neq B, \forall A \subseteq \Theta$$  \hspace{1cm} (2.31)

Il faut donc compléter $\rho^{-1}$ sur les ensembles $B$ pour lesquels elle n’est pas définie. Plusieurs solutions sont mentionnées dans la littérature [15], mais nous retiendrons la moins engagée, à savoir la réduction extérieure définie par :

$$m^\Omega \downarrow \Theta(A) = \sum_{B \subseteq \Omega | B^\Theta = A} m^\Omega(B), \forall A \subseteq \Omega$$  \hspace{1cm} (2.32)

En raison de leur utilisation fréquente, deux cas particuliers de raffinement et de grossissement méritent d’être définis. Il s’agit de l’extension vide et de la marginalisation illustrés sur la figure 2.2.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig22.png}
\caption{Extension vide et marginalisation}
\end{figure}

Définition 2.28. Extension vide

Une extension vide est le cas particulier d’un raffinement de $\Theta$ vers le produit cartésien $\Theta \times \Omega$. La BBA résultante, notée $m^\Theta \times \Omega$, est définie pour tout $B \subseteq \Theta \times \Omega$ par

$$m^\Theta \times \Omega(B) = \begin{cases} m^\Theta(A) & \text{si } \exists A \subseteq \Theta \text{ tq } B = (A, \Omega) \\ 0 & \text{sinon} \end{cases}$$  \hspace{1cm} (2.33)
Définition 2.29. Marginalisation

De même, la marginalisation de $\Theta \times \Omega$ sur $\Theta$ est un cas particulier de grossissement. La BBA marginale $m_{\Theta \times \Omega \mid \Theta}$ vaut alors pour tout $A \subseteq \Theta$ :

$$m_{\Theta \times \Omega \mid \Theta}(A) = \sum_{\{B \in 2^{\Theta \times \Omega} | B^{\Theta} = A\}} m_{\Theta \times \Omega}(B)$$ (2.34)

où $B^{\Theta}$ est la projection de $B$ sur le cadre $\Theta$ définie par :

$$B^{\Theta} = \{\theta \in \Theta | \exists \omega \in \Omega, (\theta, \omega) \in B\}$$ (2.35)

2.4 Décomposition canonique

L’objectif poursuivi par la décomposition canonique est d’écrire les fonctions de croyance comme une combinaison de fonctions simples. Si cette décomposition existe, la structure initiale peut alors être interprétée comme la combinaison de croyances favorables à certaines hypothèses. Cette démarche, intuitivement proche de la factorisation polynomiale, a été introduite dès le livre fondateur de la théorie évidentielle [64] et a depuis été généralisée aux cas non séparables [67].

2.4.1 Décomposition canonique conjonctive

Les éléments pressentis pour la décomposition d’une BBA en une $\cap$ combinaison sont les BBAs simples. Celles-ci sont des fonctions définies par :

$$m_\Omega(A) = 1 - w$$
$$m_\Omega(\Omega) = w$$
$$m_\Omega(B) = 0 \ \forall B \in 2^\Omega \setminus \{A, \Omega\}$$ (2.36)

pour $A \subseteq \Omega$ et $w \in [0, 1]$. Par la suite, nous noterons $A^w$, une BBA simple allouant la masse $1 - w$ à l’ensemble $A \subseteq \Omega$.

La décomposition envisagée soulève alors trois questions.
- Existe-t-elle pour une BBA quelconque ? Et si ce n’est pas le cas, est-il possible de définir le sous-ensemble des BBAs pour lesquelles une telle factorisation est possible ?
- Est-elle unique ? Si cette propriété n’est pas vérifiée, la pertinence de la décomposition peut en effet être remise en question.
- Comment la calculer ? Cette tâche n’est de fait pas triviale.

En ce qui concerne la première question, il est clair que la décomposition n’est pas toujours possible. Il est par exemple aisé de montrer qu’une BBA bayésienne ne peut généralement pas être factorisée en une $\cap$ combinaison de fonctions simples. Ce constat conduit donc naturellement à la définition des BBAs séparables :

Définition 2.30. Séparabilité

Une BBA $m_\Omega$ est dite séparable si elle peut être décomposée en une $\cap$ combinaison de BBAs simples.
Pour répondre au deuxième point, une décomposition n’est, à strictement parler, jamais unique à cause de la propriété suivante :

**Propriété 2.31.** La combinaison conjonctive TBM de deux BBAs simples définies sur le même élément focal A vaut :

\[ A^{w_1} \cap A^{w_2} = A^{w_1 w_2} \]  

(2.37)

Une fonction simple \( A^w \) (\( w < 1 \)) peut donc toujours être décomposée en \( A^{w_1} \) et \( A^{w_2} \) tels que \( w = w_1 w_2 \).

Cependant, si on se limite aux seules décompositions n’admettant qu’une valeur sur chaque sous-ensemble \( A \) de \( \Omega \), autrement dit si l’on recherche une décomposition définie par une fonction \( 2^\Omega \rightarrow [0, 1] \), alors la propriété suivante est vérifiée.

**Propriété 2.32.** Si une BBA est séparable, il existe une fonction unique \( w : 2^\Omega \rightarrow [0, 1] \) telle que

\[ m^\Omega = \bigodot_{A \subseteq \Omega} A^{w(A)} \]  

(2.38)

Cette décomposition est appelée la décomposition canonique conjonctive de \( m^\Omega \) et les termes \( A^w \) sont les fonctions simples de support (Simple Support Function : SSF). Nous pouvons dès lors répondre à la deuxième question. Si une BBA est séparable, elle peut être décomposée en une \( \bigodot \) combinaison unique de SSFs.

Toutes les BBAs ne sont toutefois pas séparables, mais la décomposition canonique peut être extrapolée aux BBAs non séparables au moyen des fonctions simples généralisées.

**Définition 2.33. BBA simple généralisée**

Une BBA simple généralisée est une fonction \( \mu : 2^\Omega \rightarrow \mathbb{R} \) telle que

\[
\begin{align*}
\mu(A) &= 1 - w \\
\mu(\Omega) &= w \\
\mu(B) &= 0 \quad \forall B \in 2^\Omega \setminus \{A, \Omega\}
\end{align*}
\]

(2.39)

pour \( A \subsetneq \Omega \) et \( w \in [0, +\infty[ \).

Tout comme les BBAs simples, la fonction \( \mu \) vérifiant \( \mu(A) = 1 - w \) est noté \( A^w \). Toutefois, si \( w > 1 \) la fonction traduit une défiance vis-à-vis de l’hypothèse \( A \) et il faut qu’un avis favorable à \( A \) survienne pour retrouver l’état neutre :

\[ A^w \cap \bar{A}^{1/w} = A^1 \]  

(2.40)

dans ce cas, la fonction n’est plus une SSF mais une ISSF (Inverse Simple Support Function).

La généralisation de la décomposition aux BBAs non séparables est alors possible grâce au théorème suivant.

**Théorème 2.34.** Toute BBA non dogmatique peut être décomposée en une \( \bigodot \) combinaison unique de SSFs et ISSFs :

\[ m^\Omega = \bigodot_{A \subseteq \Omega} A^{w(A)} \]  

(2.41)

avec \( w(A) \in ]0, +\infty[ \forall A \subsetneq \Omega \).
Le modèle des croyances transférables (TBM)

Enfin, les \( w \) peuvent être calculés à partir des communalités \( q \) par la formule :

\[
\ln w(A) = - \sum_{B \supseteq A} (-1)^{|B|-|A|} \ln(q(B)) \quad \forall A \subset \Omega
\]

(2.42)

ce qui répond à la dernière question.

Notons encore que \( w \) étant unique et puisque \( m = \bigcap_{A \subseteq \Omega} A^{w(A)} \), il existe une bijection entre \( w \) et \( m \). Il s’agit donc d’une représentation équivalente de la structure de croyance au même titre que \( m \), \( pl \), \( bel \), \( b \) ou \( q \).

2.4.2 \( w \) ordonnancement

Maintenant que la règle conjonctive \( TBM \) et la décomposition canonique ont été introduites, la liste des ordonnancements présentée dans la section 2.3.1 peut être étendue en définissant deux nouvelles relations d’ordre

– le \( d \) ordonnancement :

\[
m_1 \sqsubseteq_d m_2 \text{ ssi } \exists m \text{ tq } m_1 = m \bigcap m_2
\]

(2.43)

– le \( w \) ordonnancement :

\[
m_1 \sqsubseteq_w m_2 \text{ ssi } w_1(A) \leq w_2(A) \quad \forall A \subseteq \Omega
\]

(2.44)

Ces nouvelles relations d’ordre sont liées aux autres par les implications suivantes :

\[
m_1 \sqsubseteq_w m_2 \Rightarrow m_1 \sqsubseteq_d m_2 \Rightarrow m_1 \sqsubseteq_s m_2 \Rightarrow \begin{cases} m_1 \sqsubseteq pl m_2 \\ m_1 \sqsubseteq q m_2 \end{cases}
\]

(2.45)

2.4.3 Règle conjonctive prudente

Grâce au \( w \) ordonnancement, on peut construire une nouvelle combinaison conjonctive ne s’appuyant que sur le principe de moindre engagement. En effet, si l’on nomme \( S_x(m) \) l’ensemble des fonctions de croyance plus engagées que \( m \) au sens de \( x \) (\( x \in \{ pl, q, s, d, w \} \) ), une manière élégante de définir la combinaison conjonctive entre \( m_1 \) et \( m_2 \) serait de choisir la fonction de \( S_x(m_1) \cap S_x(m_2) \) la moins engagée au sens de \( x \). Ce minimum n’existe malheureusement pas pour des \( BBAs \) quelconques et pour les \( pl \), \( q \), \( s \) ou \( d \) ordonnancements. En revanche, l’existence et l’unicité ont été démontrées pour des \( BBAs \) non dogmatiques et un \( w \) ordonnancement [13].

**Theorème 2.35.** Soit \( m_1 \) et \( m_2 \) des \( BBAs \) non dogmatiques. Alors, l’élément le moins engagé au sens des \( w \) de \( S_w(m_1) \cap S_w(m_2) \) existe et est unique. Il est défini par la fonction :

\[
w_1 \otimes w_2(A) = w_1(A) \wedge w_2(A) \quad \forall A \subset \Omega
\]

(2.46)

où \( \wedge \) est la \( T \)-norme minimum

L’unicité de la solution conduit donc naturellement à la définition d’une nouvelle règle de combinaison.
Définition 2.36. **Règle conjonctive prudente**

\[ m_1 \cap m_2 = \bigcap_{A \subseteq \Omega} A^{w_1(A) \land w_2(A)} \]

(2.47)

Cette règle est dite **prudente** car elle est idempotente. La combinaison de deux BBAs identiques donnera cette même BBA. De même, si \( m_1 \subseteq_w m_2 \) alors \( m_2 \not\subseteq_w m_1 \). Cette propriété peut s'interpréter comme suit. Si la BBA \( m_2 \) est moins engagée que \( m_1 \) au sens des \( w \), elle n'apporte aucune nouvelle information qui ne soit déjà contenue dans \( m_1 \). La combinaison **prudente** des deux sources est donc égale à la BBA la plus engagée, à savoir \( m_1 \). Ce mécanisme s'oppose au comportement de la combinaison conjonctive qui fait l'hypothèse que les deux sources d'informations sont indépendantes. Dans ce cas, même si les deux BBAs \( m_1 \) et \( m_2 \) sont égales, leur combinaison conjonctive sera plus engagée puisque leurs croyances se renforcent mutuellement.

### 2.4.4 Règle disjonctive hardie

Par un raisonnement similaire à celui tenu pour construire la règle conjonctive prudente, il est possible de définir une règle disjonctive hardie [14]. Une règle disjonctive repose sur l'hypothèse qu'une des deux sources peut être non fiable. Par conséquent, le résultat de la combinaison de deux BBAs doit être moins engagé que celles-ci. Formellement, si \( G_x(m) \) désigne l'ensemble des BBAs moins engagées que \( m \) au sens de \( \subseteq_x \) (avec \( x \in \{ w, d, s, pl, q \} \)), alors la combinaison doit vérifier \( m_{12} \in G_x(m_1) \cap G_x(m_2) \). Pour minimiser la perte d'information, la démarche hardie consiste alors à choisir la BBA la plus engagée de \( G_x(m_1) \cap G_x(m_2) \), à condition que celle-ci existe. Par exemple, en choisissant \( x = w \) cette approche conduit à la combinaison :

\[ w_{12} = w_1 \lor w_2 \]

(2.48)

où \( \lor \) est la T-conorme **maximum**

Malheureusement, cette définition n'a de sens que si \( m_1 \) et \( m_2 \) sont toutes les deux séparables. Cette restriction, très contraignante pour une combinaison disjonctive, a alors encouragé la construction d'une nouvelle décomposition.

**Définition 2.37. BBA généralisée négative**

Une BBA généralisée négative est une fonction \( \lambda : 2^\Omega \to \mathbb{R} \) telle que

\[
\begin{align*}
\lambda(A) & = 1 - v \\
\lambda(\emptyset) & = v \\
\lambda(B) & = 0 \quad \forall B \in 2^\Omega \setminus \{\emptyset, A\}
\end{align*}
\]

(2.49)

pour \( A \in 2^\Omega \setminus \{\emptyset\} \) et \( v \in [0, +\infty] \).

Une BBA généralisée négative \( \lambda \) est alors notée \( A_v \), et cette nouvelle forme simple permet une décomposition disjonctive des BBAs grâce au théorème suivant [14].
**Theoreme 2.38.** Toute BBA sous normale $m^\Omega$ peut être décomposée de manière unique en une $\bigcup$ combinaison de BBAs généralisées négatives $A_v(A)$ :

$$m^\Omega = \bigcup_{A \neq \emptyset} A_v(A)$$  \hspace{1cm} (2.50)

avec $v(A) \in ]0, +\infty[ \ \forall A \subset \Omega$

Cette décomposition peut être calculée à partir de l’implicabilité $b^\Omega$ à l’aide de la formule :

$$\ln v(A) = - \sum_{B \subseteq A} (-1)^{|A| - |B|} \ln b^\Omega(B)$$  \hspace{1cm} (2.51)

A nouveau, deux relations d’ordre partiel peuvent être définies :
- le $\text{dd}$ ordonnancement :

$$m_1 \sqsubseteq_{dd} m_2 \iff \exists \text{BBA } m \text{ tq } m_2 = m \bigcup m_1$$  \hspace{1cm} (2.52)

- le $\nu$ ordonnancement :

$$m_1 \sqsubseteq_{\nu} m_2 \iff v_1(A) \geq v_2(A), \ \forall A \neq \emptyset$$  \hspace{1cm} (2.53)

qui permettent de compléter les relations :

$$m_1 \sqsubseteq_{u} m_2 \Rightarrow m_1 \sqsubseteq_{d} m_2 \qquad m_1 \sqsubseteq_{v} m_2 \Rightarrow m_1 \sqsubseteq_{dd} m_2 \qquad m_1 \sqsubseteq_{pl} m_2 \Rightarrow m_1 \sqsubseteq_{q} m_2$$  \hspace{1cm} (2.54)

Enfin, et de manière analogue au développement dans le cas conjonctif, la règle disjonctive hardie peut être définie grâce au théorème suivant [14].

**Theoreme 2.39.** Soit $m_1$ et $m_2$ deux BBA sous normales, l’élément le plus engagé au sens de $v$ de $G_v(m_1) \cap G_v(m_2)$ existe et est unique. Il est défini par la fonction suivante :

$$v_{\text{\bigvee}2}(A) = v_1(A) \land v_2(A), \ \forall A \in 2^\Omega \setminus \{\emptyset\}$$  \hspace{1cm} (2.55)

qui permet de préciser la règle souhaitée pour un engagement au sens de $v$.

**Définition 2.40. Règle disjonctive hardie**

$$m_1 \bigvee m_2 = \bigvee_{A \subset \Omega} A_{v_1(A) \land v_2(A)}$$  \hspace{1cm} (2.56)

### 2.5 Prise de décision

Tout comme dans la théorie des probabilités ou des possibilités, les fonctions de croyance ont pour finalité la prise de décision. Généralement, cette **décision** doit choisir parmi les hypothèses celle qui semble la plus **vraisemblable**. De manière plus générale, la décision peut être le choix de l’action qui minimise une fonction de risque.
2.5.1 Transformation pignistique

Le modèle des croyances transférables (*TBM*) distingue clairement deux niveaux d’analyse. Il y a d’une part le niveau crędal auquel les différentes informations sont combinées et d’autre part le niveau décisionnel (cf. figure 2.3).

Contrairement au niveau crędal qui modélise la connaissance au moyen de structures de croyance, le niveau décisionnel utilise une représentation moins riche de la connaissance. Cette deuxième structure a les mêmes caractéristiques qu’une probabilité, à la différence près que ses valeurs sont directement déduites des fonctions de croyance et ne sont jamais combinées entre elles. La transformation *BetP* permettant de construire ces *probabilités pignistiques* est une fonction de la forme :

\[
BetP^\Omega : \Omega \rightarrow [0, 1] \\
\omega \rightarrow BetP(\omega)
\]

Cette transformation peut être définie à l’aide des axiomes suivants [68][70] :

**axiome 2.41.** \( \sum_{\omega \in \Omega} BetP^\Omega(\omega) = 1. \)

**axiome 2.42.** \( \forall \omega \in \Omega, \ BetP^\Omega(\omega) \) ne peut dépendre que des \( m^\Omega(A), \ \omega \in A \).

**axiome 2.43.** \( BetP^\Omega(\omega) \) est continu pour chaque \( m^\Omega(A), \ \omega \in A \).

**axiome 2.44.** \( BetP^\Omega(\omega) \) est invariant aux permutations de \( \Omega \).

**axiome 2.45.** Si \( X \subseteq \Omega \) n’est pas plausible, alors les \( BetP^\Omega(\omega) \) et les \( BetP^X(\omega) \) calculées après conditionnement sur \( X \) doivent respecter les égalités suivantes :

\[
BetP^\Omega(\omega) = BetP^X(\omega), \ \forall \omega \notin X \\
BetP^\Omega(\omega) = 0, \ \forall \omega \in X
\]

Il n’y a alors qu’une seule définition de \( BetP^\Omega \) vérifiant ces axiomes.
**Theorème 2.46.** Soit $m^\Omega$ une BBA sur le cadre $\Omega$, soit aussi $BetP^\Omega$ une fonction de la forme $\Omega \rightarrow [0,1]$, alors, si $BetP^\Omega$ satisfait les axiomes 2.41-2.45, elle est nécessairement définie par :

$$BetP^\Omega(\omega) = \frac{1}{1 - m^\Omega(\emptyset)} \sum_{A \ni \omega} \frac{m^\Omega(A)}{|A|}, \quad \forall \omega \in \Omega$$  \hspace{1cm} (2.57)

Tout comme la fonction de masse $m^\Omega$ peut être notée $m$ lorsqu’il n’y a pas d’ambiguïté, $BetP^\Omega$ peut également s’écrire $BetP$.

Enfin, cette définition peut être généralisée à tout ensemble $B \subseteq \Omega$ :

$$BetP^\Omega(B) = \sum_{A \subseteq \Omega} \frac{|A \cap B|}{|A|} \frac{m^\Omega(A)}{1 - m^\Omega(\emptyset)}$$  \hspace{1cm} (2.58)

### 2.5.2 Minimisation du risque

De manière générale, la prise de décision sur un cadre $\Omega$ consiste à choisir l’action $\alpha \in A$ qui minimise un risque $R_P(\alpha)$ calculé à partir d’une mesure de probabilité $P(\omega) : \Omega \rightarrow [0,1]$ tq $\sum_{\omega \in \Omega} P(\omega) = 1$ et d’une fonction de coût $C(\alpha, \omega) : A \times \Omega \rightarrow \mathbb{R}$ quantifiant les répercussions des actions $\alpha$ pour chaque événement $\omega$ [51][52].

La fonction de risque $R_P(\alpha)$ est alors définie par

$$R_P(\alpha) = \sum_{\omega \in \Omega} P(\omega)C(\alpha, \omega)$$  \hspace{1cm} (2.59)

Pour $P = BetP$, la solution $\hat{\alpha}$ devient :

$$\hat{\alpha} = \arg\min_{\alpha \in A} \sum_{\omega \in \Omega} BetP(\omega)C(\alpha, \omega)$$  \hspace{1cm} (2.60)

Enfin, si la fonction de coût vaut $C(\omega_i, \omega_j) = 1 - \delta_{ij}$, la solution $\hat{\alpha} = \hat{\omega}$ vaut :

$$\hat{\omega} = \arg\min_{\omega \in \Omega} [1 - BetP(\omega)] = \arg\max_{\omega \in \Omega} [BetP(\omega)]$$  \hspace{1cm} (2.61)

Ce cas particulier, très courant en pratique, revient à considérer que l’ensemble des actions possibles $A$ est constitué des hypothèses $\omega_i \in \Omega$, que les coûts consécutifs à une mauvaise décision ($\omega_i \neq \omega_j$) sont tous équivalents, et que le coût d’une bonne décision est nul. Dans ce cas, la décision est naturellement prise selon un critère de maximisation de $BetP$.

### 2.5.3 Décisions alternatives

Le mécanisme de prise de décision proposé par le TBM repose sur le principe de maximisation de l’utilité espérée. Bien que cette approche soit très largement utilisée, il existe des alternatives [51]. Il est par exemple possible d’adopter une attitude plus optimiste face aux données. Dans ce cas, nous pouvons minimiser la fonction de risque $R_P$ définie par :

$$R_P = \sum_{\omega \in \Omega} P(\omega)C(\alpha, \omega)$$
\[ R_{P-}(\alpha) = \min_{\beta \leq P \leq \beta \leq P \leq \beta} R_P(\alpha) \] (2.63)

Si la fonction de coût vaut \( C(\omega_i, \omega_j) = 1 - \delta_{ij} \), la solution \( \hat{\alpha} = \hat{\omega} \) devient :

\[ \hat{\omega} = \argmin_{\omega \in \Omega} [1 - pl(\omega)] = \argmax_{\omega \in \Omega} [pl(\omega)] \] (2.64)

De même, une démarche pessimiste peut conduire à minimiser la fonction de risque \( R_{P+} \) définie par :

\[ R_{P+}(\alpha) = \max_{\beta \leq P \leq \beta \leq P \leq \beta} R_P(\alpha) \] (2.65)

qui correspond au risque maximum consécutif à l’action \( \alpha \) pour les mesures de probabilités \( P \) telles que \( bel(\omega) \leq P(\omega) \leq pl(\omega) \) \( \forall \omega \in \Omega \) :

\[ \hat{\omega} = \argmin_{\omega \in \Omega} [1 - bel(\omega)] = \argmax_{\omega \in \Omega} [bel(\omega)] \] (2.67)

### 2.6 Comparaison de BBAs

Deux éléments de comparaison entre BBAs sont parfois avancés dans la théorie des fonctions de croyance. Il s’agit de la similarité et du conflit. Ces deux mesures peuvent être exploitées indépendamment ou être couplées pour former une mesure hybride ou bidimensionnelle.

#### 2.6.1 Similarité

Un grand nombre de mesures permettant de quantifier la similarité (ou de manière analogue la dissimilarité) entre deux fonctions de croyance ont été proposées dans la littérature. Ces mesures sont notamment utilisées pour mesurer la vitesse de convergence d’algorithmes ou pour évaluer la perte d’information suite à des simplifications réalisées sur des BBAs [41]. Elles peuvent également intervenir dans l’optimisation de certains paramètres ou la caractérisation des sources [27].
Parmi ces mesures, on trouve très naturellement la généralisation de certaines mesures probabilistes telles que la divergence de Kullback-Liebler \([59]\) :

\[
d_{KL}(m_1, m_2) \triangleq \sum_{A \subseteq \Omega} m_1(A)m_2(A) (1 - (m_1 \oplus m_2)(A))
\]

ou la distance de Bhattacharyya \([62]\) \([26]\) :

\[
d_B(m_1, m_2) \triangleq \left[ 1 - \sum_{A \subseteq \Omega} \sqrt{m_1(A)m_2(A)} \right]^p \quad p \in \mathbb{R}_0^+
\]

Cependant, la plupart des mesures de similarités s'inspire davantage d'une interprétation géométrique des fonctions de croyance. Aussi, des distances de type \(L_1\), \(L_2\) ou \(L_\infty\) ont été suggérées. Il y a par exemple :

- la distance \(L_1\) sur les credibilités \([35]\) :

\[
d_H(m_1, m_2) \triangleq \sum_{A \subseteq \Omega} |\text{Bel}_1(A) - \text{Bel}_2(A)|,
\]

- \(L_\infty\) sur les probabilités pignistiques :

\[
d_T(m_1, m_2) = \max_{A \subseteq \Omega} |\text{BetP}_1(A) - \text{BetP}_2(A)|
\]

connue sous le nom de distance de Tessem \([73]\),

- la distance euclidienne \(L_2\) entre les masses :

\[
d_E(m_1, m_2) \triangleq \sqrt{\frac{1}{2} \left( \langle m_1, m_1 \rangle_E + \langle m_2, m_2 \rangle_E - 2 \langle m_1, m_2 \rangle_E \right)}
\]

ou

\[
\langle m_1, m_2 \rangle_E = \sum_{A \subseteq \Omega} m_1(A)m_2(A)
\]

Cette dernière n’est généralement pas directement utilisée. On lui préfère souvent une des variantes redéfinissant le produit scalaire \(\langle m_1, m_2 \rangle\). Parmi celles-ci, il y a la pseudo distance proposée par Fixsen & Mahler \([25]\), pour laquelle

\[
\langle m_1, m_2 \rangle_{FM} = \sum_{A \subseteq \Omega} \sum_{B \subseteq \Omega} m_1(A)m_2(B) \frac{|A \cap B|}{|A||B|}
\]

(si les hypothèses sont \textit{a priori} équiprobables) ou encore la distance de Jousselme \([41]\) :

\[
\langle m_1, m_2 \rangle_J = \sum_{A \subseteq \Omega} \sum_{B \subseteq \Omega} m_1(A)m_2(B) \frac{|A \cap B|}{|A \cup B|}
\]

Nous n’avons ici donné qu’un aperçu de la très grande variété des mesures existantes. Pour un inventaire plus complet ou une étude comparative nous invitons le lecteur à consulter les articles de Jousselme \([40]\) ou \([17]\).
2.6.2 Conflit

Le conflit constitue un deuxième élément de comparaison des fonctions de croyance. Alors que les mesures de similarité tentent de quantifier la ressemblance entre des BBAs, le conflit évalue le degré de contradiction. Par exemple, une BBA bayésienne n’est pas similaire à une BBA vide, mais elles ne sont pas conflictuelles.

Dempster définit [11] le facteur de conflit comme la masse de l’ensemble vide obtenue par combinaison conjonctive des deux structures de croyance avant normalisation. Ce conflit peut être défini de manière équivalente à l’aide de la combinaison TBM par :

\[
m_{\text{TBM}}(\emptyset) = \sum_{A \cap B = \emptyset} m_1(A)m_2(B) \tag{2.76}\]

Cette valeur est d’autant plus grande que les deux BBAs sont engagées sur des hypothèses d’intersection vide.

Plusieurs travaux ont étudié cette expression du conflit. Suivant cette tendance, Shubert [63] a par exemple proposé de fractionner le conflit à l’aide d’une décomposition canonique des deux BBAs.

Remarquant que si une BBA n’est pas consonante, alors \( m_{\text{TBM}}(\emptyset) \neq 0 \), Martin et al. [50] ont introduit la notion d’auto-conflit reflétant la part de contradiction interne à une BBA. Suivant cette logique, certains auteurs soutiennent alors l’idée qu’il est préférable d’utiliser plusieurs mesures pour séparer le conflit interne du conflit externe [16]. L’exploitation conjointe de plusieurs indicateurs a également été exploitée pour qualifier de manière plus fine la relation entre les deux BBAs [46]. Dans ce deuxième cas de figure, la frontière entre la notion de conflit et de similarité devient malheureusement moins claire.

2.7 Interprétations de la théorie des croyances

Bien que la théorie des fonctions de croyance constitue un cadre mathématique formel très intéressant, elle a dès son apparition fait l’objet d’un certain nombre de critiques. Le point central de ces discussions concerne la question de l’interprétation du modèle.

Dans son article original, Dempster [11] interprète les fonctions \( \text{bel} \) et \( \text{pl} \) comme des mesures \( P_* \) et \( P^* \) encadrant la distribution de probabilités \( P \) inconnue régissant le processus étudié. Il parle alors de probabilités généralisées. Par la suite, lorsque Shafer reprend la même représentation [64], il présente le modèle comme une théorie des preuves (evidence theory). Plus il y a d’éléments cautionnant une hypothèse, plus grande devient la croyance en celle-ci.

Plusieurs études théoriques [69][58], ainsi que l’analyse d’exemples conduisant à des résultats contre intuitifs ou faux, ont depuis lors montré les limites de chacun de ces modèles. De nouvelles interprétations ont donc été proposées mais d’une manière générale les points de vue probabiliste et crédibiliste coexistent encore.

Un des premiers reproches à l’encontre du modèle a été formulé par Zadeh [79] et concerne la normalisation de la combinaison conjonctive. En particulier, et
sous réserve d’adopter l’interprétation évidentielle, la combinaison de BBAs fortement conflictuelles conduit à un résultat non intuitif. En effet, si par exemple deux médecins auscultant un patient évoquent tous deux la possibilité d’une tumeur avec une croyance de 0.01 contre deux diagnostics différents avec une croyance de 0.99, la règle de Dempster conduit à accorder une crédibilité de 1 à la tumeur alors que les deux experts sont quasiment convaincus qu’il ne s’agit pas d’une tumeur.

Cet exemple a également été repris par Smets pour justifier la fusion conjonctive TBM [69]. Il note au passage que si le conflit permet d’envisager un diagnostic en dehors du cadre de discernement (monde ouvert), l’exemple de Zadeh ne discrédite pas pour autant la combinaison de Dempster par rapport aux probabilités. En effet, les fonctions combinées sont des BBAs bayésiennes, par conséquent une résolution probabiliste aurait donné le même résultat que la combinaison de Dempster.

Pearl [58] a quant à lui étudié au moyen d’exemples les limites de l’interprétation probabiliste. Il a ainsi montré que si une fonction de croyance représente une famille de distributions de probabilités, l’inverse n’est pas vrai : toutes les familles de distribution ne peuvent pas être représentées par une BBA. Par exemple, la famille des probabilités comprises entre 0 et 1/2 et définie sur trois hypothèses mutuellement exclusives et exhaustives ne peut pas être représentée par une fonction de croyance. Cette famille impose en effet

\[ bel(\omega_i) = 0 \quad \text{et} \quad pl(\omega_i) = 0.5 \quad \forall i \in \{1, 2, 3\} \quad (2.77) \]

Or, si l’on ajoute la condition imposant que la somme des masses doit valoir 1, aucune fonction de croyance ne permet de satisfaire toutes les contraintes.

Notons toutefois que l’exemple est valable que sous hypothèse de monde fermé. En effet, si la masse \( m(\emptyset) \) peut être non nulle, la fonction \( m(\Omega) = m(\emptyset) = 0.5 \) satisfait bien toutes les conditions.

Pearl [58] a également étudié le conditionnement d’une croyance \( bel \) sur un sous-ensemble \( B \subseteq \Omega \). Si l’on adopte la règle de conditionnement de Dempster (directement déduite de sa combinaison conjonctive) :

\[
bel_D(A | B) = \frac{bel(A \cup B) - bel(B)}{1 - bel(B)} \quad (2.78)
\]
\[
pl_D(A | B) = \frac{pl(A \cap B)}{pl(B)} \quad (2.79)
\]
on obtient alors l’inégalité suivante :

\[
\min_{P \in \mathcal{P}} (P(A | B)) \leq bel_D(A | B) \leq pl_D(A | B) \leq \max_{P \in \mathcal{P}} (P(A | B)) \quad (2.80)
\]
ou \( \mathcal{P} \) est la famille des probabilités. Autrement dit, et comme Dempster l’avait déjà noté, l’intervalle \([bel, pl]\) obtenu lors d’un conditionnement est inclus dans l’intervalle formé par les bornes min et max du conditionnement appliqué aux distributions de la famille \( \mathcal{P} \).

Afin que le conditionnement reste cohérent avec l’interprétation de probabilité généralisée, Halpern et Fagin [34] ont proposé la règle :

\[
bel_{HF}(A | B) = \frac{bel(A \cap B)}{bel(A \cap B) + pl(A \cap B)} \quad (2.81)
\]
\[ p_{HF}(A|B) = \frac{pl(A \cap B)}{bel(A \cap B) + pl(\bar{A} \cap B)} \]  

Cette règle respecte effectivement l’inégalité

\[ \min_{P \in P} (P(A|B)) = bel_{HF}(A|B) \leq pl_{HF}(A|B) = \max_{P \in P} (P(A|B)) \]  

\[ (2.82) \]

Cette règle respecte effectivement l’inégalité

\[ \min_{P \in P} (P(A|B)) = bel_{HF}(A|B) \leq pl_{HF}(A|B) = \max_{P \in P} (P(A|B)) \]  

\[ (2.83) \]

Malheureusement, ce deuxième conditionnement a d’autres inconvénients. En particulier, il n’est pas commutatif. Le conditionnement sur deux sous-ensembles \(B_1\) et \(B_2\) ne peut donc pas être calculé de manière séquentielle, en conditionnant d’abord sur \(B_1\) puis sur \(B_2\). Il faut nécessairement calculer l’intersection \(B_1 \cap B_2\) avant de procéder au conditionnement.

Ces difficultés rencontrées également lors de la combinaison de BBAs [34] ont contribué au succès du TBM. En s’affranchissant des contraintes probabilistes et en acceptant une masse non nulle sur l’ensemble vide, beaucoup d’écueils peuvent en effet être évités. Il ne faut cependant pas perdre de vue que ce modèle étant de nature subjective, la signification des valeurs numériques manipulées peut être sujette à caution.
Chapitre 3

Cadre crédibiliste pour le pistage

Sommaire

3.1 Définition du problème ........................................ 44
3.2 Cadres de discernement ........................................ 45
   3.2.1 Modélisation de la localisation des pistes .......... 46
   3.2.2 Spécificités de l’application ..................... 46
   3.2.3 Arguments en faveur du TBM .................... 47
3.3 Construction des BBAs .................................... 48
   3.3.1 Représentation des détections .................. 48
   3.3.2 Informations topographiques .................. 51
   3.3.3 Conditionnement ................................ 52
   3.3.4 Changement d’échelle ............................ 53
3.4 Fusion des pistes et des détections ...................... 54
   3.4.1 Règle de combinaison ............................ 54
   3.4.2 Prédiction ........................................ 56
   3.4.3 Rétroprojection des pistes ....................... 57
   3.4.4 Mise à jour des paramètres .................... 58
3.5 Analyse de sensibilité .................................... 59
   3.5.1 Localisation mono capteur ......................... 60
   3.5.2 Localisation multi capteurs ..................... 63
   3.5.3 Synthèse de l’analyse ............................ 65
3.6 Conclusions .................................................. 65
3.1 Définition du problème

Les systèmes de surveillance multi capteurs présentés dans le chapitre introductif peuvent maintenant être reconsidérés dans le contexte des fonctions de croyance.

En adoptant une représentation des pistes conforme au formalisme TBM, chacun des blocs fonctionnels de l'architecture générique peut alors être réinterprété en terme d’opérateurs et de manipulations de BBAs.

Modélisation des états de pistes

L’état d’une piste regroupe les informations relatives à la localisation et à l’identification d’une cible à un moment donné. Ces deux aspects indépendants vont être traités de manière distincte et deux cadres de discernement vont être définis : l’un pour la localisation et l’autre pour l’identification. La définition de ces cadres de discernement et des BBAs correspondantes constitue une des originalités de l’approche défendue dans ce travail et sera détaillée dans la section 3.2. Précisons encore que si ces deux points sont abordés, c’est principalement le problème de la localisation qui sera étudié.

Prétraitement et alignement des détections

Dans le cadre du TBM, le prétraitement consiste à construire des BBAs modélisant l’incertitude des détections en fonction de la nature des détecteurs. Ces BBAs doivent également être alignées afin de pouvoir comparer et fusionner les détections provenant de capteurs différents. Ces traitements seront détaillés dans la section 3.3.

Prédiction et mise à jour

Le filtrage récursif permettant d’actualiser l’état des pistes est ici reconsidéré dans le cadre des fonctions de croyance.

Tout comme dans l’architecture générique, ce traitement comporte une phase de prédiction suivie d’une étape de mise à jour. Ensemble, ces deux opérations accomplissent la fusion des pistes et des détections qui sera développée dans la section 3.4.

Association

Afin de ne pas surcharger la description du modèle proposé, nous considérerons dans un premier temps le cas simple de la mise à jour d’une seule piste. L’étape d’association, nécessaire lors du suivi simultané de plusieurs pistes, sera introduite dans le chapitre 4. A cette occasion, nous verrons que le formalisme des BBAs peut directement être exploité pour évaluer la vraisemblance qu’une détection et une piste se rapportent à la même cible.
Gestion des pistes

Contrairement aux autres blocs fonctionnels, le module de gestion des pistes dépend davantage de l’application que du formalisme adopté. Par conséquent, ce point ne sera pas développé avant la partie expérimentale du travail (chapitre 7).

3.2 Cadres de discernement

La littérature mentionne plusieurs travaux consacrés à l’utilisation des fonctions de croyances dans le contexte de la fusion multi capteurs.

Parmi ceux-ci, Ayoun [3] propose de placer les cibles sur un pavage discret de la scène. Cette première étude fait l’hypothèse que les détections sont localisées avec une incertitude inférieure à la taille d’un pavé et que chaque capteur est capable de détecter au plus une cible. Le travail se focalise alors sur le problème du dénombrement des cibles. Celui-ci est effectué en recherchant le nombre minimum de cibles permettant d’expliquer toutes les détections et les aspects liés à la localisation ne sont pas abordés.

Bien que la discrétisation de la scène semble une piste intéressante, la mise en œuvre de cette méthode doit toutefois faire face à plusieurs difficultés. Tout d’abord, les capteurs sont supposés localiser les détections avec des précisions comparables. Or, beaucoup de systèmes de surveillance intègrent des détecteurs caractérisés par des précisions très différentes. Il paraît alors judicieux de tenir compte de ces spécificités dès la modélisation du problème.

De plus, la nécessité de placer les détections sur un seul pavé introduit une deuxième source de perturbation par effet de bord. En effet, si une cible est localisée à la frontière de deux pavés, les capteurs risquent de placer leurs détections respectives sur les deux pavés voisins. Par conséquent, en l’absence de prise en compte du voisinage, la cible sera comptée deux fois.

L’approche présentée par Nassreddine et al. [56] se concentre davantage sur la question de la localisation. Ce travail, directement inspiré de l’étude de Gning et al. [29], propose de généraliser l’analyse par intervalles en remplaçant les boîtes englobantes par des éléments focaux consonnants. En adaptant la propagation des contraintes, ce modèle permet de combiner la richesse de représentation des BBAs avec la garantie de localisation de l’analyse par intervalles. La qualité de cette représentation se paie toutefois par une charge plus importante de calculs. Il faut également noter qu’il est impératif de procéder régulièrement à des approximations afin de maintenir un nombre acceptable d’éléments focaux.

Le cadre de discernement dans lequel l’état de la piste est représenté est le résultat d’un produit de cadres continus monodimensionnels. Il n’y a donc plus de problèmes liés à discrétisation de l’espace. Cependant, avec ce modèle, tous les éléments focaux ont la forme d’une boîte. Cette simplification est donc très pratique pour la propagation des contraintes et pour une quantification continue de l’incertitude, mais elle rend difficile la modélisation de la géométrie des incertitudes.
3.2.1 Modélisation de la localisation des pistes

Le cadre de discernement proposé pour la localisation repose, comme dans l’approche d’Ayoun [3], sur un découpage discret et bidimensionnel de la scène. Sur ce pavage, chaque cellule, hypothèse singleton du cadre de discernement, représente la localisation possible d’une cible à un moment donné. Cependant, plutôt que considérer le cas de BBAs n’occupant qu’une seule cellule, les BBAs modéliseront l’imprécision de localisation relative aux détections et aux pistes. Chaque élément focal peut alors être vu comme une croyance accordée indistinctement à plusieurs cellules de la grille spatiale et il n’y a en théorie aucune contrainte sur leur géométrie (figure 3.1). Les éléments focaux ne doivent ni avoir des formes particulières telles que des rectangles ou des ellipses ni même correspondre à des hypothèses spatialement connexes. En revanche, la précision de la localisation sera inéluctablement limitée par la discrétisation du cadre de discernement.

![Fig. 3.1: BBA de localisation d’une cible](image)

3.2.2 Spécificités de l’application

Certaines caractéristiques de l’application considérée dans ce travail doivent être prises en compte dans la représentation crédibiliste. Ces spécificités sont la taille du cadre de discernement et la décentralisation de l’architecture.

**Taille du cadre de discernement**

Le cadre de discernement résultant de la discrétisation de la scène est beaucoup plus grand que ceux usuellement manipulés dans la théorie des fonctions de croyance. Rappelons qu’un cadre de discernement $\Omega$ admet potentiellement $2^{|\Omega|}$ éléments focaux différents. Les zones surveillées par le système étudié ayant typiquement des surfaces de plus d’un hectare, la localisation métrique des pistes sur ceux-ci implique donc que les BBAs peuvent contenir plus de $2^{10000}$ éléments focaux. Non seulement les calculs sur des structures de cette taille sont exclus, mais même la représentation d’un élément focal parmi les $2^{10000}$ possibilités devient critique. Ce constat a au moins deux conséquences sur le traitement des données. D’une part la représentation
des BBAs doit être compatible avec la taille de la scène et l’hétérogénéité des capteurs. D’autre part, tout comme pour la généralisation de l’analyse par intervalles [56], il est nécessaire de limiter le nombre d’éléments focaux.

**Architecture décentralisée**

L’intégration de la fusion des pistes au sein d’une architecture décentralisée induit également des contraintes sur la représentation des BBAs. En effet, la décentralisation est notamment caractérisée par la scalabilité et la modularité du système. Or, ces caractéristiques risquent d’être compromises par le choix d’un cadre de discernement englobant toute la scène. Pour que ces propriétés soient respectées, il faut que la taille de $\Omega$ ne soit pas bornée et qu’elle puisse évoluer en cours de fonctionnement.

3.2.3 Arguments en faveur du TBM

Soulignons d’abord que nous ne prétendons pas à l’universalité du modèle TBM. En fonction de la qualité des données, des objectifs, des contraintes ou encore des ressources de calcul disponibles, chaque solution peut bénéficier d’un avantage sur les approches concurrentes.

Cependant, outre son originalité, la représentation de la localisation par des BBAs adoptée dans ce travail présente un certain nombre de caractéristiques qui peuvent s’avérer intéressantes.

**Absence de contraintes géométriques**

Comme cela a déjà été mentionné, le modèle n’impose pas de formes prédéfinies aux éléments focaux (à la résolution spatiale des pavés près).

Cette spécificité est intéressante lorsque les détections sont entachées d’incertitudes de géométrie variable. C’est notamment le cas si le système est constitué de capteurs hétérogènes. Par exemple, les détections d’un capteur passif infrarouge (PIR) peuvent être selon le cas, de formes circulaires, linéaires ou angulaires. Pour peu que le module de fusion d’un nœud de détection connaisse la nature des capteurs locaux, il peut alors construire les BBAs de formes appropriées et les structures résultantes peuvent être manipulées de manière homogène par tous les nœuds du système.

Les BBAs construites directement à partir des informations provenant des capteurs peuvent également être ajustées en intégrant des informations provenant de données topographiques. Bien entendu, le TBM n’a pas l’exclusivité de cette caractéristique, mais la souplesse de représentation facilite grandement la tâche.

Notons toutefois que ces deux spécificités se retrouvent également dans les approches par grilles probabilistes [24].

**Préservation du conflit**

Chaque élément focal d’une structure de croyance représente un morceau d’information favorable à l’idée que la cible se trouve quelque part à l’intérieur de la
zone considérée. Deux éléments focaux disjoints reflètent donc deux avis divergents présents dans une seule BBA. Tant que les nouvelles détections ne contredisent aucune de ces hypothèses de localisation, chacune d’elles sera alors affinée par combinaison conjonctive sans que leurs plausibilités relatives ne soient altérées. L’analyse du conflit et l’identification des sources en contradiction systématique avec leurs voisins permettent ainsi de mettre en évidence certains dysfonctionnements de capteurs.

**Indépendance entre la taille et la masse des éléments focaux**

La taille d’un élément focal est indépendante de sa masse. L’idée soutenue par cette propriété est que ce n’est pas parce qu’une détection est imprécise qu’elle est peu fiable. En comparaison, un modèle probabiliste estimerait la probabilité de la position de la piste. Plus l’incertitude de localisation est grande plus la probabilité de se trouver à un endroit est faible. Bien qu’il soit délicat de trancher pour l’une ou l’autre de ces deux approches, cette distinction illustre bien la différence entre les deux points de vue.

### 3.3 Construction des BBAs

Lorsqu’une détection est transmise à la fusion par un des capteurs du système, la première opération consiste à projeter l’information dans l’espace de représentation et de décision de la fusion de données. Cette tâche, qui correspond aux étapes de prétraitement et d’alignement des données se traduit par la construction d’une BBA à partir de la détection, des caractéristiques du capteur ainsi que d’éventuelles informations contextuelles.

Les différents points relatifs à la construction des BBAs à partir des détections et des informations disponibles seront détaillés dans la suite de cette section.

#### 3.3.1 Représentation des détections

La forme et la taille des BBAs de localisation peuvent varier selon le capteur. Un modèle ad hoc de construction des BBAs a été défini afin de tenir compte de leurs spécificités. Ce modèle est ici motivé par son sens pratique.

Tout d’abord, nous modéliserons toutes les détections par des BBAs consonantes. Ce choix repose sur l’analyse suivante. Comme nous le verrons plus loin, le conflit apparaissant lors de la combinaison conjonctive TBM de deux BBAs peut être interprété comme la crédibilité que ces BBAs représentent les localisations de deux pistes différentes. Or, si une BBA comprend des éléments focaux disjoints, sa combinaison conjonctive avec elle-même induira nécessairement une masse non nulle sur l’ensemble vide. Cet auto-conflit reflète donc la crédibilité que la BBA représente les localisations de deux pistes différentes. Ce cas de figure a du sens si la BBA est formée à partir de la combinaison de sources élémentaires (clustering de source), mais s’il s’agit d’un seul capteur physique, son interprétation est plus difficile. Ce raisonnement justifie le choix de BBAs sans auto-conflit, mais pas de se limiter aux
seuls cas consonants. Toutefois, ce type de \textit{BBAs} étant suffisant pour modéliser les incertitudes des capteurs, nous l’adoptons par souci de simplicité. Insistons enfin sur le fait que cette restriction ne concerne que les \textit{BBAs} provenant des capteurs. Lors du suivi d’une piste, la \textit{BBA} résultante ne sera, après plusieurs combinaisons, plus consonante et un conflit interne peut apparaître.

Comme l’illustre la figure 3.2, les \textit{BBAs} de détections sans \textit{a priori} construites lors des expériences, sont constituées d’au plus trois éléments focaux. Ce nombre provient de la méthode de construction des \textit{BBAs} appliquée aux détections. L’élément le plus engagé correspond à la localisation transmise par le capteur, le second tient compte de l’imprecision de détection et enfin le troisième traduit l’incertitude de position et d’orientation du capteur. La plupart des capteurs fournissent des positions ou des orientations de détections dans un référenciel relatif. Dans les applications multi capteurs il est donc nécessaire de connaître la position et l’orientation de tous les capteurs afin de procéder à l’alignement des données. En effet, pour être comparées, et si besoin fusionnées, les \textit{BBAs} doivent être représentées dans le même cadre de discernement. C’est cet alignement, source d’une incertitude supplémentaire, qui justifie le troisième élément focal de notre modèle.

\textbf{Fig. 3.2: BBA d’une détection}

**Application à différents types de capteurs**

Nous pouvons appliquer ce principe général pour définir la géométrie des éléments focaux pour les différents types de capteurs rencontrés au cours de ce travail.

**Capteur isotrope** Certains capteurs, comme la plupart des capteurs acoustiques, n’estiment ni la distance ni la direction des événements qu’ils détectent. Dans ce cas, les \textit{BBAs} de détections seront constituées de deux disques consonants centrés sur le capteur (à la discrétisation du pavage près), comme illustré sur la figure 3.3. Le diamètre du plus petit élément focal est alors défini par la portée du capteur et le deuxième élément focal, de plus grand cardinal, tient compte de l’incertitude sur la position du capteur.

**Capteur directionnel** Les capteurs passifs infrarouges (\textit{PIR}) peuvent parfois estimer la direction relative des détections. Dans ce cas, les \textit{BBAs} sont constituées

![Diagramme de BBA d’une détection](image_url)
Capteur estimant la distance Lorsque le capteur est capable d’estimer la distance de détection, comme en sont capables certains traitements d’images de caméras, l’élément focal le plus engagé est le pavé sur lequel a eu lieu la détection. Le deuxième élément focal est un secteur d’anneau centré sur la détection, dont l’ouverture angulaire et les rayons minimum et maximum sont définis par l’incertitude angulaire et de distance du capteur. Le troisième élément focal est un secteur de disque dont le rayon est fixé par la portée du capteur et dont l’ouverture angulaire dépend des incertitudes de position et d’orientation du capteur.

Répartition de la masse

Jusqu’à présent, nous avons détaillé la construction géométrique des éléments focaux, mais nous n’avons pas abordé la question de la répartition de la masse sur ceux-ci. Étant donné que les BBAs de détection sont consonantes, l’ordonnancement des hypothèses selon un critère de plausibilité ou de probabilité pignistique
ne dépend pas de cette répartition. Cependant, lorsque ces BBAs sont combinées pour former une piste, un conflit partiel peut apparaître et l’attribution initiale de la masse peut modifier la décision de localisation. L’étude de son incidence est toutefois relativement complexe et liée aux règles de combinaison ainsi qu’au critère de décision. En pratique, et à défaut d’une connaissance suffisante des incertitudes, nous répartirons la masse de manière égale sur tous les éléments focaux.

3.3.2 Informations topographiques

Le modèle par pavage discret de la scène facilite la prise en compte d’a priori topographiques. Si par exemple la fusion a connaissance d’un obstacle masquant partiellement la vue d’un capteur, les BBAs formées à partir des détections de ce capteur peuvent être affinées en contraignant les éléments focaux sur les surfaces non masquées. De même, à défaut d’informations contraires, nous pouvons privilégier l’hypothèse selon laquelle les cibles se trouvent sur une des routes de la scène. Ces informations topographiques peuvent être modélisées de manière élégante au moyen de BBAs.

Les a priori topographiques sont de deux types. Il y a d’une part les informations de masquages telles que les bâtiments ou les murs (figure 3.6) et d’autre part les informations privilégiant certaines localisations (figure 3.7) telles que les routes ou des zones goudronnées. Les BBAs correspondant à ces deux types d’informations sont construites selon des approches différentes.

La forme d’une BBAs de masquage dépend à la fois des obstacles et de la position du capteur. Pour chaque élément d’obstacle se trouvant dans le champ de vision du capteur, une BBA est construite. Celle-ci peut être catégorique si l’obstacle est totalement opaque ou composée de deux éléments focaux s’il est plausible que la cible soit détectée à travers l’obstacle. Toutes les BBAs d’obstacles sont ensuite combinées par une règle conjonctive pour former la BBA de masquage.

Contrairement aux éléments de masquage, le deuxième type d’a priori ne dépend pas de la position des capteurs. L’objectif est de favoriser le positionnement des détections, et donc des pistes sur les routes, sans exclure la possibilité que la cible puisse se trouver à côté. Les BBAs sont donc des BBAs simples dont l’élément focal différent de $\Omega$ correspond à la route. Lorsque plusieurs éléments de ce type sont
présents, leurs BBAs respectives sont combinées de manière disjonctive puisque, par exemple, la présence de deux routes encourage la présence de la cible sur l’un des deux tronçons et pas spécifiquement sur l’intersection.

La BBA des masquages et la BBA des routes sont alors combinées conjointement afin de construire une BBA topographique indépendante des détections. Cette BBA d’a priori peut à son tour être combinée avec les BBAs de détections afin d’en préciser la localisation.

3.3.3 Conditionnement

Étant donné que chaque capteur ne scrute qu’une petite partie de la scène, seules les cibles se trouvant dans cette zone limitée peuvent générer des détections. La fonction de croyance qui en résulte peut donc être confinée à un cadre réduit Ωₜ dont la position relative par rapport au cadre général Ω de la scène varie pour chaque détection. Ce changement de cadre peut être vu comme un conditionnement :

\[ m^{Ωₜ} = m^{Ω}[Ωₜ] \]  

(3.1)

Ce conditionnement est purement formel puisque le complémentaire de Ωₜ n’est pas plausible \( p(Ωₜ) = 0 \) mais il permet de réduire considérablement la taille nécessaire pour représenter la BBA. Notons que cette représentation contribue également à la scalabilité du système car sa taille ne dépend pas de la taille de la zone surveillée.
3.3.4 Changement d’échelle

La surface sur laquelle s’étend une BBA varie également considérablement en fonction du capteur responsable de la détection. Le choix du pas de discrétisation de la scène doit alors assurer un compromis entre la précision et la taille des surfaces représentables, ce qui peut s’avérer délicat. Afin d’exploiter au mieux les informations de tous les capteurs coexistants, un facteur d’échelle variable $s_i$ est adjoint à chaque BBA, de sorte que le cadre de discernement $\Omega^s_i$ soit de la forme :

$$m^{\Omega^s_i} = (m^{\Omega_s}[\Omega^s_i])^{\Omega^s_i}$$

Grâce à l’introduction de ce rapport d’échelle variable, chaque BBA peut être définie sur un cadre de discernement ayant une résolution spatiale différente. Ce gain de flexibilité nécessite toutefois de ramener les BBAs définies sur $\Omega^s_{d_1}$ et $\Omega^s_{d_2}$ sur le même cadre de discernement $\Omega^{s_{d_{12}}}$ avant de les combiner. L’étendue spatiale $d_{12}$ de ce cadre est imposée par la règle de combinaison et nous choisirons le plus petit facteur $s_{d_{12}}$ permettant de représenter les deux BBAs sur $\Omega_{d_{12}}$. Celles-ci sont alors ramenées sur $\Omega^{s_{d_{12}}}_{d_{12}}$ et $\Omega^{s_{d_{12}}}_{d_{12}}$ par conditionnement et ensuite sur $\Omega^{s_{d_{12}}}_{d_{12}}$ par raffinement ou grossissement (figure 3.8).

![Fig. 3.8: Changement d’échelle par grossissement](image)

Dans le cas d’une combinaison conjonctive, le cadre de discernement est défini par $\Omega_{d_{12}} = \Omega_{d_1} \cap \Omega_{d_2}$. Par conséquent, le facteur $s_{d_{12}}$ est plus petit ou égal à $s_1$ et $s_2$, et le changement d’échelle est le raffinement suivant :

$$m^{\Omega^s_{d_{12}}} = \left( m^{\Omega^s_{d_i}}[\Omega^s_{d_{12}}] \right)^{\Omega^s_{d_{12}}} \quad i \in \{1, 2\}$$

Au contraire, dans le cas d’une combinaison disjonctive, le cadre de discernement vaut $\Omega_{d_{12}} = \Omega_{d_1} \cup \Omega_{d_2}$. Le facteur $s_{d_{12}}$ est alors plus grand ou égal à $s_1$ et $s_2$, et le changement d’échelle est le grossissement suivant :

$$m^{\Omega^s_{d_{12}}} = \left( m^{\Omega^s_{d_i}}[\Omega^s_{d_{12}}] \right)^{\Omega^s_{d_{12}}} \quad i \in \{1, 2\}$$

En reprenant l’analogie entre un élément focal et un masque défini sur une grille (cf. paragraphe 3.2.1), le conditionnement revient à spécifier la position et la taille d’une sous grille suffisante pour décrire tous les éléments focaux relatifs à une détection et le facteur d’échelle permet d’en adapter la résolution.
3.4 Fusion des pistes et des détections

L’ensemble des opérations présentées dans cette section réalise le filtrage récursif de mise à jour des pistes dans le contexte du TBM. La fusion comprend donc les étapes de prédiction et de correction des BBAs, mais également l’estimation pignistique de valeurs numériques à partir du modèle des pistes.

3.4.1 Règle de combinaison

Ayant adopté le formalisme des fonctions de croyance, la mise à jour de la localisation d’une piste est assez naturelle. Dans la mesure où il s’agit de combiner deux BBAs (la position de la piste et la position de la détection), cette opération est entièrement définie par le choix d’une règle de combinaison. Parmi les nombreuses règles existantes, deux options ont été envisagées : la règle conjonctive TBM et la règle conjonctive prudente.

Le choix de ces règles est motivé par leur justification théorique solide, par leur comportement conjonctif et par leur associativité. La propriété conjonctive est en effet nécessaire pour affiner la localisation de la piste par l’apport de nouvelles informations et l’associativité permet d’éviter l’ambiguïté causée par deux détections simultanées.

Règle conjonctive TBM

La justification théorique de la règle conjonctive TBM, ou de son équivalent normalisé (Dempster), constitue un argument en sa faveur. Cette règle de combinaison est d’ailleurs probablement la plus utilisée et toute solution alternative doit presque nécessairement être comparée à celle-ci.

En revanche, l’hypothèse d’indépendance des sources peut être sujette à discussion. En effet, si l’on peut raisonnablement considérer que l’erreur de localisation est indépendante pour deux capteurs différents, il n’en va pas nécessairement de même pour les détections successives d’un même capteur. Une dépendance peut alors apparaître entre la piste (construite par agrégation d’anciennes données) et la nouvelle détection.

Règle conjonctive prudente

L’usage de la règle prudente peut être justifié par le fait que les détections provenant d’un seul capteur sont corrélées temporellement. Ce choix présente également l’avantage de rendre les valeurs des masses (ou des $w$) attribuées à chaque élément focal moins sensibles à la fréquence temporelle à laquelle les détections sont fusionnées. Ce choix peut donc être intéressant, mais il nécessite de prendre quelques précautions. Cette combinaison impose en effet que les BBAs soient non dogmatiques. Or, ce n’est généralement pas le cas, ni pour les détections ni pour les pistes puisqu’elles sont toutes deux localisées sur une petite portion de la scène. Pour réaliser cette combinaison, nous sommes donc contraints de conditionner préalablement les deux BBAs sur l’intersection $\Omega_d \cap \Omega_p$, où $\Omega_d$ et $\Omega_p$ sont les unions des
éléments focaux respectivement de la détection et de la piste. En toute généralité, ce conditionnement ne nous garantit pas encore que les BBAs soient non dogmatiques sur $\Omega_d \cap \Omega_p$. Ce sera toutefois vrai si les détections sont consonantes et que l’état d’une piste est construit par combinaison conjonctive de détections.

**Propriété 3.1.** Toute BBA $m^\Omega$ formée par des combinaisons conjonctives TBM de BBAs consonantes possède un élément focal incluant tous les autres :

$$\exists A \subseteq \Omega, \ m^\Omega(A) > 0 \text{ tel que } \forall B \subseteq \Omega \ m^\Omega(B) > 0 \Rightarrow B \subseteq A \quad (3.5)$$

*Preuve.* Supposons que deux BBAs $m_1^\Omega$ et $m_2^\Omega$ aient chacune un élément focal incluant tous les autres. Ces éléments sont notés respectivement $B_{\text{max}}$ et $C_{\text{max}}$.

Dans ce cas, $m_1^\Omega \cap m_2^\Omega$ a un élément focal sur $A_{\text{max}} = B_{\text{max}} \cap C_{\text{max}}$ dont la masse vaut $m_1^\Omega(B_{\text{max}})m_2^\Omega(C_{\text{max}})$. De plus, $A_{\text{max}}$ inclut tous les autres éléments focaux puisque par définition :

$$m_1 \cap m_2(A) = \sum_{B \cap C = A} m_1(B)m_2(C) \quad \forall A \subseteq \Omega$$

et

$$B \subseteq B_{\text{max}}, C \subseteq C_{\text{max}} \Rightarrow B \cap C \subseteq B_0 \cap C_{\text{max}} = A_{\text{max}}$$

Puisque les BBAs consonantes vérifient l’hypothèse de départ, la propriété est également démontrée par récurrence pour toute BBA formée par la combinaison conjonctive de BBAs consonantes.

Cette propriété est aussi vérifiée par la combinaison conjonctive prudente puisque le résultat de cette combinaison est nécessairement non dogmatique. Il en va de même pour la règle de Dempster car, bien que n’étant pas conjonctive, les éléments focaux de celle-ci sont définis sur les mêmes ensembles que la règle conjonctive TBM (à l’exception de l’ensemble vide).

Notons enfin que le calcul de la règle prudente à partir de fonctions de masse est relativement lourd, car il exige de calculer la décomposition canonique. Cependant, si nous opting pour cette combinaison, les structures de croyance des pistes peuvent être maintenues sous la forme des $w$. Cette représentation équivalente aux fonctions de masse permet de préserver la même complexité de calcul pour les deux combinaisons.

**Choix de la règle**

Le choix parmi l’une des deux règles envisagées dépend avant tout de l’autocorrelation des sources. Ce facteur est bien entendu lié à la nature des capteurs, mais aussi à des éléments secondaires tels que l’organisation de transmissions d’informations. Plusieurs capteurs transmettent en effet leurs messages par polling. C’est-à-dire qu’ils répondent à des requêtes d’un système maître. Cette stratégie implique que si les demandes sont trop fréquentes, une détection peut être transmise plusieurs
fois au module de fusion. L’erreur de localisation des capteurs est également une information corrélée puisque cette incertitude ne change pas au cours du temps.

En conclusion, aucun argument ne permet de trancher pour une de ces deux règles dans l’absolu. Comme son nom l’indique, la deuxième option est peut-être plus prudente.

3.4.2 Prédiction

Avant de fusionner la position d’une piste avec une nouvelle détection, il faut projeter la piste au temps \( t \) de la nouvelle détection. Cette étape, comparable, au moins sur le principe, à la prédiction d’un filtre de Kalman, doit intégrer l’augmentation de l’imprécision survenue entre deux observations et peut aussi prendre en compte un modèle de déplacement des pistes.

Transformation de généralisation

Dans notre cas, l’opération va consister en une modification des fonctions de croyance de la localisation. Si une cible suit une trajectoire erratique, aucune prédiction ne peut être faite sur le changement survenu entre deux observations successives. Il est alors préférable de n’appliquer aucun modèle de déplacement. Dans ce cas, les positions possibles de la piste sont prédites en *dilatant* chaque élément focal d’une taille dépendant de la vitesse maximale admissible pour la piste et du délai entre la dernière mise à jour et la nouvelle détection. Cette approche simple illustrée par la figure 3.9 permet de garantir \( m_{t-1} \subseteq m_t \), aucune information arbitraire n’a donc été introduite. De plus, puisqu’aucun élément focal supplémentaire n’a été créé, la complexité des structures manipulées n’augmente pas.

![Figure 3.9: Généralisation des masses de t-1 vers t](image)

**Fig. 3.9:** Généralisation des masses de t-1 vers t

Modèle de trajectoires linéaires

Un modèle linéaire peut également être intégré à la prédiction si les déplacements des cibles correspondent à de telles trajectoires. Cette hypothèse peut par exemple être raisonnable lors de la détection de véhicules à une fréquence supérieure à 1 Hz. En effet, les possibilités de variations de direction et de vitesse sont sufisamment...
faibles à cette fréquence pour approcher localement les trajectoires par des segments de droites.

Les éléments focaux sont alors dilatés comme décrit plus haut, mais également déplacés autour de la prédiction de la position. Le modèle introduit donc une information qui repose sur une hypothèse a priori de déplacement.

Il faut enfin noter que l'utilisation d'un tel modèle suppose une vitesse de déplacement. Ce paramètre ne peut pas toujours être estimé de manière fiable si les capteurs ne sont pas suffisamment précis.

### 3.4.3 Rétroprojection des pistes

Tout comme l'état d'une piste à l'instant $t$ est estimé à partir de la combinaison des détections courantes et de la projection à $t$ de l'état antérieur, il est possible de corriger les localisations antérieures de la piste en projetant dans le passé la nouvelle détection. Dans ce cas, une première estimation incertaine (figure 3.10) suivie d'une deuxième plus précise (figure 3.11) est corrigée en reprojétant dans le passé la nouvelle détection (figure 3.12).

Dans les systèmes de surveillance, la localisation est généralement exploitée en temps réel. Lorsqu'une position plus récente est disponible, la correction des anciens états n'est donc pas directement utilisable. Cependant, cette démarche permet de corriger indirectement l'estimation de variables utiles pour les moteurs d'alarmes telles que la vitesse ou la distance parcourue.

La rétroprojection des détections peut être comparée au contracteur utilisé en analyse par intervalle. Toutefois, comme les éléments focaux ne sont pas approchés par des boîtes englobantes, il n'est pas nécessaire d'effectuer des projections et rétro-projections successives jusqu'à convergence des corrections. La géométrie des éléments focaux restera inchangée après une seule itération.

![Première localisation](image)
3.4.4 Mise à jour des paramètres

Estimation de la position

Bien que les BBAs permettent de maintenir une description très riche de la connaissance, un observateur ou un système expert va généralement exiger que la localisation soit transmise sous la forme d’une position dans un repère métrique ou cartographique. Dans ce cas, la mise à jour de la fonction de croyance doit être suivie d’une prise de décision. Or, bien souvent, plusieurs hypothèses auront la même probabilité pignistique \( \text{BetP} \). Étant généralement équivalentes en terme de fonctions de croyance, aucun traitement crédibiliste ne permettra de les distinguer. En adoptant une démarche comparable au PDAF, nous pouvons toutefois tirer parti des positions spatiales de ces hypothèses.

Cette idée est développée dans [56] pour un modèle évidentiel de localisation par intervalles. La position est estimée par maximisation de l’espérance pignistique [60]:

\[
\mathbb{E}(m) = \sum_{i=1}^{p} m_i c_i \tag{3.6}
\]
où $c_i$ est le centre de l'intervalle $i$, $m_i$ sa masse et $p$ le nombre d'intervalles.

Cette méthode peut être généralisée à notre modèle par :

$$
\mathbb{E}(m) = \sum_{A \subseteq \Omega} m(A)c_A
$$

(3.7)

si $c_A$ est le centre de gravité de l'élément focal $A$.

L'approche peut aussi n'être appliquée qu'aux hypothèses maximisant la probabilité pignistique ou la plausibilité.

$$
\mathbb{P}_{\text{Betp}}(\omega) = \sum_{\omega_i \in \text{max}(\text{BetP}(\omega))} c_{\omega_i}
$$

(3.8)

$$
\mathbb{P}_{\text{pl}}(\omega) = \sum_{\omega_i \in \text{max}(\text{pl}(\omega))} c_{\omega_i}
$$

(3.9)

où $c_{\omega_i}$ est le centre de gravité de l'hypothèse $\omega_i$. Cette deuxième solution ne tient donc compte que des hypothèses considérées comme les plus probables.

### Identification

Lorsque les capteurs sont capables de classer les détections, la fusion peut combiner les informations relatives à une même piste afin de produire une classification multi capteurs. Cette tâche est indépendante de la localisation, mais dans la mesure où elle a été implémentée dans l'application qui sera présentée au chapitre 7, nous la décrivons ici brièvement.

Une BBA simple est tout d'abord générée à partir de chaque classification. Cette BBA, dont l'élément focal différent d'$\Omega$ est l'hypothèse renseignée par la classe, est ensuite affaiblie en fonction de la confiance accordée au capteur. Toutes les BBAs relatives à un même capteur sont combinées de manières prudentes et les BBAs résultantes sont à leur tour fusionnées par la règle conjonctive $\text{TBM}$. Enfin, la décision est prise en maximisant la probabilité pignistique.

Notons que cette méthode simple s'appuie sur l'hypothèse qu'une cible ne change jamais de classe. Elle exclut donc, par exemple, le cas d'un groupe qui se sépare.

### 3.5 Analyse de sensibilité

Avant d'implanter dans un système réel la fusion présentée dans ce chapitre, nous avons effectué quelques tests préliminaires pour évaluer la sensibilité de la méthode à plusieurs paramètres et sources de bruits. Ces essais ont été réalisés en comparant la position d'une cible virtuelle avec la localisation obtenue par fusion de détections provenant de capteurs simulés. Dans un premier temps, un seul capteur a été intégré aux simulations de sorte que les expériences portaient sur la fusion de détections successives d'une seule source. Une deuxième série de tests a ensuite été réalisée en ajoutant un deuxième capteur.

Précisons dès à présent que les simulations présentées dans cette section ont été réalisées avec des combinaisons conjonctives $\text{TBM}$. Certains essais ont également
été effectués avec la combinaison prudente mais celle-ci a produit sensiblement les mêmes résultats.

### 3.5.1 Localisation mono capteur

Le principal objectif des simulations ne faisant intervenir qu’un seul capteur a été d’évaluer dans quelle mesure la fusion avec une détection antérieure peut améliorer la précision de la localisation.

#### Présentation du dispositif expérimental

Les expériences ont été réalisées en disposant un capteur sur une scène virtuelle et en simulant le déplacement d’une cible dans la zone surveillée. La première position de cette cible est tirée de manière aléatoire et la seconde est le résultat d’un déplacement d’une distance prédéfinie dans une direction aléatoire. Cette deuxième position constitue la vérité terrain que nous allons tenter d’approcher par la fusion.

Les détections sont simulées en ajoutant un bruit gaussien aux positions relatives de la cible par rapport au capteur (figure 3.13). Selon les caractéristiques du capteur, ces détections peuvent contenir une estimation de la distance et/ou de la direction relative.

Une BBA est d’abord formée à partir de la première détection et est projetée au temps de la deuxième détection. Les deux BBAs sont ensuite combinées et la localisation obtenue est comparée à la position réelle de la cible. Enfin, pour s’affranchir des effets cumulés de plusieurs combinaisons successives, la simulation est réinitialisée avant de procéder au test suivant.

![Fig. 3.13: Simulation d’un pistage mono capteur : détection et BBA](image)

Ces premiers essais ont été réalisés avec un capteur d’une portée de 120m, d’une ouverture angulaire de 60° et dont les détections comportent à la fois une estimation de la distance et de l’orientation des cibles.

#### Détections sans déplacement de la cible

Lorsque la cible ne se déplace pas entre les deux détections, la fusion aura un effet *moyenneur* sur les erreurs de détection et la fusion aura toujours un effet bénéfique
Analyse de sensibilité

sur la localisation.

La figure 3.14 illustre ce principe pour deux constructions différentes de BBAs. Les courbes indiquent l’erreur moyenne de localisation en fonction de l’écart-type des erreurs de détection, ces écarts-types étant exprimés en mètre pour la distance et en degré pour la direction.

Les courbes bleues présentent le cas de BBAs engagées constituées de trois éléments focaux consonants formés à partir de deux expansions successives de 5° et 5 m de l’élément focal le plus engagé. Les courbes rouges illustrent le cas moins engagé caractérisé par des facteurs d’expansion de 15° et 15 m. Pour chacun de ces deux cas de figure, la courbe en pointillé indique l’erreur observée lorsque la position est estimée à partir d’une seule détection et la courbe continue représente l’erreur obtenue après fusion conjonctive.

Nous pouvons observer que la courbe continue est toujours située en dessous de la courbe discontinue de la même couleur, ce qui traduit le fait que la fusion améliore toujours la localisation. Nous pouvons également noter que les courbes continues se croisent, indiquant que la fusion sera meilleure si l’engagement des BBAs est bien dimensionné : l’erreur de localisation est plus faible avec des BBAs engagées tant que les capteurs sont précis mais la tendance s’inverse lorsque l’écart-type des détections dépasse 10 m et 10°.

Influence de la vitesse

Lorsque la cible se déplace entre les deux détections, l’estimation de la localisation après fusion n’est plus nécessairement meilleure. En effet, si le capteur est très précis, une seule observation donne déjà de très bons résultats et la prise en compte de la première détection va tirer l’estimation vers la position antérieure. Nous pouvons toutefois observer sur la figure 3.15 que cet effet parasite est compensé à la fois pour des BBAs engagées (en bleu) et pour des BBAs prudentes (en rouge) dès que les écarts-types des détections sont supérieurs à la moitié du déplacement. De plus, l’erreur de localisation est de moins en moins sensible au déplacement lorsque les capteurs deviennent imprecis.
Généralisation de l’état antérieur

Les simulations présentées jusqu’à présent ont été effectuées sans généraliser la première détection. Les BBAs modélisent donc à la fois l’imprécision des détections et l’incertitude liée au déplacement de la cible. Nous pouvons espérer obtenir de meilleurs résultats en traitant les deux détections de manière asymétrique : grâce à la généralisation, la BBA plus ancienne est désengagée et ne contribue donc que dans une moindre mesure à l’estimation de la localisation. L’effet est illustré sur la figure 3.16 sur laquelle chaque courbe représente l’erreur moyenne de localisation en fonction du déplacement de la cible pour différentes précisions de capteurs. Les courbes continues représentent l’erreur lorsque les BBAs sont projetées en dilatant les éléments focaux de 10 m dans toutes les directions et les courbes en pointillés représentent l’erreur sans projection pour les mêmes caractéristiques de capteur.

**Fig. 3.16:** Effet de la projection sur la localisation

Nous observons que la projection a un effet stabilisant sur l’erreur de localisation. Lorsque le capteur est précis et la projection bien dimensionnée, l’ensemble de l’information est incluse dans la deuxième détection et la fusion va se comporter...
comme s’il n’y avait qu’un seul capteur. Par conséquent, l’effet attractif de la pre-
mière détection est évité et l’erreur de localisation reste constante. Au contraire, si
les capteurs sont moins précis, la projection n’est plus l’élément prépondérant et la
fusion se rapproche du cas de deux BBAs symétriques. Comme nous l’avons déjà
observé sur la figure 3.15, l’erreur moyenne est encore constante, et elle est inférieure
table obtenue avec un seul capteur (22 m).

3.5.2 Localisation multi capteurs

Ces essais ont avant tout été réalisés pour tester la fusion avec des capteurs
directionnels. Ceux-ci, très courants en pratique, ne peuvent estimer la localisation
des cibles que par recoupement des détections.

Présentation du dispositif expérimental

Pour réaliser ces simulations, un deuxième capteur a été ajouté au dispositif
initial. Les deux capteurs, d’une portée de 120 m et d’une ouverture angulaire de
60° ont d’abord été orientés l’un vers l’autre et placés aux deux coins opposés d’un
carré de 100 m de côté. Cette configuration a été testée avec des capteurs capables
d’estimer la distance ou non. Ensuite, les capteurs directionnels ont été testés selon
une disposition plus favorable en les plaçant sur deux coins adjacents du carré comme
illustré sur la figure 3.17.

![Fig. 3.17: Configuration de deux capteurs](image)

Chaque position de la cible a cette fois été tirée aléatoirement et sans corrélation
temporelle de sorte que seule la fusion des détections simultanées a été testée. Le
choix de ne pas tester la fusion des détections consécutives a été motivé par la volonté
de limiter les paramètres de la simulation.

Détections simultanées avec estimation de la distance

Nous avons tout d’abord testé la fusion de deux détections simultanées provenant
de capteurs capables d’estimer la distance. Cette première simulation, dont la dispo-
sition correspond au premier schéma de la figure 3.17 donne des résultats comparables
table fusion de deux détections d’une cible qui ne se déplace pas et provenant d’un
seul capteur (cnf. 3.5.1).
Comme lors des simulations ne faisant intervenir qu’un seul capteur, les courbes bleues représentent des BBAs engagées constituées de trois éléments focaux consonants caractérisés par un facteur d’expansion de 5° et 5 m. De même, les courbes rouges illustrent un cas moins engagé défini par un facteur d’expansion de 15° et 15 m. Pour chacun de ces deux cas de figure, la courbe en pointillé indique l’erreur observée lorsque la position est estimée à partir d’une seule détection et la courbe continue représente l’erreur obtenue après fusion conjonctive.

Les courbes pleines se croisent, confirmant le fait qu’un bon dimensionnement des BBAs améliore la localisation. Contrairement au cas des détections consécutives, nous pouvons également remarquer que la fusion de deux détections de capteurs précis mais modélisées par des BBAs peu engagées (courbe rouge), génère une localisation moins bonne que celle produite à partir d’une seule détection. Cet effet est toutefois très marginal et est dû à des erreurs d’approximation spatiale introduites par la fusion.

**Choix de la position relative des capteurs**

La fusion a ensuite été testée avec des capteurs n’estimant que la direction des détections. Dans ce cas, c’est avant tout le positionnement des capteurs qui détermine la qualité des résultats. En effet, la localisation des cibles situées sur l’axe passant par les deux capteurs ne pourra pas être retrouvée par recoupement des détections. La fusion donne donc très logiquement de meilleurs résultats lorsque cet axe est situé en dehors du champ des capteurs. Ces dispositions défavorables et favorables correspondent respectivement au deuxième et au troisième schéma de la figure 3.17 et les erreurs de localisation correspondantes sont reportées sur la figure 3.19. Notons que cette fois, l’erreur d’estimation de la position à partir d’une seule BBA est indépendante de son engagement. Par soucis de clarté, une seule courbe a donc été tracée.

Le premier graphique de la figure 3.19 montre qu’un engagement excessif des BBAs donne de mauvais résultats lorsque des capteurs imprecis sont dans une configuration défavorable. L’estimation de la localisation de cible proche de l’axe passant par les deux capteurs est en effet très sensible à une erreur de direction.
3.5.3 Synthèse de l’analyse

En conclusion, cette analyse de sensibilité a montré que la fusion des détections améliore généralement la localisation des cibles. Cette amélioration a été observée à la fois lors de deux détections successives par un même capteur et lors de deux détections simultanées par des capteurs différents. De même, bien que la localisation obtenue soit sensible à l’engagement des BBAs, la fusion est bénéfique quel que soit l’engagement. De plus, la projection des détections antérieures et la disposition judicieuse des capteurs se sont également avérées être deux éléments contribuant à la qualité des localisations.

Deux cas défavorables ont toutefois été mis en évidence. La fusion de détections très précises mais modélisées par des BBAs trop peu engagées peut donner un résultat moins bon que les détections initiales. De même, la fusion de deux détections très imprécises mais modélisées par des BBAs trop engagées peut, dans certains cas défavorables, produire de mauvais résultats.

Il faut enfin noter que ces remarques ne concernent que la qualité de la localisation. Parallèlement à celle-ci, le suivi sans interruption de la piste constitue un autre objectif important de la fusion et impose de rester modéré dans l’engagement des BBAs.

3.6 Conclusions

La représentation des incertitudes de localisation présentée dans ce chapitre constitue une première contribution ainsi que la base sur laquelle le reste de ce travail va s’appuyer. Cette représentation au moyen du modèle des croyances transférables (TBM) offre plusieurs avantages déterminants pour la fusion multi capteurs. Elle permet tout d’abord de modéliser des détections ayant des formes d’incertitude très différentes. Les *a priori* topographiques peuvent aussi facilement être intégrés en les modélisant par des BBAs. Enfin, l’assimilation d’une nouvelle détection est réalisée par un traitement de généralisation et une combinaison, deux traitements qui respectent le principe de moindre engagement et qui sont bien définis dans la TBM.
Malgré ces avantages, la taille du cadre de discernement constitue un obstacle qui s’oppose à l’exploitation de la $TBM$ pour cette application. Pour contourner cette difficulté, les détections sont contitionnées sur une petite région de la scène et un facteur d’échelle est ajouté par grossissement. Avant d’être combinées, deux $BBAs$ sont alors ramenées sur un même cadre de discernement par conditionnement et raffinement.
Chapitre 4

Suivi multi pistes

Sommaire

4.1 Association détections-pistes : généralités .................................. 68
4.1.1 Stratégies d’association .................................................. 68
4.1.2 Notations et hypothèses de travail ..................................... 68
4.1.3 Méthodes de résolution ..................................................... 70
4.2 Traitements dans le cadre du TBM ........................................... 72
  4.2.1 BBAs d’association ...................................................... 72
  4.2.2 Coûts crédibilistes ....................................................... 73
  4.2.3 Plausibilité globale ...................................................... 75
  4.2.4 Prise en compte de l’engagement ..................................... 77
4.3 Résultats ........................................................................... 80
4.4 Conclusion .......................................................................... 84
4.1 Association détections-pistes : généralités

La méthode de pistage présentée dans le chapitre précédent propose de constituer une piste en combinant des détections modélisées par des BBAs. Elle repose donc implicitement sur l’hypothèse que toutes les informations se rapportent à une même cible. En pratique, plusieurs cibles peuvent cependant être présentes au même moment dans la scène. Il est donc nécessaire de procéder à une mise en correspondance entre les détections et les pistes avant de les combiner. Une nouvelle piste n’est alors créée que si une détection ne peut être associée à aucune piste existante. L’objectif sous-tendant cette démarche est de générer le nombre minimum de pistes nécessaires pour expliquer toutes les détections [3].

4.1.1 Stratégies d’association

L’association entre les pistes et les détections peut être envisagée selon deux stratégies. La première solution, suggérée notamment dans les systèmes hiérarchiques (1.6.1), repose sur un traitement en deux étapes. Les détections provenant simultanément des capteurs sont combinées entre elles avant d’être associées aux pistes.

La solution alternative consiste à traiter chaque capteur l’un après l’autre en associant directement ses détections avec les pistes. Le principal inconvénient de cette seconde approche est que le résultat de l’association peut varier selon l’ordre dans lequel les capteurs sont traités. La mise à jour des pistes va en effet modifier leur état de sorte que l’association suivante peut en être affectée. Cette solution permet en revanche de traiter les détections à la volée sans nécessiter de synchronisation. Elle est en outre particulièrement pertinente pour fusionner les données de capteurs travaillant à des fréquences différentes. Ces deux arguments s’étant avérés décisifs, c’est cette deuxième option qui a été retenue.

4.1.2 Notations et hypothèses de travail

Selon la stratégie choisie, la méthode d’association peut concerner uniquement les détections ou faire aussi intervenir les pistes. Toutefois, ces objets étant modélisés de manière identique, les deux cas peuvent être étudiés sans distinction.

Soit une mesure (ou fonction de coût) capable de comparer les différentes associations possibles. Si l’ensemble des pistes est noté $\mathcal{P} = \{p_1, \ldots, p_m\}$ et l’ensemble des détections $\mathcal{D} = \{d_1, \ldots, d_n\}$, une association $A_l$ peut être vue comme un ensemble d’arêtes $A_l = \{a_1, \ldots, a_q\}$ de la forme $a_k = (p_i, d_j)$ liant une piste $p_i$ et une détection $d_j$. L’objectif est alors de rechercher parmi l’ensemble des associations $A_l \in \mathcal{A}$, la solution $A_0$ qui minimise une fonction de coût $C(A_l)$:

$$A_0 = \min_{A_l \in \mathcal{A}} (C(A_l))$$

La résolution de ce problème dépend de l’expression de la fonction de coût $C(A_l)$ ainsi que d’éventuelles contraintes limitant les associations contenues dans $\mathcal{A}$. 
Contraintes d’associations

La nature des pistes et des détections peut justifier l’introduction de certaines contraintes d’association.

Si nous supposons que les capteurs ne fragmentent pas les cibles et qu’aucun groupe d’intrus ne se sépare, une piste ne peut alors pas être associée à plus d’une détection. Si en outre, nous ajoutons la contrainte duale interdisant l’association des détections à plus d’une piste, nous obtenons des associations de type 1-1. Notons que la notation 1-1 n’exclut pas les pistes et les détections isolées.

Lorsqu’au contraire, la possibilité de regroupements et de séparations de pistes est prise en compte, il faut introduire une contrainte moins forte. Nous pouvons alors imposer à chaque association de ne pas contenir à la fois plusieurs pistes et plusieurs détections. Cette limitation ne se justifie pas par la nature des éléments traités mais elle permet d’éviter certaines solutions dégénérées telle que l’association de toutes les pistes avec toutes les détections. Dans ce cas, les associations sont de type 1-N (figure 4.1).

![Fig. 4.1: Associations 1-1, 1-N et M-N](image)

Coûts additifs

Cette propriété, intéressante pour la mise en œuvre de certains algorithmes de recherche de minimum est définie comme suit. Le coût total d’une association \( C(A_l) \) est dit additif s’il est le résultat de la somme de coûts élémentaires \( c(a_k) \geq 0 \) attribués à chaque arête \( a_k = (p_i, d_j) \). Dans ce cas, l’association peut être vue comme un problème d’affectation de graphe biparti. Les pistes et détections non associées peuvent en outre se voir attribuer un coût fixe \( c_p \) et \( c_d \), de sorte que :

\[
C(A_l) = \sum_{a_k \in A_l} c(a_k) + n_p c_p + n_d c_d
\]

où \( n_p \) et \( n_d \) sont les nombres de pistes et de détections n’appartenant à aucune arête \( a_k \in A_l \) :

\[
n_p = |p_i| \ \forall a_k = (p_i, d_j) \in A_l
\]

\[
n_d = |d_j| \ \forall a_k = (p_i, d_j) \in A_l
\]

**Remarque** : Afin de ne pas surcharger la notation, le coût local \( c(a_k) \) associé à une arête \( a_k = (p_i, d_j) \) sera par la suite noté \( c(p_i, d_j) \).
4.1.3 Méthodes de résolution

Le choix d’une méthode spécifique pour la résolution du problème d’association dépend des hypothèses et des contraintes. Par exemple, lorsque les localisations des pistes et les mises en correspondance peuvent être représentées sous forme probabiliste, l’approche classique consiste à mettre en place un filtrage de type PDAF ou JPDAF [5].

Dans un contexte non probabiliste, la définition d’une fonction de coût caractérisant les associations va constituer le cœur du problème. La plupart du temps, cette fonction va dépendre de la proximité spatiale des pistes et des détections, mais une fonction *ad hoc* tenant compte d’autres paramètres tels que des *descripteurs* de la piste peut aussi être envisagée.

Avant d’aborder cette question, commençons par présenter les principales méthodes permettant de minimiser différents types de fonctions de coût. Dans ce contexte, rappelons également que le choix d’un coût additif peut faciliter la recherche du minimum. Il est toujours possible de tester toutes les associations, mais cet avantage peut devenir crucial lorsque le nombre de pistes et de détections devient important. En effet, en l’absence de contraintes, une recherche exhaustive doit examiner $2^{m \times n}$ associations différentes (où $m$ est le nombre de pistes et $n$ le nombre de détections).

**Associations 1-1**

Lorsque les hypothèses de travail permettent de limiter la recherche aux seules associations de type 1-1 et lorsque les coûts d’association sont additifs, la minimisation peut être réalisée de manière efficace à l’aide d’une des variantes de l’algorithme hongrois [54] [38].

En cas de situation ambiguë, plusieurs solutions d’associations peuvent éventuellement être gardées provisoirement selon une approche de type *MHT*. Dans ce cas, l’algorithme de Murty [55] permet de trouver les $n$ associations de coûts minima.

**Associations 1-N**

La contrainte 1-1 est intéressante car elle réduit considérablement la taille de l’espace des solutions et elle permet la mise en œuvre d’algorithmes efficaces pour la recherche du minimum. Cependant, le bien fondé de cette contrainte peut être mis en défaut dans plusieurs cas pratiques. Beaucoup de capteurs sont, par exemple, incapables de compter les cibles. Ils ne produiront donc qu’une seule détection, et ce quel que soit le nombre d’intrus présents. De plus, même si un capteur est capable de dénombrement, deux cibles trop proches l’une de l’autre peuvent tantôt être détectées séparément tantôt être repérées par une seule détection. Dans ces conditions, une technique de type *graph cuts* [43] permet la recherche d’un minimum lorsque la contrainte 1-1 n’est pas vérifiée.

Il est également intéressant de noter que si le coût est additif et que les coûts élémentaires sont positifs, alors l’association de coût minimum sera nécessairement de type 1-N. Autrement dit, lorsqu’une arête $a_k = (p_i, d_j)$ appartient à l’association
optimale $A_0$, alors deux arêtes de type $a_m = (p_i, d_m)$ et $a_n = (p_n, d_j)$ (avec $d_m$ et $p_n$ quelconques) ne peuvent pas appartenir simultanément à $A_0$.

**Propriété 4.1.** Soit $A = \{A_1, ..., A_s\}$ l’ensemble des solutions d’associations de la forme $A_l = \{a_1, ..., a_q\}$ où chaque arête $a_k = (p_i, d_j)$ représente l’association entre une piste $p_i \in P$ et une détection $d_j \in D$, soit $A_0 \in A$ la solution d’association qui minimise une fonction de coût

$$C(A_l) = \sum_{a_k \in A_l} c(a_k) + n_p c_p + n_d c_d$$  \hspace{1cm} (4.5)

où $n_p$ et $n_d$ sont les nombres de pistes et de détections n’appartenant à aucune arête $a_k$, alors :

$$a_k = (p_i, d_j) \in A_0 \Rightarrow \nexists (a_m, a_n) \in A_2$$

Preuve. Supposons au contraire que l’association $A_1$ ne respecte pas cette propriété. Elle comporte dans ce cas trois arêtes de la forme : $a_k = (p_i, d_j)$, $a_m = (p_i, d_m)$ et $a_n = (p_n, d_j)$. L’association $A_2$ formée par la suppression de l’arête $a_k$ ne contient alors aucune piste ni détection supplémentaire. Par conséquent, son coût vaut $C(A_2) = C(A_1) - c(a_k)$ et est inférieur au coût de $A_1$ puisque $c(a_k) \geq 0$. Autrement dit, si une association ne respecte pas la propriété, il sera toujours possible de supprimer une arête sans isoler d’éléments. Le coût de la nouvelle association ainsi créée sera dès lors nécessairement inférieur.

Afin d’éviter la mise en place d’une méthode *graphs cuts* assez lourde, une solution approchée peut être estimée en commençant par rechercher la solution 1-1 et en tentant ensuite d’ajouter des arêtes pour lier les pistes et les détections restées isolées.

Bien que cette approche empirique ne garantisse pas un coût total $C$ optimal, elle peut toutefois donner de bons résultats car elle favorise les associations 1-1. Elle évite donc les paires d’agrégations de type 1-N et M-1. Cet effet parasite peut aussi être éliminé en introduisant un coût pénalisant les associations multiples, mais l’estimation numérique d’un tel coût est empirique et délicate.

**Comparaison des méthodes**

L’approche 1-N plus générale, permet de mettre en évidence des séparations et des regroupements de pistes qui se traduisent en terme d’association 1-1 par des créations et disparitions de piste. Ces informations peuvent être intéressantes pour un module de fusion de plus haut niveau.

Malheureusement, la possibilité de mettre en évidence ces comportements de groupes induit un risque de dédoubllement de pistes. Ce problème survient lorsque deux pistes sont associées par erreur à une même détection dont la localisation est très précise. Ces deux pistes vont alors se superposer et vont rester liées puisque leurs localisations vont être confondues. Dans le cadre de ce travail, les pistes produites par
la fusion sont destinées à être visualisées sur un poste de contrôle. Par conséquent, la stratégie 1-1 a été retenue après avoir observé cet effet parasite lors des premières expérimentations.

4.2 Traitements dans le cadre du TBM

Plusieurs travaux ont proposé d’exploiter le cadre des fonctions de croyance pour les problèmes d’association [3], [61], [52], [30] et [45]. Généralement, les pistes et les détections ne sont toutefois pas modélisées par des BBAs et ce formalisme est uniquement utilisé pour l’étape d’association [52], [30] et [45]. Les méthodes développées selon ce principe sont alors placées dans un système de pistage classique comportant par exemple un filtre de Kalman et un module de gestion des pistes. Cette première idée sera développée dans la section 4.2.1.

Lorsque les pistes et les détections sont modélisées par des BBAs, les coûts d’associations locaux peuvent être dérivés d’une mesure du conflit ou de la similarité entre BBAs [3]. Cette deuxième approche sera étudiée dans la section 4.2.2. Enfin, il est possible de définir un critère crédibiliste global [61]. Cette troisième possibilité, plus élégante d’un point de vue théorique sera traitée dans la section 4.2.3.

4.2.1 BBAs d’association

Cadres de discernement

La construction de BBAs d’association repose sur l’hypothèse d’associations de type 1-1. Trois cadres de discernement sont introduits [52].

- Le cadre

\[ \Omega_{i,j} = \{y_{i,j}, n_{i,j}\} \]  (4.7)

modèle la réponse à la question : “Oui ou non, la piste \( p_i \) est associée à la détection \( d_j \)?”. C’est dans ce cadre que sont exprimées les informations constituant le point de départ du raisonnement proposé.

- Pour pouvoir être combinées, les BBAs \( m_{\Omega_{i,j}} \) doivent ensuite être raffinées vers un cadre de discernement plus large. Ce cadre peut être

\[ \Omega_{p_i} = \{d_1, ..., d_j, \star\} \]  (4.8)

dont les hypothèses sont les réponses à la question : “A quelle détection est associée la piste \( p_i \)?”. L’hypothèse \( \star \) représente la non association de la piste.

- Alternativement, les \( m_{\Omega_{i,j}} \) peuvent aussi être raffinées vers

\[ \Omega_{d_j} = \{p_1, ..., p_j, \star\} \]  (4.9)

Ce deuxième cadre contient l’ensemble des réponses à la question : “A quelle piste est associée la détection \( d_j \)?”. De façon similaire au cas précédent, l’hypothèse \( \star \) représente la non association de la détection.
Présentation de la méthode

La première étape consiste à construire une BBA $m^{\Omega_{i,j}}$ pour chaque association possible entre une piste $p_i \in \mathcal{P}$ et une détection $d_j \in \mathcal{D}$, $m \times n$ fonctions de croyance sont ainsi construites.

Les $n$ BBAs $m^{\Omega_{i,1}} \ldots m^{\Omega_{i,n}}$ faisant intervenir la piste $p_i$ sont projetées dans le cadre de discernement $\Omega_{p_i}$ à l'aide d'un raffinement $m^{\Omega_{i,j}\Omega_{p_i}}$. De manière similaire, les $m$ BBAs $m^{\Omega_{1,j}} \ldots m^{\Omega_{m,j}}$ relatives à la détection $d_j$ sont transformées par un raffinement $m^{\Omega_{i,j}\Omega_{d_j}}$.

Bien que ce formalisme n’ait été introduit que dans Mercier et al. [52], des travaux antérieurs sous-entendaient la même approche [30] et [45].

Outre l’introduction d’un formalisme rigoureux, l’article [52] se distingue également par la partie combinatoire et par l’étape de décision. Selon cette approche, les BBAs $m^{\Omega_{p_i}}$, relatives à la piste $p_i$, sont fusionnées par des combinaisons conjonctives et les probabilités pignistiques $Bet^{\Omega_{p_i}}$ sont ensuite calculées. L’association retenue est alors celle qui maximise la probabilité jointe $Bet^{\Omega_{p_1} \times \ldots \times \Omega_{p_m}}$ tout en respectant la contrainte 1-1. De même, les $Bet^{\Omega_{d_j}}$ calculées à partir de la combinaison des $m^{\Omega_{d_j}}$ constituent un deuxième moyen de calculer une association. Il n’est malheureusement pas garanti que ces deux solutions soient équivalentes. Des travaux plus récents ont donc réévalué cette approche afin de traiter les pistes et les détections de manière symétrique. Ces travaux proposent alors de rechercher l’association la plus plausible après avoir exprimé toutes les croyances disponibles sur un seul cadre de discernement El Zoghby et al. [23].

Construction des BBAs d’association

La construction des BBAs $m^{\Omega_{i,j}}$ constitue la principale difficulté de la méthode présentée ci-dessus. En effet, la définition d’une fonction de coût est déjà souvent délicate, mais ici pour chaque correspondance entre une piste et une détection, il faut définir jusqu’à trois éléments focaux.

De plus, dans la mesure où dans notre cas les détections et les pistes sont déjà représentées par des BBAs, il est regrettable d’introduire un nouveau cadre de discernement spécifique à l’étape d’association et sans lien avec le cadre précédemment défini.

Bien que l’approche soit intéressante, ces deux arguments justifient le fait que cette solution n’ait pas été retenue. Notons toutefois que malgré ces inconvénients, la définition de BBAs spécifiques permet d’intégrer des informations hétérogènes pour faciliter l’association. Cette possibilité peut s’avérer un atout dans certaines applications.

4.2.2 Coûts crédibilistes

Les méthodes d’associations classiques présentées dans la section 4.1 nécessitent la définition d’une mesure ou une fonction de coût évaluant la pertinence de mettre à jour une piste existante avec une détection donnée.
Quel que soit le cadre d’analyse et sans tenir compte d’une quelconque forme d’incertitude de localisation, il est possible d’utiliser la distance euclidienne séparant la position de la détection et de la piste. Cette mesure, bien que simpliste, constitue une référence intéressante pour positionner toute solution alternative. De même, l’indicateur de référence pour une approche probabiliste serait la distance de Mahalanobis.

Dans notre cas, étant donné que les détections et les pistes sont toutes deux représentées par des fonctions de croyance, le coût d’association peut directement être évalué à partir de cette information. Une telle option présente alors l’intérêt de contribuer à la cohérence du système.

Distance de Jousselme

Dans cette optique, une première solution consiste à exploiter une distance entre BBAs telle que la distance de Jousselme :

\[ c_J(p_i, d_j) = \sqrt{\frac{1}{2} \left( \langle m_i, m_i \rangle + \langle m_j, m_j \rangle - 2 \langle m_i, m_j \rangle \right)} \]  

(4.10)

où

\[ \langle m_i, m_j \rangle = \sum_{A \subseteq \Omega} \sum_{B \subseteq \Omega} m_i(A)m_j(B) \frac{|A \cap B|}{|A \cup B|} \]  

(4.11)

et \( m_i \) et \( m_j \) sont les BBAs de \( p_i \) et \( d_j \) respectivement.

Une autre distance peut bien entendu être envisagée, mais il ne faut pas perdre de vue que ces distances ont pour objectif de quantifier la similitude entre deux BBAs. Cet objectif n’est pas orthogonal à la notion intuitive de coût d’association, mais il en diffère tout de même quelque peu. En effet, ce coût est supposé refléter la pertinence d’une association. Or, les différentes détections d’une même cible peuvent provenir de capteurs différents. Il n’y a donc rien d’étonnant à ce que les BBAs qui en découlent aient des formes différentes.

Masse de l’ensemble vide

Plus que par leur similitude géométrique, la pertinence de combiner deux BBAs serait mesurée par l’absence d’éléments de contradiction. Selon cette logique, la masse qui se retrouverait sur l’ensemble vide en cas d’association constitue peut-être un choix plus adéquat. Dans ce cas, les coûts élémentaires d’association valent :

\[ c_E(p_i, d_j) = \sum_{A \cap B = \emptyset} m_i(A)m_j(B) \]

\[ = m_i \otimes m_j(\emptyset) \]  

(4.12)

Cette idée a été proposée, au moins dans un cas simple, par Ayoun [3].

Bien que le conflit semble être une mesure intéressante, il souffre de quelques inconvénients. Tout d’abord, lorsque les BBAs sont très imprécises, cette valeur peut être nulle pour plusieurs associations différentes. Par conséquent, le minimum
est indéterminé et le choix de l’association sera arbitraire. D’autre part, le risque de conflit entre deux BBAs dépend de l’engagement. Par exemple, l’absence de conflit avec une BBA vide est totalement prévisible et n’apporte aucune information alors que cette même absence de conflit entre deux BBAs catégoriques sur des hypothèses singletons, a très peu de chance d’être due au hasard.

Il faut enfin noter que si un faible conflit local peut justifier le choix d’associer une piste avec une détection, la minimisation de la somme des coûts élémentaires est en revanche plus arbitraire.

### 4.2.3 Plausibilité globale

Dans un contexte un peu différent, Ristic a proposé [61] une méthode d’association reposant sur un raisonnement crédibiliste à partir de BBAs modélisant les classes des objets à associer. Nous transposons son raisonnement à notre application.

#### Association d’une seule piste

La démarche repose sur la comparaison des BBAs projetées sur un cadre commun. Ainsi, si l’on considère une piste \( p_i \) et une détection \( d_j \) modélisées respectivement par les BBAs \( m^\Omega\{p_i\} \) et \( m^\Omega\{d_j\} \), la plausibilité d’association est évaluée sur l’espace produit \( \Omega^2 = \Omega \times \Omega \), en combinant de manière conjonctive les extensions vides de \( m^\Omega\{p_i\} \) et \( m^\Omega\{d_j\} \) comme suit :

\[
m^\Omega^2\{p_i, d_j\} = m^\Omega\{p_i\}\cap\Omega^2 \odot m^\Omega\{d_j\}\cap\Omega^2
\]

Par conséquent :

\[
m^\Omega^2\{p_i, d_j\}(C) = \begin{cases} m^\Omega\{p_i\}(A)m^\Omega\{d_j\}(B) & \text{ssi } C = (A, B), \ A \subseteq \Omega, \ B \subseteq \Omega \\ 0 & \text{sinon} \end{cases}
\]

Cette BBA représente la croyance jointe relative aux positions respectives de la piste et de la détection.

À défaut d’autres informations, nous considérerons alors qu’une piste et une détection représentent la même cible (\( p_i \equiv d_j \)) si elles se trouvent à la même position. Par conséquent, la plausibilité relative à l’association de \( p_i \) et \( d_j \) est donnée par la somme des produits \( m^\Omega\{p_i\}(A)m^\Omega\{d_j\}(B) \) pour lesquelles \( A \cap B \neq \emptyset \). Plus formellement :

\[
pl^\Omega^2\{p_i, d_j\} (p_i \equiv d_j) = \sum_{A \cap B \neq \emptyset} m^\Omega\{p_i\}(A)m^\Omega\{d_j\}(B)
\]

\[
= 1 - \sum_{A \cap B = \emptyset} m^\Omega_i(A)m^\Omega_j(B) \tag{4.15}
\]

où les notations \( m^\Omega\{p_i\} \) et \( m^\Omega\{d_j\} \) ont été abrégées en \( m^\Omega_i \) et \( m^\Omega_j \). De même, puisque \( bel(A) = 1 - pl(\bar{A}) \) :

\[
bel^\Omega^2\{p_i, d_j\} (p_i \neq d_j) = \sum_{A \cap B = \emptyset} m^\Omega_i(A)m^\Omega_j(B) \tag{4.16}
\]
Or, le deuxième membre n’est autre que le conflit après combinaison conjonctive :
\[ m_{\Omega}^{\Omega}(\emptyset) = \sum_{A \cap B = \emptyset} m_{\Omega}^{\Omega}(A)m_{\Omega}^{\Omega}(B) \]  (4.17)

Ce développement montre donc que la fonction de coût proposée dans la section 4.2.2 revient à maximiser la somme des plausibilités de chaque association.

**Associtation multi pistes**

La généralisation de cette approche à l’association de \( n \) pistes \( p_i \in \mathcal{P} = \{p_1, ..., p_n\} \) avec \( n \) détections \( d_j \in \mathcal{D} = \{d_1, ..., d_n\} \) suit la même démarche. Cette fois, les BBAs \( m_{\Omega}^{\Omega} \) et \( m_{\Omega}^{\Omega} \) sont projetées par extension vide sur l’espace \( \Omega^{2n} \). La plausibilité d’une association est ensuite évaluée sur cet espace croisé [61].

**Theorème 4.2.** Soit un ensemble de \( n \) pistes \( \mathcal{P} = \{p_1, ..., p_n\} \) et \( n \) détections \( \mathcal{D} = \{d_1, ..., d_n\} \).
Si les croyances des localisations des \( p_i \in \mathcal{P} \) et \( d_j \in \mathcal{D} \) sont quantifiées respectivement par des fonctions \( m_{\Omega}^{\Omega} \) et \( m_{\Omega}^{\Omega} \) et si le vecteur \( V = [v_1, ..., v_n] \) des permutations d’indices représente l’association entre les pistes et les détections : \( p_i \equiv d_{v_i} \) pour \( i = 1, ..., n \) alors, la plausibilité de l’association \( V \) vaut :
\[ p^{\Omega^{2n}}(p_i \equiv d_{v_i} : i = 1, ..., n) = \prod_{i=1}^{n} (1 - m_{\Omega}^{\Omega}(\emptyset)) \]  (4.18)

Il apparaît donc que si Ristic cautionne [61] le choix du conflit comme critère d’association, c’est en revanche par un produit (et non par une somme) qu’il propose de combiner les différents coûts.

**Recherche de l’association la plus plausible**

Ce critère doit encore être modifié afin d’obtenir un coût additif, nécessaire pour l’exploitation d’un algorithme d’optimisation. C’est donc naturellement l’opposé du logarithme de la plausibilité que Ristic propose de minimiser. Un coût élémentaire est alors attribué à chaque association entre une piste \( p_i \) et une détection \( d_j \) :
\[ c_R(p_i, d_j) = -\log \left(1 - m_{\Omega}^{\Omega}(\emptyset)\right) \]  (4.19)

Enfin, si le nombre de pistes \( m \) est différent du nombre \( n \) de détections, l’article suggère de compléter l’ensemble le plus petit avec des BBAs vides. Les pistes ou les détections mises en correspondance avec ces BBAs dans l’association optimale resteront alors isolées.

**Discussion à propos du choix de la plausibilité**

La transposition de l’approche proposée par Ristic à notre application est particulièrement séduisante car elle propose un critère d’optimisation global justifié par un raisonnement crédibiliste. Il ne faut toutefois pas perdre de vue que la recherche de
l’association la plus plausible reste un objectif arbitraire. Ce critère est bien entendu raisonnables, mais il n’est pas nécessairement le seul pertinent.

La plausibilité est une fonction de nature optimiste. Tous les morceaux de croyance non conflictuels soutiennent l’hypothèse d’une association sans tenir compte du degré d’engagement. Bien que cette approche respecte le principe du nombre minimum de pistes suffisant pour expliquer les détection, cet optimisme provoque également certains effets indésirables déjà évoqués dans la section 4.2.2. Il s’agit de l’absence totale de conflit lors de l’association de BBAs trop peu engagées et l’absence de considération du conflit a priori attendu.

Comparaison avec le critère d’association de Mercier

L’approche proposée par Ristic se distingue de la méthode de Mercier exposée au paragraphe 4.2.1 avant tout par la nature des BBAs exploitées pour définir le critère d’association. Ristic définit son critère à partir des \( m + n \) BBAs de localisation (ou de classification) des \( m \) pistes et des \( n \) détections alors que Mercier travaille à partir de \( m \times n \) BBAs définies pour chaque association possible entre une piste et une détection. Ces hypothèses conduisent à des développements différents mais si l’on définit sur le cadre de discernement \( \Omega_{i,j} = \{y_{i,j}, n_{i,j}\} \) introduit par Mercier, la BBA d’association spécifique suivante :

\[
m(y_{i,j}) = 0 = \sum_{A \cap B = \emptyset} m_i(A) m_j(B) \tag{4.20}
m(\Omega_{i,j}) = 1 - \sum_{A \cap B = \emptyset} m_i(A) m_j(B) \tag{4.21}
\]

Où les \( m_i(A) \) et \( m_j(B) \) sont les masses de localisation des pistes et des détections, alors le critère de Ristic revient à maximiser la plausibilité sur un des cadres \( m^{\Omega_{i,j} \setminus \Omega_{pi}} \) ou \( m^{\Omega_{i,j} \setminus \Omega_{dj}} \) défini par Mercier.

4.2.4 Prise en compte de l’engagement

Comme cela a déjà été mentionné, le principal inconvénient de l’utilisation du conflit comme critère d’association est qu’il ne tient pas compte de l’engagement des BBAs. Par exemple, le coût d’association avec une BBA vide sera toujours nul. Cette approche est raisonnable car plus une piste est imprécise, plus son association avec une détection éloignée est plausible. Cependant, lorsque deux associations de pistes avec une seule détection disponible sont plausibles, il paraît plus naturel de privilégier la piste la plus engagée. En effet, plus les BBAs sont engagées plus il y a de chance pour que l’absence de conflit ne soit pas due au hasard.

Estimation du conflit attendu

Le conflit attendu entre deux BBAs peut être estimé en fonction du cardinal des éléments focaux. Cette valeur peut servir de référence pour déterminer si le conflit
Suivi multi pistes

observé lors d’une association est étonnant ou non.

Si $A_i$ et $A_j$ deux sous-ensembles de cardinaux respectifs $i$ et $j$ tirés aléatoirement sur un ensemble $A_n$ de cardinal $n$, la probabilité que $A_i \cap A_j = A_m$ est donnée par :

$$ P \left( A_i \cap A_j = A_m \right) = \frac{C^{i-m}_{n-m}C^{j-m}_{n-i}}{C^n_iC^n_j} (n - m)!((n - i)!(n - j)!i!j!)$$

$$= \frac{(n - m)!(n - i)!((n - j)!i!j!)}{(i - m)!(j - m)!(n - i - j + m)!(n!)^2}$$

(4.22)

à condition que $n + m \geq i + j, n \geq i \geq m$ et $n \geq j \geq m$.

En particulier, pour $A_m = \emptyset$

$$ P \left( A_i \cap A_j = \emptyset \right) = \frac{C^n_j}{C^n_i} \frac{(n - i)!((n - j)!)}{(n - i - j)!n!}$$

(4.23)

La probabilité que deux ensembles tirés aléatoirement soient d’intersection vide sera donc d’autant plus petite que les ensembles sont grands.

A partir de ce résultat préalable, nous pouvons étudier la variable aléatoire correspondant au conflit résultant de la combinaison conjonctive de deux BBAs $\tilde{m}_1$ et $\tilde{m}_2$ dont les masses et les cardinaux des éléments focaux $m^i_1, i \in \{1, \ldots, k\}$ et $m^j_2, j \in \{1, \ldots, l\}$ sont fixés mais dont les hypothèses constituant chaque élément focal sont tirées aléatoirement :

$$\tilde{m}_{12}(\emptyset) = \sum_{i=1}^{k} \sum_{j=1}^{l} m^i_1 m^j_2 P \left( A_i \cap A_j = \emptyset \right)$$

(4.24)

L’espérance $\mu$ et la variance $\sigma^2$ sont alors données par :

$$\mu \left( \tilde{m}_{12}(\emptyset) \right) = \sum_{i=1}^{k} \sum_{j=1}^{l} m^i_1 m^j_2 P \left( A_i \cap A_j = \emptyset \right)$$

$$= \sum_{i=1}^{k} \sum_{j=1}^{l} m^i_1 m^j_2 \frac{C^{i-m}_{n-m}C^{j-m}_{n-i}}{C^n_iC^n_j} \left(1 - \frac{C^n_{i-m}}{C^n_i} \right)$$

(4.25)

$$\sigma^2 \left( \tilde{m}_{12}(\emptyset) \right) = \sum_{i=1}^{k} \sum_{j=1}^{l} m^i_1 m^j_2 P \left( A_i \cap A_j = \emptyset \right) \left(1 - P \left( A_i \cap A_j = \emptyset \right) \right)$$

$$= \sum_{i=1}^{k} \sum_{j=1}^{l} m^i_1 m^j_2 \frac{C^{i-m}_{n-m}C^{j-m}_{n-i}}{C^n_iC^n_j} \left(1 - \frac{C^n_{i-m}}{C^n_i} \right)$$

(4.26)
L’évolution des probabilités de $P(A_i \cap A_j = \emptyset)$ illustrée sur la figure 4.2 indique alors que l’espérance du conflit va diminuer lorsque les cardinaux des éléments focaux grandissent. Bien que représentant des objets différents, certaines BBAs risquent donc de ne pas être en conflit si elles sont trop peu engagées.

Ce résultat théorique met en évidence les limites du conflit comme critère d’association mais son utilisation directe n’a pas été concluante. Les difficultés rencontrées s’expliquent avant tout par le non respect de l’hypothèse selon laquelle les BBAs correspondent à des tirages uniformes. Cette hypothèse est d’ailleurs en contradiction avec la prise en compte d’a priori topographique détaillée dans le chapitre 3. Nous avons donc cherché un critère crédibiliste qui, à défaut de tenir compte explicitement de la part aléatoire du conflit, varie en fonction de cardinal des BBAs.

**Probabilité pignistique d’une association**

En reprenant l’expression de la croyance jointe relative aux positions d’une piste $p_i$ et d’une détection $d_j$ :

$$m_{Ω^2}\{p_i, d_j\}(C) = \begin{cases} m_i^Ω(A)m_j^Ω(B) & \text{ssi } C = (A, B), \ A \subseteq Ω, \ B \subseteq Ω \\ 0 & \text{sinon} \end{cases} \quad (4.27)$$

nous pouvons calculer la probabilité pignistique que la piste et la détection se trouvent toutes les deux à une même position $ω_k$ :

$$\text{Bet} P_{Ω^2}\{p_i, d_j\}(ω_k, ω_k) = \sum_{C \cap (ω_k, ω_k) \neq \emptyset} m_{Ω^2}\{p_i, d_j\}(C) \frac{1}{|C|}$$

$$= \sum_{A \ni ω_k} \sum_{B \ni ω_k} m_i^Ω(A)m_j^Ω(B) \frac{1}{|A||B|} \quad (4.28)$$

En faisant la somme sur toutes les localisations $ω_k \in Ω$, nous obtenons alors la probabilité pignistique de $p_i \equiv d_j$ :

![Fig. 4.2: Évolution $P(A_i \cap A_j = \emptyset)$ en fonction de $|A_i| + |A_j|$ pour $n = 50$](image)
Suivi multi pistes

\[ \text{Bet}P^{\Omega^2}\{p_i, d_j\} (p_i \equiv d_j) = \sum_{\omega_k \in \Omega} \text{Bet}P^{\Omega^2}\{p_i, d_j\}(\omega_k, \omega_k) \]

\[ = \sum_{A \subseteq \Omega} \sum_{B \subseteq \Omega} m_i^\Omega(A)m_j^\Omega(B) \left| \frac{A \cap B}{|A||B|} \right| \]  

\[(4.29)\]

Comparativement au critère s’appuyant sur le conflit, chacun des produits est donc pondéré par un facteur qui tient compte de la taille des éléments focaux. Les éléments les plus engagés auront ainsi plus de poids.

Enfin, la généralisation à \( n \) associations s’obtient par le produit :

\[ \text{Bet}P^{\Omega^{2n}} (p_i \equiv d_{v_i} : i = 1, \ldots, n) = \prod_{i=1}^{n} \left( \text{Bet}P^{\Omega^2}(p_i \equiv d_{v_i}) \right) \]

\[ = \prod_{i=1}^{n} \left( \sum_{A \subseteq \Omega} \sum_{B \subseteq \Omega} m_i^\Omega(A)m_j^\Omega(B) \left| \frac{A \cap B}{|A||B|} \right| \right) \]

\[(4.30)\]

dont le maximum peut être trouvé en minimisant la somme des coûts locaux de la forme :

\[ c_{\text{Bet}P}(p_i, d_{v_i}) = -\log \left( \text{Bet}P^{\Omega^2}(p_i \equiv d_{v_i}) \right) \]

\[(4.31)\]

La probabilité pignistique constitue une alternative intéressante à la plausibilité et permet de favoriser les associations entre BBAs les plus engagées.

### 4.3 Résultats

En vue de déterminer la fonction de coût la plus pertinente, les différentes méthodes d’association ont été comparées par simulations. L’objectif est de choisir parmi les différentes méthodes proposées celle qui est la mieux adaptée à notre cas particulier. Cette étude a donc été influencée par les spécificités de l’application et les conclusions ne peuvent par conséquent pas être extrapolées à tous les problèmes d’association.

Description du dispositif expérimental

Afin de réduire le nombre de paramètres et de s’affranchir des perturbations dues au pistage, les essais ont été réalisés en simulant deux capteurs avec une importante zone de recouvrement. Chaque méthode a alors été évaluée en comparant l’association des détections provenant des deux capteurs.

A chaque pas de temps, un nombre fixe de positions de cibles est tiré de manière uniforme sur une région rectangulaire englobant les zones de couvertures des deux capteurs. Ces positions, sans corrélation temporelle, constituent la vérité terrain. Les détections provenant des deux capteurs sont ensuite simulées en ajoutant un bruit.
gaussien à l’estimation de l’angle et de la distance des cibles. De plus, les probabilités de non détection et de fausse détection ont été supposées nulles. Toutefois, certaines cibles pouvant se trouver en dehors des zones de couverture d’un ou des deux capteurs, certaines détections peuvent être non associées.

Fig. 4.3: Simulations de détections pour le pistage. Capteurs estimant la distance ou non.

Les BBAs formées à partir des détections bruitées (figure 4.4) sont introduites dans les différentes méthodes d’association et les résultats sont comparés à la vérité terrain.

Fig. 4.4: Construction de BBAs. Capteurs estimant la distance ou non

Tous les tests relatifs à l’association ont été réalisés en simulant des capteurs d’une portée de 120m et une ouverture angulaire de 60°. Ces deux capteurs ont été placés aux deux coins opposés d’un carré de 100m de côté et orientés l’un vers l’autre, comme illustré sur la figure 4.3. Chacun des essais a alors été réalisé en comparant l’association avec la distance de Jousselme, la plausibilité \(-p(l(p_i \equiv d_j) = m_{ij}(\emptyset) - 1\), le critère de Ristic \(-\log(\text{pl}(p_i \equiv d_j)\), \(-\text{BetP}(p_i \equiv d_j))\) et \(-\log(\text{BetP}(p_i \equiv d_j)).\)

De manière générale, le conflit et la probabilité pignistique ont toujours donné des résultats très proches de leurs versions logarithmiques (moins d’1% de différence). Par conséquent, et par souci de clarté, seule la version logarithmique a été représentée sur les graphes.
**BBAs peu engagées**

Les premières simulations ont été réalisées en construisant des BBAs avec une expansion de 25 m et 25° pour chacun des trois éléments focaux consonants. Il s’agit donc, au regard de la portée et de la position des capteurs, de BBAs très peu engagées. De telles BBAs peuvent refléter une mauvaise connaissance de la position des capteurs ou un certain pessimisme quant à la précision des détections. Ce pessimisme peut être justifié par la connaissance de propriétés statistiques ou simplement être le reflet de l’ignorance de la précision du capteur.

![Fig. 4.5: Taux de bonnes associations en fonction du nombre de cibles simultanées : cas de BBAs peu engagées](image)

La figure 4.5 présente l’évolution du taux de bonnes associations en fonction du nombre de cibles simultanées. Les trois mesures ont été testées avec des erreurs de localisation d’écarts-types : \( \sigma \in \{0, 5, 7.5\} \). Dans ce cas, la distance de Jousselme donne toujours le meilleur résultat alors que la plausibilité est toujours plus mauvaise. Cette tendance s’explique par le faible engagement des BBAs qui pénalise davantage la plausibilité. Celle-ci, déjà peu engagée par nature, va attribuer des coûts équivalents à plusieurs associations concurrentes et la solution de coût minimal devient indéterminée. Nous pouvons toutefois remarquer que contrairement aux autres coûts étudiés, le taux de réussite de la plausibilité ne diminue pas lorsque les erreurs de localisation augmentent.

**BBAs engagées**

Lorsque les BBAs sont plus engagées (5 m et 5° d’expansion par élément focal), la tendance s’inverse. Nous pouvons observer sur la figure 4.6 que le critère s’appuyant sur la plausibilité devient plus efficace que la distance de Jousselme. De plus, nous obtenons de meilleurs résultats que l’expérience précédente pour tous les coûts d’associations. Il ne faut toutefois pas perdre de vue que si deux BBAs sont très engagées, elles risquent d’être totalement disjointes et leur association devient donc exclue. Par conséquent, une non association étant généralement plus dommageable qu’une association erronée, il est préférable de rester prudent dans l’engagement des BBAs.
Taille variable de \textit{BBAs}

Pour cette troisième série de simulations, chaque \textit{BBA} a été construite en tirant aléatoirement l’option d’une \textit{BBA} engagée ou pas, avec les propriétés respectives décrites dans les deux sections précédentes. Ce cas de figure n’est pas représentatif du comportement d’un capteur physique, mais il permet de simuler l’association de détections avec des pistes ayant des degrés d’engagements différents. L’objectif de ces simulations est de tester les critères d’association lorsqu’ils sont confrontés à des \textit{BBAs} de tailles et de formes différentes.

La figure 4.7 montre que le taux de bonne association décroit rapidement avec le nombre de pistes simultanées et ce, pour tous les coûts testés. C’est toutefois le coût construit à partir du $BetP(p_i \equiv d_j)$ qui est presque toujours le meilleur. Le coût formé à partir de la plausibilité est lui presque toujours le plus mauvais.
Synthèse de l’étude

Aucun des coûts testés lors des essais comparatifs ne s’est montré systémati-ment meilleur que les autres. Les performances relatives varient en effet selon l’en-gagement des BBAs et la précision des capteurs. Il ressort toutefois que $-\log(BetP(p_i \equiv d_j))$ est toujours proche sinon le meilleur des critères. Comme il bénéficie en outre d’une justification théorique solide, c’est cette expression de coût qui a été choisie.

4.4 Conclusion

Au cours de ce chapitre, nous avons passé en revue plusieurs méthodes pouvant être exploitées pour l’association des pistes et des détections. Cette étude nous a amené à constater que les BBAs modélisant les localisations des pistes et des détections pouvaient être directement exploitées pour résoudre le problème en suivant un raisonnement crédibiliste cohérent. Cette démarche, initialement obtenue en trans-posant le travail de Ristic [61] à notre problème, a été adaptée pour tenir compte de l’engagement des BBAs.

Après avoir comparé les différentes solutions par simulation, nous proposons enfin de procéder comme suit. Les associations sont contraintes par des relations de type 1-1 auxquelles sont associés des coûts de la forme $-\log(BetP(p_i \equiv d_j))$ et dont l’optimum est trouvé par la méthode hongroise [54] [38].
Chapitre 5

Simplification des fonctions de masse

Sommaire

5.1 Introduction ............................................. 86
5.2 Méthodes de simplification ............................... 86
  5.2.1 Summerization ..................................... 87
  5.2.2 Agrégation itérative ................................. 87
  5.2.3 Minimisation d’une distance ......................... 88
  5.2.4 Simplification de SSFs ............................... 89
5.3 Expérimentations ......................................... 91
  5.3.1 Taux de bonne classification après combinaison ... 91
  5.3.2 Distance de Jousselme .............................. 93
5.4 Conclusion ............................................... 95
5.1 Introduction

Malgré l’intérêt grandissant accordé aux fonctions de croyance, certaines difficultés s’opposent encore à la mise en œuvre de cette théorie. Parmi les arguments régulièrement avancés contre la TBM, il y a la taille des fonctions manipulées. Celle-ci peut en effet rendre les temps de calcul considérablement plus longs que dans le cas de méthodes plus classiques comme la théorie bayésienne ou la théorie des possibilités.

Ce problème devient critique lorsque la quantité de combinaisons successives et la taille de l’espace de discernement sont importantes. Dans ce cas, le nombre d’éléments focaux peut croître de manière exponentielle et non seulement les calculs deviennent rapidement impossibles à réaliser dans des temps raisonnables, mais aussi la distribution de la masse sur un grand nombre d’éléments focaux rend difficile toute interprétation des valeurs numériques.

Cette situation critique survient notamment lors du pistage crédibiliste présenté dans ce travail puisque plusieurs dizaines de BBAs de localisation définies sur un grand espace de discernement (environ 1000 hypothèses) sont successivement combinées. Il est donc nécessaire de simplifier régulièrement les BBAs pour assurer la stabilité en temps de calcul du système. De même, la capacité limitée du réseau de communication impose aussi une simplification drastique des BBAs lors de l’envoi d’une piste.

Au cours de ce chapitre, nous allons commencer par détailler les principales approches de la littérature. Ensuite, une méthode originale s’appuyant sur la décomposition canonique de la BBA sera également présentée. Enfin, les différentes méthodes seront comparées par quelques simulations.

5.2 Méthodes de simplification

L’objectif de la simplification d’une BBA est de réduire le nombre d’éléments focaux tout en préservant les informations essentielles présentes dans la BBA initiale. Contrairement à la prise de décision, ce traitement n’est pas réalisé au niveau pignistique, mais au niveau crédal. C’est donc toute la structure de la fonction de croyance qu’il faut essayer de préserver et pas uniquement la probabilité pignistique des singletons.

Plusieurs auteurs ont proposé des méthodes de simplification des BBAs. Une première démarche consiste à remplacer la BBA par une fonction ayant des caractéristiques prédéfinies. Parmi ces approches, il est notamment proposé de remplacer la BBA par une approximation bayésienne [74] ou possibiliste [20].

Une deuxième stratégie consiste à regrouper ou supprimer certains éléments focaux sans contraindre le résultat par une forme prédéfinie. Ces méthodes, présentant l’intérêt d’être rapides et simples à mettre en œuvre, se distinguent alors par le critère de sélection et le mode de regroupement. Parmi celles-ci, quelques méthodes choisies en raison de leur popularité vont être brièvement détaillées.
5.2.1 **Summerization**

Une des techniques les plus simples pour réduire le nombre d’éléments focaux d’une BBA consiste à remplacer les $n$ éléments focaux de masses les plus faibles par un seul élément correspondant à leur union. Cette méthode [47] est connue sous le nom de *summarization*. Bien que très simple, cette approche permet la suppression d’un nombre arbitraire d’éléments focaux en une seule passe. Elle respecte en outre le principe de prudence car l’approximation est moins engagée que la BBA originale.

Cette méthode est suffisante pour la suppression d’un petit nombre de masses résiduelles, mais une simplification plus radicale donne généralement un résultat médiocre. La masse de l’élément formé par l’union des éléments supprimés peut en effet devenir non négligeable par rapport aux autres éléments focaux. La plausibilité de chacune des hypothèses inclues s’en trouve alors fortement augmentée. Notons également que deux simplifications successives ne donneront pas le même résultat qu’une simplification effectuée en une seule fois. La méthode n’est donc pas associative.

5.2.2 **Agrégation itérative**

L’agrégation consiste à regrouper de manière itérative les éléments focaux en clusters. Initialement, le critère proposé était la minimisation de la perte de crédibilité [35]. Dans ce cas, les deux éléments focaux sélectionnés et regroupés à chaque itération sont ceux qui minimisent la fonction définie par:

$$D_{Bel}(A, B) = 2^{|\Omega - A|} m(A) + 2^{|\Omega - B|} m(B) - 2^{|\Omega - A \cup B|} (m(A) + m(B))$$  \hspace{1cm} (5.1)

La minimisation de la perte de crédibilité totale n’est pas garantie sur plus d’une itération, mais cette technique heuristique permet d’approcher le critère dans des temps de calcul réalisables.

D’autres critères sont envisageables. Par exemple [12] propose d’utiliser la mesure de précision d’une BBA [20] définie par:

$$|m^\Omega| \triangleq \sum_{A \subseteq \Omega} |A|m(A)$$  \hspace{1cm} (5.2)

C’est alors l’accroissement de plausibilité des singletons qui est minimisé. En effet, la plausibilité d’un singleton est égale à la somme des masses des éléments focaux auxquel il appartient. Un élément focal $m(A)$ contribue donc à la plausibilité de $|A|$ singletons. Par conséquent :

$$\sum_{\omega \in \Omega} pl(\omega) = \sum_{A \subseteq \Omega} |A|m(A)$$  \hspace{1cm} (5.3)

Les deux éléments focaux $A$ et $B$ regroupés selon la mesure de précision seront ceux qui minimisent :

$$D_P(A, B) = m(A) (|A \cup B| - |A|) + m(B) (|A \cup B| - |B|)$$  \hspace{1cm} (5.4)
$D_P(A, B)$ a donc la même forme que $D_{Bel}(A, B)$ où les $2^{|\Omega - X|}$ sont remplacés par $|X|$, pour $X \in \{A, B, A \cup B\}$.

Notons aussi que si $A \not\subseteq B$, les critères deviennent :

$$D_{Bel}(A, B) = m(A) \left(2^{|\Omega - B|} - 2^{|\Omega - A|}\right)$$  \hfill (5.5)

$$D_P(A, B) = m(A) \left(|B| - |A|\right)$$  \hfill (5.6)

Ces critères sont indépendants de la masse de $B$ et s’il existe plusieurs éléments focaux $B_i \supseteq A$ de même cardinal, les coûts $D_P(A, B_i)$ seront équivalents. Il y a donc parfois plusieurs *minima* équivalents par rapport à ces critères.

Enfin, pour notre application, le critère $D_P$ a été préféré à $D_{Bel}$ car ce dernier dépend de la taille du cadre de discernement $\Omega$. Cette dépendance est non seulement gênante car $|\Omega|$ dépend du conditionnement, mais elle rend même souvent impossible le calcul de $D_{Bel}$ en raison des grandeurs qui doivent être calculées.

### 5.2.3 Minimisation d’une distance

Les mesures de similarités introduites au chapitre 2.6.1 permettent de quantifier la ressemblance entre deux $BBA$s. Il semble donc naturel d’utiliser ces distances comme critère de minimisation lors de la simplification d’une $BBA$. Cependant, la minimisation globale selon ce critère est difficile et l’unicité n’est bien souvent pas garantie. Aussi, ces distances sont préférentiellement utilisées dans la méthode d’agrégation détaillée dans le paragraphe précédent. Remarquons d’ailleurs que le critère $D_{Bel}$ est la distance $L_1$ sur les crédibilités.

En remplaçant $D_{Bel}$ par le carré de la distance de Jousselme $D_J$ [41] souvent utilisée pour caractériser la similitude entre deux $BBA$s, nous obtenons alors un nouveau critère :

$$D_J(A, B) = \frac{1}{2} m(A)^2 + \frac{1}{2} m(B)^2 + \frac{1}{2} (m(A) + m(B))^2$$

$$+ \ m(A) (m(A) + m(B)) \frac{|A|}{|A \cup B|}$$

$$- \ m(B) (m(A) + m(B)) \frac{|B|}{|A \cup B|}$$

$$+ \ m(A)m(B) \frac{|A \cap B|}{|A \cup B|}$$

qui donne après simplification :

$$D_J(A, B) = m(A)^2 \left(1 - \frac{|A|}{|A \cup B|}\right)$$

$$+ \ m(B)^2 \left(1 - \frac{|B|}{|A \cup B|}\right)$$
5.2.4 Simplification de SSFs

Plutôt que de travailler directement sur les masses, il est également possible de passer par la décomposition canonique de la BBA. Étant donné que les (I)SSFs factorisent la BBA en un ensemble de croyances élémentaires favorables (ou défavorables) à différentes hypothèses (cf chapitre 2), il suffit alors de sélectionner les (I)SSFs les plus significatives et de reconstruire la BBA.

L’intérêt de cette approche réside avant tout dans la redistribution de la croyance des (I)SSFs supprimées. Plutôt que d’ajouter un élément focal sur l’union des deux éléments supprimés, la reconstruction de la BBA à partir des (I)SSFs va répartir la croyance d’un élément manquant sur tous les éléments focaux qui l’incluaient. La BBA résultante sera donc moins engagée et la perte d’informations sera plus importante. En revanche, cette approche redistribue la croyance sur un plus grand nombre d’éléments focaux et évite ainsi de concentrer les résidus de croyance sur un petit nombre d’hypothèses.

BBAs consonantes

Si une BBA est consonante, les poids $w$ des SSFs de sa décomposition sont compris entre 0 et 1. De plus, le soutien apporté par une SSF est inversément proportionnel à $w$. Par conséquent, une première méthode de simplification consiste simplement à supprimer les SSFs ayant les $w$ les plus grands.

Nous pouvons cependant sélectionner plus finement les éléments à supprimer. En effet, la masse de chaque élément focal dépend de l’ensemble des SSFs, selon le produit :

$$m(A) = (1 - w(A)) \prod_{A_i \subseteq A} w(A_i)$$ (5.7)

La masse d’un élément focal ne dépend donc pas que du $w$ défini sur le même ensemble, mais elle dépend aussi des $w$ de tous les éléments focaux de cardinal plus petit (dans le cas consonant). Un critère s’appuyant sur un seul $w$, sans tenir compte de la dépendance avec les autres valeurs, est donc peut être trop simple.

Suite à ce constat, nous proposons un deuxième critère de sélection des SSFs :

$$D_w(A) = (1 - w(A)) \prod_{A_i \subseteq A} w(A_i)$$ (5.8)

Chaque fois qu’une SSF $A^w$ est supprimée, les valeurs des $D_w(A_i)$ sont alors mises à jour en les divisant par $w$ si $A \subseteq A_i$. Cette mise à jour peut changer l’ordre d’importance des éléments et les éléments supprimés ne seront donc pas nécessairement les mêmes que par une summerization.

Tout comme pour l’agrégation des masses, la distance de Jousselme $D_{Jw}(A_i)$ constitue également un critère intéressant de sélection des SSFs :
\[ D_{Jw}(A) = \left( 1 - \frac{1}{w(A)} \right)^2 \sum_{A_i \supset A, A_j \supset A_i} \frac{|A_j|}{|A_i|} m(A_i)m(A_j) \]

\[ + \left( 1 - \frac{1}{w(A)} \right) \sum_{A_i \supset A} \frac{|A_i|}{|A|} m(A_i)m(A) \]

\[ + \frac{1}{2} m(A)^2 + \frac{1}{2} \left( 1 - \frac{1}{w(A)} \right)^2 \sum_{A_i \supset A} m(A_i)^2 \]

où la relation

\[ m(A) = \left( \frac{1}{w(A)} - 1 \right) \sum_{A_i \supset A} m(A_i) \]

(5.9)

permet encore d'écrire le dernier membre sous une autre forme :

\[ D_{Jw}(A) = \left( 1 - \frac{1}{w(A)} \right)^2 \sum_{A_i \supset A, A_j \supset A_i} \frac{|A_j|}{|A_i|} m(A_i)m(A_j) \]

\[ + \left( 1 - \frac{1}{w(A)} \right) \sum_{A_i \supset A} \frac{|A_i|}{|A|} m(A_i)m(A) \]

\[ + m(A)^2 - \left( 1 - \frac{1}{w(A)} \right)^2 \sum_{A_i, A_j \supset A, A_i \neq A_j} m(A_i)m(A_j) \]

La relation 5.8 permet de formuler ce critère à partir des poids de la décomposition canonique.

**BBAs séparables**

Si une BBA est séparable mais pas consonante, la croyance d'une SSF supprimée sera, comme dans le cas consonant, répartie sur les éléments focaux dans lesquels elle est incluse. En revanche, la relation 5.8 n’est plus vérifiée. Pour être rigoureux, il faut donc calculer les masses par combinaison des SSFs restantes après chaque suppression d’un élément. Cependant, la relation 5.8 peut être utilisée comme approximation afin de limiter les calculs.

Notons que ces approches sont particulièrement intéressantes lorsque la simplification est exploitée dans un système général qui manipule des SSFs car dans ce cas la décomposition canonique, qui constitue une part importante du calcul est déjà disponible.
BBAs non séparables

Lorsque les BBAs ne sont pas séparables, il est impératif de commencer par supprimer les ISSFs. Cette contrainte, nécessaire pour que toutes les masses obtenues après reconstruction de la BBA soient comprises entre 0 et 1, empêche l’application des critères $D_w$ et $D_{Jw}$ proposés dans le cas séparable.

Le premier critère, consistant à supprimer les (I)SSFs ayant les poids les plus grands reste, lui, applicable. Cependant, plus une ISSF a un poids élevé plus elle est significativement défavorable à l’hypothèse. Il est donc préférable de commencer par supprimer les ISSFs en commençant par celles qui ont un poids plus proches de 1.

Bien que mathématiquement possible, l’intérêt de cette adaptation au cas non séparable est toutefois réduit de par la nécessité de supprimer toutes les ISSFs avant les SSFs.

5.3 Expérimentations

La simplification d’une BBA suppose une fusion ultérieure avec de nouvelles informations. C’est en effet ce traitement à venir au niveau crédal qui justifie la nécessité de préserver toute la structure de croyance et de ne pas se contenter d’une décision pignistique. C’est pour cette raison que le critère caractérisant le mieux la qualité d’une simplification est probablement le taux de bonne décision après combinaison de plusieurs BBAs simplifiées. Cependant, ce critère dépend de plusieurs paramètres indépendants de la simplification tels que le nombre et la forme des BBAs ou les règles de combinaison et de décision appliquées. Un deuxième critère d’évaluation a donc été ajouté à ce premier élément de comparaison.

5.3.1 Taux de bonne classification après combinaison

Les premières simulations ont été réalisées en suivant la démarche suivante. Des observations $\omega_i$ sont tirées selon une loi de probabilité centrée sur une vérité. Les incertitudes relatives aux observations sont ensuite modélisées par des BBAs consécutives construites à partir de la même loi de probabilité recentrée autour de l’observation. Cette approche fait donc l’hypothèse que la vérité est cachée à l’observateur mais que ce dernier connaît la loi probabilité avec laquelle l’observation a été tirée.

Les BBAs obtenues sont simplifiées avec la méthode testée avant d’être combinées selon la règle conjonctive $TBM$. Enfin, le taux de bonne classification relatif à une règle de simplification est obtenu en comptant le nombre de bonnes classifications (maximum de $BetP$) sur un grand nombre d’expériences (10000).

La figure 5.1 représente le taux de bonne classification après combinaison de 5 BBAs construites selon la méthode présentée ci-dessus, avec une distribution de probabilité gaussienne discrète ($\sigma = 2$) définie sur un cadre de 25 hypothèses.

La courbe verte claire représente la simplification par summerization ; la rouge, le regroupement selon le critère de précision ; la bleue, le regroupement selon la distance...
Simplification des fonctions de masse

Fig. 5.1: *Taux de bonne classification en fonction du nombre d’éléments focaux.*

Nous pouvons observer que la simplification utilisant le critère $D_w$ et la summarization donnent les meilleurs résultats. Au contraire, la simplification à partir de la valeur des $w$ est nettement moins bonne que les autres approches. De plus, le taux de bonne classification reste constant pour toutes les méthodes (à l’exception du $w$) tant que le nombre d’éléments focaux est supérieur à 5. Cela s’explique par la présence de plusieurs éléments focaux de masse très faible sur des ensembles de cardinal proche de $|\Omega|$.

Pour ne pas limiter l’approche au cas d’une distribution gaussienne, nous avons également réalisé la même expérience en tirant une loi de probabilité différente pour chaque observation. Ces lois sont alors tirées de manière uniforme parmi les lois symétriques et décroissantes (plus un élément est proche de la vérité, plus sa probabilité est grande).

Fig. 5.2: *Taux de bonne classification en fonction du nombre d’éléments focaux.*

Sur la figure 5.2, les deux graphes représentent le taux de bonne classification après combinaison de 10 BBAs simplifiées. Le premier graphe a été réalisé en tirant chaque loi de probabilité sur un cadre de 25 hypothèses et le second sur un cadre de 15 hypothèses. Sur ces deux graphes, l’ordre des méthodes de simplification change peu : $D_w$ (en mauve) et $D_J w$ (en vert foncé) et la summarization (en vert clair) sont les meilleurs, le regroupement selon la distance de Jousselme (en bleu) se trouve au
milieu et les méthodes selon le critère \( w \) (en jaune) et le critère de précision (en rouge) sont les plus mauvais.

### 5.3.2 Distance de Jousselme

Outre le taux de bonne classification, les méthodes de simplification ont également été évaluées en comparant les distances de Jousselme séparant la \( BBA \) originale de la \( BBA \) simplifiée. Cette approche, bien que plus éloignée de notre objectif de la simplification, présente l'intérêt de faire intervenir moins de paramètres arbitraires dans la simulation.

Cette mesure a d’abord été testée sur des \( BBAs \) consonantes qui constitue le cas à partir duquel le critère de simplification des \( SSFs \) a été défini. La mesure a ensuite été appliquée à des \( BBAs \) séparables représentatives des \( BBAs \) de notre système de fusion multi capteurs. Enfin, la simplification de \( BBAs \) quelconques tirées selon la méthode proposée par Burger [10] est également présentée à titre d’information.

**Simplification de BBAs consonantes**

![Distance de Jousselme en fonction du nombre d'éléments focaux](image)

Sur le graphe 5.3, la courbe verte claire représente la simplification par *summer-ization* ; la rouge, le regroupement selon le critère de précision ; la bleue, le regroupement selon la distance de Jousselme ; la jaune, la simplification à partir des valeurs de \( w \); la mauve, à partir du critère \( D_w \) et enfin la courbe verte foncée, à partir du critère \( D_{Jw} \).

Il apparaît sur cette figure que la plupart des méthodes de simplification donnent des résultats similaires, à l’exception de la *summerization* (en vert clair) et de la simplification selon \( w \) (en jaune).

**Simplification de BBAs séparables**

Les \( BBAs \) séparables utilisées pour ces tests ont été construites en combinant deux \( BBAs \) consonantes. Il ne s’agit donc pas du cas général, mais ces tirages permettent de générer simplement des \( BBAs \) séparables et non consonantes tout en limitant la masse sur l’ensemble vide.
Fig. 5.4: Distance de Jousselme en fonction du nombre d’éléments focaux

Le graphe 5.4 confirme les résultats obtenus dans le cas consonant. A nouveau, seuls le critères $w$ et la summarization de distinguent clairement et de manière négative des autres méthodes. Rappelons toutefois que les BBAs testées sont formées par la combinaison conjonctive TBM de deux BBAs consonantes. Il ne s’agit donc pas du cas général de BBAs séparables.

Simplification de BBAs uniformes

Fig. 5.5: Distance de Jousselme en fonction du nombre d’éléments focaux

Lorsque les BBAs ne sont pas séparables, il n’est plus possible d’appliquer les critères $D_w$ et $D_{Jw}$ car les (I)SSFs après simplification ne se combinent plus nécessairement en une BBA. De plus, bien que la simplification à partir des valeurs de $w$ soit possible (courbe jaune), le résultat obtenu est nettement moins bon (du moins en terme de distance de Jousselme).

En ce qui concerne les autres méthodes de simplification testées, la summarization (courbe verte) est à nouveau légèrement moins bonne que les deux méthodes par clustering selon la distance de Jousselme (courbe bleue) et le critère de précision (courbe rouge).
5.4 Conclusion

La méthode que nous avons développée à partir du critère $D_w$, s'appuyant sur la décomposition canonique des BBAs, semble donner de bons résultats, lorsque les BBAs sont consonantes. C'est également le cas de la summerization, mais cette dernière est relativement mauvaise en terme de distance de Jousselme. C'est donc la méthode $D_w$ qui été utilisée pour simplifier les BBAs dans le système de fusion décentralisé car toutes les BBAs sont séparables. Cette condition est en effet nécessaire pour que cette approche soit exploitable. Soulignons enfin que la comparaison des méthodes de simplification de BBAs n’est pas triviale et qu’en fonction de l’objectif recherché, d’autres simulations peuvent s’avérer nécessaires.
Chapitre 6

Fusion décentralisée

Sommaire

6.1 Définition du problème ........................................ 98
   6.1.1 Éléments du déploiement ................................. 98
   6.1.2 Objectifs et difficultés .................................... 99
6.2 Principes de la méthode ...................................... 100
   6.2.1 Application locale de la fusion ......................... 100
   6.2.2 Unicité des pistes ......................................... 100
   6.2.3 Unicité des associations ................................. 100
6.3 Communications entre agents ................................. 100
   6.3.1 Attribution des détections .............................. 100
   6.3.2 Création d’une piste ..................................... 101
   6.3.3 Transfert d’une piste .................................... 102
   6.3.4 Disparition d’une piste .................................. 103
6.4 Conclusion ..................................................... 103
6.1 Définition du problème

Outre la définition d’un système de fusion multi capteurs dans le cadre des fonctions de croyance, un autre objectif de ce travail concerne le déploiement de l’application selon une architecture décentralisée.

Avant de détailler la stratégie proposée, précisons le cadre du problème ainsi que les objectifs visés.

6.1.1 Éléments du déploiement

Les architectures décentralisées présentées dans la section 1.6.1, sont caractérisées par le fait qu’aucun élément n’occupe une position centrale. L’ensemble du système est alors constitué d’un maillage de nœuds de détections dont chaque élément est capable de communiquer avec ses voisins. En plus des nœuds de détections, nous allons également supposer que notre système comporte un ou plusieurs postes de supervision chargés de collecter les informations pour l’affichage. Précisons les caractéristiques et fonctions respectives de ces deux éléments.

Nœud de détection

Un nœud de détection est constitué d’un ou plusieurs capteurs, d’un moyen de communication et de ressources de calcul. Des agents, implantés sur chaque nœud, sont alors chargés d’effectuer la fusion des données en collaboration avec leurs voisins et à partir d’une connaissance partielle du déploiement. Pour mener à bien cette tâche, nous supposerons également que chaque agent dispose des informations qui suivent.

- Caractéristiques des capteurs locaux : chaque agent connaît la position, l’orientation et les caractéristiques physiques des capteurs présents sur son nœud. Dans le cadre des fonctions de croyance, cela signifie qu’il dispose des informations nécessaires pour construire les BBAs à partir des détections locales.
- Configuration du voisinage : deux nœuds sont considérés comme voisins s’ils sont susceptibles de détecter simultanément la même cible. Cette notion de voisinage dépend donc non seulement de la proximité spatiale et de la topographie du terrain mais aussi de l’orientation et de la portée des capteurs. En plus des caractéristiques locales, nous supposerons alors que les agents connaissent aussi celles des nœuds voisins. Cette hypothèse ne nuit pas à la modularité du système dans la mesure où ces informations peuvent être transmises à un agent par ses voisins.
- Topologie du réseau : nous considérerons qu’un agent est capable de communiquer avec tous ses voisins physiques (éventuellement par l’intermédiaire d’un nœud tierce), ainsi qu’avec les postes de supervision. Ce deuxième type de communication est unilatéral : un poste de supervision ne peut pas envoyer de messages à un nœud spécifique.
**Définition du problème**

Pour ne pas constituer un point faible dans le système, un poste de supervision ne peut pas occuper de position centrale dans une architecture décentralisée. Dans le cas de notre application de surveillance et de pistage, son rôle devrait donc, selon ce critère, simplement consister à collecter les informations et à les présenter à un opérateur. Cependant, les postes de supervision sont les seuls nœuds qui reçoivent des informations de l’ensemble du système et avec lesquels un opérateur humain est susceptible d’interagir. Ces spécificités leur confèrent une position privilégiée pour accomplir certaines tâches supplémentaires. Celles-ci débordent du cadre strict de la fusion de niveau un et deux et concernent les points suivants.

- **Moteur d’alarmes** : contrairement aux nœuds de détection, un poste de supervision dispose d’une vue d’ensemble sur toute la zone surveillée ainsi que sur les cibles qui y sont détectées. Il est donc le mieux placé pour réaliser une analyse globale de la situation, identifier des comportements de groupe ou définir la menace que représente un ensemble d’événements.

- **Configuration du système** : en cours de fonctionnement, seul un poste de supervision peut présenter une interface à un opérateur. Par conséquent, tout paramétrage dynamique du système doit nécessairement être effectué à partir d’un poste de supervision. Puisqu’aucun message ne peut être envoyé à un nœud spécifique, cette fonctionnalité nécessite de mettre en place un mécanisme de propagation des informations à travers tout le réseau.

Ces deux points seront détaillés dans le chapitre 7, lors de la description du démonstrateur.

**6.1.2 Objectifs et difficultés**

Dans le système présenté dans la section précédente, chaque agent de fusion va collaborer avec ses voisins afin que toutes les pistes envoyées aux postes de supervision constituent un ensemble cohérent et non redondant.

Pour atteindre cet objectif, nous sommes confrontés à de nouvelles difficultés absentes dans un système centralisé. Il s’agit des détections simultanées ou consécutives d’une même cible par plusieurs nœuds.

**Détections simultanées d’une cible par plusieurs nœuds.**

Si une cible est détectée simultanément par plusieurs nœuds, ceux-ci doivent se mettre d’accord afin de ne pas envoyer deux pistes aux postes de supervision. Dans la mesure du possible, ils doivent également partager leurs informations afin que toutes les détections disponibles puissent améliorer la localisation de la piste.

**Détections consécutives d’une cible par plusieurs nœuds**

Lors de son passage à travers la zone surveillée, une piste peut être détectée successivement par plusieurs nœuds. Ceux-ci doivent alors collaborer pour reconstruire le trajet complet de la piste à partir des données locales. Notons que si les zones
surveillées par les capteurs ne se chevauchent pas, l’absence de redondance peut rendre cette tâche plus difficile.

6.2 Principes de la méthode

Sur la base des considérations développées ci-dessus, nous proposons une méthode de fusion décentralisée s’appuyant sur trois principes : l’application locale de la fusion multi capteurs classique, l’unicité des pistes et l’unicité des associations.

6.2.1 Application locale de la fusion

Chaque agent procède localement à la fusion des détections selon la méthode décrite dans les chapitres précédents. Les spécificités liées à l’organisation décentralisée vont alors se limiter à la répartition des détections et des pistes entre les nœuds.

6.2.2 Unicité des pistes

Chaque piste est gérée par un seul agent de fusion. Celui-ci est le seul autorisé à créer, mettre à jour et faire disparaître la piste dont il est gestionnaire. Lorsqu’un intrus se déplace, un agent peut céder une piste à un agent situé sur un autre nœud, auquel cas il en perd les droits de gestion. De même, seul l’agent gestionnaire d’une piste peut envoyer des mises à jour aux postes de supervision.

Le respect de ce principe assure donc de manière simple l’absence de duplication des envois d’une piste aux postes de supervision.

6.2.3 Unicité des associations

Pour éviter d’associer une détection à plusieurs pistes, chaque agent est gestionnaire des détections provenant de son nœud. Lui seul peut alors attribuer une détection à une piste, même s’il n’en assure pas la gestion. Il envoie alors la détection à l’agent gestionnaire de la piste qui se charge de la mise à jour. Il se peut donc que l’association ne soit pas réalisée sur le même nœud que la mise à jour.

6.3 Communications entre agents

Sur base des principes d’unicité des pistes et d’unicité des associations, des mécanismes d’échange d’informations ont été développés afin que les agents puissent attribuer leurs détections et céder la gestion de leurs pistes à des voisins.

6.3.1 Attribution des détections

Lorsqu’un agent dispose de nouvelles détections, il entame un dialogue avec ses voisins pour essayer de les attribuer à des pistes existantes. Cette négociation s’ef-
fectue en trois temps comme illustré sur les figures 6.1 et 6.2. L’agent commence par demander à chaque voisin s’il gère des pistes susceptibles d’être associées aux détections. Le cas échéant, le voisin calcule le coût élémentaire d’association et il renvoie le résultat. Après réception de toutes les réponses, l’agent gestionnaire des détections procède à l’association globale (sur base des coûts élémentaires envoyés par les voisins) et communique à chaque voisin les détections associées à des pistes dont il est le gestionnaire.

**6.3.2 Création d’une piste**

Selon le dialogue proposé dans la section précédente, une nouvelle piste est créée lorsqu’une détection n’est attribuée à aucune piste existante (figures 6.3 et 6.4). C’est donc tout naturellement le premier agent ayant détecté la nouvelle piste qui en sera le premier gestionnaire.

Nous pouvons noter une faiblesse potentielle du principe d’attribution des détections que nous venons de décrire. Lorsqu’une piste est détectée simultanément par deux voisins, un des deux nœuds sera gestionnaire de la piste et le deuxième devra lui transmettre toute ses détections, après requête. Si un seul message est perdu, l’agent qui avait cédé toutes ses détections jusque là, va considérer que la dernière ne peut être associée et va créer une nouvelle piste. Lorsque des données
sont perdues lors des communications, il y a alors un risque de dédoublement de piste. En pratique, il est donc nécessaire d’attendre plusieurs détections successives non associées avant d’initier une nouvelle piste.

6.3.3 Transfert d’une piste

Lorsqu’une piste n’est plus mise à jour par des détections locales, l’agent gestionnaire demande aux voisins qui continuent à fournir des détections, s’ils acceptent de prendre en charge la piste. Sur base des coûts renvoyés par ceux-ci, le gestionnaire
peut alors décider de renoncer à la gestion de la piste. Pour ce faire, il envoie la piste au voisin choisi et supprime la copie locale (figures 6.5 et 6.6).

Fig. 6.6: Diagramme des interactions lors de la transmission d’une piste

6.3.4 Disparition d’une piste

Tout comme une détection non associée induit la création d’une nouvelle piste, l’absence de détections provoquera la disparition de la piste. C’est le gestionnaire qui décide de supprimer une piste après avoir constaté l’absence de nouvelles détections pendant un délai fixé.

6.4 Conclusion

La fusion décentralisée exposée dans ce chapitre est une adaptation du cas centralisé qui avait été développé dans les chapitres précédents. Les nouvelles contraintes sont alors intégrées en implantant un agent de fusion sur chaque nœud du système et en définissant des stratégies de transferts de pistes et de détections. La cohérence de l’ensemble est alors assurée par les droits de gestion des détections et des pistes. Grâce à cette approche simple, une piste peut être suivie de manière cohérente lors de ses détections successives ou simultanées par plusieurs nœuds.

Il faut toutefois aussi mentionner le principal inconvénient de cette approche. Il s’agit de la quantité et de la fragilité des informations échangées. Non seulement ces communications nuisent à la discrétion du système, mais la qualité du pistage est également assez sensible à la perte de messages.
Chapitre 7

Système de surveillance Smartmesh

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Description générale du projet</td>
<td>106</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Capteurs et sources de données</td>
<td>106</td>
</tr>
<tr>
<td>7.1.2</td>
<td>Découpage fonctionnel du projet</td>
<td>107</td>
</tr>
<tr>
<td>7.1.3</td>
<td>Composants logiciels en interaction avec la fusion</td>
<td>109</td>
</tr>
<tr>
<td>7.2</td>
<td>Agent de fusion SmartMesh</td>
<td>112</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Plan de déploiement</td>
<td>112</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Construction des BBAs</td>
<td>114</td>
</tr>
<tr>
<td>7.2.3</td>
<td>Gestion des communications</td>
<td>115</td>
</tr>
<tr>
<td>7.2.4</td>
<td>Module de fusion</td>
<td>116</td>
</tr>
<tr>
<td>7.3</td>
<td>Installation et configuration du système</td>
<td>116</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Préparation du déploiement</td>
<td>117</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Initialisation de la fusion</td>
<td>117</td>
</tr>
<tr>
<td>7.3.3</td>
<td>Configuration dynamique</td>
<td>118</td>
</tr>
<tr>
<td>7.4</td>
<td>Conclusions</td>
<td>119</td>
</tr>
</tbody>
</table>
7.1 Description générale du projet

Le projet SmartMesh vise à étudier et à réaliser le démonstrateur d’un système de surveillance modulaire constitué de nœuds de détections autonomes en énergie et communiquant entre eux au moyen d’un réseau sans fil ad hoc. Chacun des nœuds du dispositif est composé d’une base générique comportant une plate-forme énergétique, une unité de calcul, un module de communication, un dispositif de localisation et des interfaces permettant le branchement de différents détecteurs. Les pistes, construites par recoupement des informations provenant des capteurs, sont alors rapportées à un poste de supervision distant (PS). Le poste de surveillance présente à l’opérateur l’ensemble des informations disponibles, à savoir les positions de tous les capteurs déployés, l’état des batteries, les tentatives de neutralisation des nœuds et bien entendu la localisation des pistes. Lorsqu’il y a des caméras parmi les capteurs et que celles-ci envoient des images au PS, ces dernières sont également affichées.

SmartMesh a été l’occasion de tester le comportement des agents de fusion distribuée en situation réelle. Dans ce chapitre, nous allons tout d’abord décrire brièvement l’ensemble des éléments constituant le système. Le module de fusion sera ensuite décrit plus spécifiquement et la troisième section sera consacrée aux procédures d’initialisation et de configuration du système. Les différentes expériences menées grâce au démonstrateur seront détaillées dans le chapitre suivant.

Précisons ici que beaucoup de partenaires ont participé à la réalisation de ce projet. Ces sociétés et organismes, dont les rôles seront détaillés au cours de la description du projet, sont : Sagem, Reflex CES, INEO, Orelia, Evitec, Inria, CEA List et Accuwatt.

7.1.1 Capteurs et sources de données

Quatre types de détecteurs peuvent être branchés à chacun des nœuds du démonstrateur. Ils ont été choisis afin de pouvoir tester le système avec des capteurs de différentes natures, fournissant des informations ayant une précision et une complexité variable. Ces capteurs sont des caméras IRNR, caméras visibles, capteurs acoustiques et détecteurs PIR.

Caméras IRNR

Les images de ces caméras thermiques non refroidies sont traitées par un logiciel développé par la société Evitech. Ce capteur, associé au traitement des images, permet de détecter des intrus de jour comme de nuit jusqu’à une distance de plus de 50 m. Il peut également dénombrer des intrusions simultanées et, pour chacune des cibles, fournir des informations telles que l’estimation de la taille, de la vitesse ainsi que la distance qui sépare de la caméra. Les cibles sont en outre pistées au cours du temps et des images peuvent être envoyées au poste de surveillance afin que l’observateur puisse analyser la situation de visu. Ce détecteur fournit donc les informations les plus riches, il faut toutefois noter qu’il est également le plus coûteux et exige plus de ressources de calcul et d’énergie.
L’allusion au pistage réalisé par le capteur thermique nécessite de préciser un point passé sous silence jusqu’ici. Lorsqu’une information de pistage permettant de faire le lien entre plusieurs détections successives d’un seul capteur est transmise à la fusion, cette information est exploitée lors de l’étape d’association. Une nouvelle détection est alors prioritairement associée à la piste de fusion regroupant les détections antérieures de la même piste mono capteur.

Caméras visibles

Les images de ces caméras visibles sont traitées par un logiciel que nous avons développé et la méthode utilisée est décrite en annexe.

La plupart des caractéristiques des détections IRNR s’appliquent également aux détections visibles mais ces deux capteurs ont chacun des avantages. Les capteurs visibles sont moins chers, consomment moins d’énergie et sont moins exigeants en ressources de calcul. En revanche, ils sont plus sensibles aux conditions climatiques et le traitement des images visibles n’est pas aussi robuste que le traitement des images IRNR.

Capteurs acoustiques

Les capteurs acoustiques sont des micros dont les signaux de sortie sont analysés par un algorithme développé par la société Orelia. Cet algorithme est capable de reconnaître des événements sonores appris lors d’une phase d’apprentissage.

Pour SmartMesh, les événements détectés sont les conversations humaines, les cris, les bruits de moteurs et les bris de verre. Ce capteur est relativement bon marché et il peut détecter des voitures jusqu’à une centaine de mètres.

Leur discrétion constitue un avantage souvent mis en avant. En revanche, la portée du capteur va dépendre du type de détection et aucune estimation de la distance ou la direction des détections n’est fournie. Un autre inconvénient réside dans le nombre restreint d’événements détectables.

Capteurs PIR

Il s’agit de détecteurs passifs infrarouge. Le modèle utilisé dispose de 15 faisceaux d’une largeur de 4°. Au total, ce capteur couvre donc une ouverture de 60° avec une portée annoncée de 35m. Ce capteur bon marché est très fiable mais il est incapable de distinguer plusieurs intrus présents en même temps ni d’estimer la distance.

7.1.2 Découpage fonctionnel du projet

Lors de l’étude préparatoire, cinq points techniques clés ont été identifiés pour mener à bien le développement des nœuds SmartMesh. Outre le module de fusion, il s’agit de la définition et de la réalisation matérielle du nœud, de la mise en place du réseau et des services de communication, de la gestion de l’énergie ainsi que de la réalisation d’un pupitre de supervision.
Architecture matérielle

La définition de l’architecture matérielle des nœuds a exigé la recherche d’un compromis entre les besoins en ressource de calcul et les contraintes énergétiques. Ces discussions arbitrées à la fois par Sagem en charge du pilotage et la société Reflex CES responsable de la fabrication des nœuds ont conduit à la définition d’un calculateur mixte composé d’un ARM et d’un processeur Atom X86. La fusion, le traitement des images visibles et les détections acoustiques ont été implantés sur l’ARM alors que le traitement des images IRNR\(^1\), nécessitant plus de ressources, est assuré par l’Atom. Outre ces deux calculateurs, chacun des nœuds est également équipé d’un GPS et d’un contact sec permettant de déclencher une alarme en cas de déplacement ou d’ouverture des nœuds. Ils disposent également d’un frame grabber\(^2\) pour la numérisation des images IRNR, d’un composant radio pour les communications, d’un module de gestion de la batterie et d’interfaces pour le branchement de tous les capteurs. Notons enfin que l’alimentation de chacun de ces composants peut être coupée par commande depuis l’ARM afin de réduire la consommation d’énergie.

Communications

Le module de communication et l’ensemble des services de routage et de configuration du réseau sont pris en charge par l’INRIA. D’un point de vue utilisateur, ce réseau dispose d’une couche de transport UDP avec un adressage IPv6. La couche physique suit le standard 802.15.4 avec une compression des en-têtes de type 6Low-PAN. Ces contraintes limitent le débit théorique à 250 kbps et le MTU\(^3\) à 70 octets.

---

1. Infra Rouge Non Refroidi
2. Composant destiné à capturer des images numériques à partir d’un signal analogique
Les images envoyées au pupitre de supervision doivent donc être drastiquement compressées pour ne pas saturer le réseau.

**Gestion de l’énergie**

L’autonomie énergétique des nœuds est également un élément important du projet. Cette autonomie a nécessité un effort de réduction de consommation de la part de tous les composants, mais aussi un dimensionnement et une gestion efficace de la batterie ainsi que l’estimation de son état de charge. Ce travail de caractérisation des produits existants, de dimensionnement et de gestion de la batterie a été réalisé par le CEA List et la société Accuwatt.

**Poste de supervision**

Le poste de supervision présente à l’opérateur l’état du système ainsi que la localisation et les caractéristiques des détections. Cette interface est développée par la société Ineo à partir du logiciel *Open Control*. Chacun des nœuds y est localisé sur un fond de carte par une icône indiquant d’éventuelles anomalies telles que la faible charge de la batterie, le déplacement d’un nœud, l’ouverture d’un boîtier ou encore l’absence d’un nœud dans le réseau. De même, les détections y sont affichées au moyen d’une icône ou d’une photo si une caméra a contribué au pistage de l’intrus. En cliquant sur cette icône il est également possible de visualiser dans une fenêtre contextuelle certaines informations telles que la taille, la vitesse de déplacement ou la classification de la piste.

**7.1.3 Composants logiciels en interaction avec la fusion**

Dans le cadre du projet *SmartMesh*, la fusion distribuée est réalisée au moyen d’agents implantés sur chaque nœud de détection ainsi que d’un agent spécifique situé sur la passerelle du poste de surveillance (PS).

Comme illustré sur la figure 7.3, chaque agent de fusion dispose d’interfaces avec les capteurs ainsi qu'avec les composants chargés du réveil des capteurs endormis.
et de l’envoi d’images vers le poste de supervision. D’autres exécutables tels que ceux chargés de la gestion de l’énergie ou de la détection de l’ouverture du nœud, sont également présents sur chaque nœud, mais ces composants n’ayant aucune interaction avec la fusion, leurs rôles et interfaces ne seront pas abordés.

Pour des raisons de compatibilité logicielle, l’interface entre le PS et le système de détection est assurée au moyen d’une passerelle. Il s’agit d’un nœud particulier sur lequel aucun capteur n’est branché et qui est relié au poste de supervision par une connexion éthernet. Sur ce nœud de passerelle, l’exécutable de fusion collecte les données et met en forme les pistes pour l’affichage. Il assure également certaines fonctions de supervision propre à la fusion telle que la configuration du déploiement ou l’initialisation du système. C’est également sur ce nœud de passerelle que se trouve le serveur chargé de stocker les images envoyées par les nœuds de détection.

**Fig. 7.3:** Diagramme des composants

**Interfaces avec les capteurs**

Toutes les détections provenant des capteurs sont transmises à la fusion sous forme de *traps SNMP*. Il s’agit d’un protocole initialement destiné à l’administration et à la surveillance d’équipements distants, dans lequel les *traps* constituent les messages d’alerte pour signaler des dysfonctionnements. Depuis son introduction, ce protocole a également été largement adopté par la communauté des systèmes de surveillance car il permet aussi bien d’envoyer les détections que de configurer les capteurs.
Bien que le choix de ce protocole ait rapidement été approuvé par l’ensemble des partenaires, il n’a jamais été possible de converger vers un formalisme commun. Il a donc été nécessaire de développer des interfaces distinctes pour traiter les messages provenant des différents capteurs.

**Réveil de capteurs**

Afin d’accroître l’autonomie énergétique des nœuds, la chaîne d’acquisition et de traitement des images IR s’arrête automatiquement lorsqu’aucune détection ne survient. C’est alors à la fusion que revient la responsabilité de redémarrer le processus. Cette fonction annexe est menée selon le principe suivant. Lorsqu’un agent de fusion reçoit une détection locale, il envoie un message de réveil à ses voisins. Le nœud initial, ainsi que tous ses voisins redémarrent alors les capteurs endormis. Ce procédé très simple permet, grâce aux relations de voisinage entretenues par chaque agent de fusion, de maintenir éveillés des capteurs distants tant qu’il y a des détections.

**Envoi d’images**

Outre le réveil des capteurs, la fusion est aussi chargée de l’envoi d’images des intrus détectés. Cette tâche secondaire doit être assurée par l’agent de fusion car il est le seul à pouvoir faire le lien entre la détection d’une caméra et la piste correspondante envoyée au poste de supervision.

La capacité du réseau de *SmartMesh* autorise l’envoi d’une image de 4 ko à partir de chaque nœud toutes les 10 secondes. Lorsqu’un nouvel envoi peut être effectué, l’agent de fusion va alors choisir une image en alternant, les pistes représentées. L’image est ensuite découpée pour ne garder que la boîte englobante de la détection (2 fois la hauteur et la largeur), compressée et transmise à la fonction d’envoi. Si l’envoi est réalisé avec succès, l’agent signale à la passerelle qu’une nouvelle image a été envoyée.

![Fig. 7.4: Exemple d’images envoyées par le réseau SmartMesh](image)

**Interface avec le poste de supervision**

Tous les messages provenant des agents de détections et destinés au poste de supervision transitent par le module de fusion situé sur la passerelle. C’est donc celui-ci qui assure l’interface avec le logiciel *Open Control*. 


Ces données sont de deux types : d’une part les pistes produites par la fusion et d’autre part les messages signalant qu’une nouvelle image est disponible sur le serveur. À l’instar de l’interface entre les agents de fusion et les capteurs, ces informations sont transmises sous forme de *traps SNMP*.

### 7.2 Agent de fusion SmartMesh

L’agent de fusion *SmartMesh* interagit avec les autres composants du système selon l’organisation présentée dans la section précédente. Son rôle est d’assurer la fusion des détections selon l’approche crédibiliste décentralisée détaillée dans les premiers chapitres. Pour réaliser cette tâche, chaque agent se compose des modules représentés sur la figure 7.5 et qui seront détaillés par la suite. Il s’agit d’un module gérant le plan de déploiement, d’un autre chargé de la construction des *BBAs*, d’un gestionnaire des communications et d’un module de fusion.

![Composants d’un agent de fusion](image)

**Fig. 7.5:** Composants d’un agent de fusion

#### 7.2.1 Plan de déploiement

Chaque agent de fusion doit connaître les informations locales, ainsi que celles relatives aux nœuds voisins. En pratique, pour le démonstrateur *SmartMesh*, les agents disposent d’un plan décrivant la totalité du déploiement. Dans la mesure où il ne s’agit que d’un démonstrateur, ce choix est raisonnable car le nombre de nœuds constituant le système est limité à dix. Notons que cette approche n’est pas conforme à la définition stricte d’un système décentralisé car chaque nœud dispose de ce fait de toutes les informations du déploiement. Elle constitue toutefois un compromis acceptable car en dépit de ses connaissances, le réseau ne permet pas à un agent de communiquer avec tous les nœuds distants.
Deux types d’informations sont présentes sur ce plan : les informations relatives au déploiement du système de surveillance et les données topographiques, ces dernières étant optionnelles.

Déploiement du système

Les informations relatives au déploiement du système de surveillance sont constituées de la description de chacun des nœuds de surveillance. Un nœud est décrit par un identifiant, une position plane et une adresse IP. Tous les capteurs branchés sont également spécifiés. Ils peuvent être de type caméra ou PIR, auquel cas une position, une orientation ainsi qu’une portée sont précisées. Ils peuvent également être acoustiques, seules la position et la portée sont alors spécifiées. En résumé, le déploiement du système est décrit par le diagramme de classes suivant.

![Diagramme des classes du déploiement](image)

Données topographiques

Le plan de déploiement peut également inclure une carte décrivant certains éléments topographiques. Pour le démonstrateur, les objets constituant la carte peuvent être des routes, des bâtiments ou des barrières. Chacun de ces éléments est décrit par son empreinte au sol (une ligne d’au moins deux points pour les routes ou les barrières et un polygone pour les bâtiments). Les éléments de type routes sont aussi décrits par une largeur ainsi que les types d’objets (véhicule, piéton ou les deux) auxquels leur effet attractif s’applique. Enfin, les bâtiments et les barrières possèdent un paramètre de hauteur, uniquement utilisé pour l’affichage et un indicateur de transparence. Ce dernier permet de favoriser les positions non masquées, sans exclure totalement la possibilité que l’intrus se trouve derrière l’obstacle.
7.2.2 Construction des BBAs

Ce module est chargé de générer les BBAs modélisant l'imprécision de localisation relative aux détections provenant des capteurs. Celles-ci sont construites conformément à la méthode présentée dans la section 3.3. Une première BBA est d’abord définie à partir de la détection et des caractéristiques du capteur avant d’être combinée avec une deuxième BBA modélisant les informations topographiques.

**BBAs de détection**

Étant donné que tous les capteurs de SmarMesh localisent les cibles dans un référentiel relatif, il faut connaître la position et l'orientation du capteur pour déduire la localisation absolue des détections. La forme d’une BBA dépend donc à la fois de la détection et de la position du capteur.

Les informations relatives à la position d’une détection renvoyées par les capteurs sont toutes en coordonnées polaires. Qu’il s’agisse d’une direction, d’une distance ou d’un rayon de portée, toutes ces informations peuvent en effet s’exprimer par un angle ou un rayon. Il est donc naturel de répercuter ces deux dimensions dans la forme des BBAs. Ainsi, tous les éléments focaux auront des formes symétriques par rapport à l’axe passant par la détection et le capteur. Plus précisément, les BBAs des capteurs acoustiques seront des disques consonants dont le rayon est fixé par la portée. Les BBAs des PIRs seront des secteurs de disques car ceux-ci estiment la direction des détections. Enfin, étant donné que les caméras fournissent à la fois une direction et une distance, elles formeront des BBAs constituées de secteurs d’anneaux consonants.

**Prise en compte de la topographie**

Les barrières et les bâtiments présents dans le champ d’un capteur donnent lieu à une BBA de masquage. Cette BBA peut être catégorique si l’obstacle est totalement opaque, ou simple si ce dernier est partiellement transparent. Toutes ces BBAs sont ensuite combinées de manière conjonctive.
La prise en compte des routes favorise la localisation des détections sur celles-ci. Suivant la procédure proposée dans la section 3.3, celles-ci sont modélisées par des BBAs simples et sont combinées entre elles de manière disjonctive.

### 7.2.3 Gestion des communications

Le module de gestion des communications fournit une interface spécifique à l’application. Il est chargé de rendre les messages conformes aux contraintes de taille et de capacité imposées par le réseau ad hoc de SmartMesh. Pour réaliser cet objectif, ce module va non seulement coder, découper et reconstruire les messages échangés entre agents, mais il va également simplifier la représentation des BBAs afin de réduire la taille des messages de description des pistes.

#### Mise en forme des messages

La taille des messages envoyés ne peut dépasser 70 octets et le respect de ce format n’est pas assuré par les services du réseau. Le découpage et la reconstruction des messages doivent donc être gérés au niveau applicatif. En ce qui concerne l’agent de fusion, cette restriction implique deux conséquences. D’une part, les messages doivent être concis, afin d’éviter autant que possible de dépasser la taille critique. D’autre part, la gestion des longs messages doit être assurée lorsque cela s’avère nécessaire. Il est évidemment préférable de limiter ces cas en raison de l’augmentation du risque de pertes de données et de la charge sur le réseau.

En pratique, tous les messages de détections sont envoyés en un seul paquet inférieur à 70 octets. Le message n’inclut alors pas la BBA de localisation qui peut être reconstruite sur le nœud destinataire grâce au plan de déploiement. Cette technique ne peut cependant pas être appliquée aux pistes puisque leur BBA, résultat d’un grand nombre de combinaisons, n’a pas une forme prédéterminée. De même, un plan de déploiement envoyé par le réseau dépasse largement les 70 octets.

Lorsqu’un message est découpé en plusieurs morceaux, l’en-tête du premier paquet informe alors le destinataire de la taille totale du message. Un numéro d’ordre est aussi ajouté à chaque paquet afin d’assurer la reconstruction à l’arrivée et vérifier que rien n’a été perdu.

#### Simplification des BBAs

Comme il n’est pas possible d’éviter l’envoi des BBAs de localisation des pistes, la taille des messages est également réduite en simplifiant leur représentation. Cette simplification est faite à deux niveaux. Tout d’abord, le nombre d’éléments focaux est réduit par la technique présentée dans le chapitre 5. Ensuite, chaque élément focal est approché par sa boîte englobante. Cette approximation, bien que sévère, permet dans la plupart des cas d’envoyer l’état d’une piste en un seul paquet.
7.2.4 Module de fusion

Ce module constitue le cœur de l’algorithme de fusion. Il se charge de l’association et de la fusion des éléments ainsi que de la gestion des pistes.

Modes de fonctionnement

Cette fonction, initialement prévue pour assurer une intégration progressive de la fusion dans le démonstrateur, a également permis de tester l’effet de la fusion. Trois modes de fonctionnement, modifiables par propagation d’une commande sont ainsi possibles. Il s’agit des modes de fusion transparente, locale et globale.

Lorsque l’agent opère de manière transparente, aucune fusion n’est effectuée. Il se contente alors de représenter les détections selon un formalisme homogène et de les envoyer vers la passerelle du poste de supervision. Chaque détection donne donc naissance à une piste, sans durée dans le temps.

En fusion locale, chaque agent traite les détections de son nœud sans se soucier de ce qui se passe chez les voisins. Toutes les pistes sont envoyées vers la passerelle, de sorte qu’un intrus détecté simultanément par deux nœuds génère deux pistes différentes. De plus, le lien entre des bouts de pistes détectés successivement par deux nœuds n’est pas assuré.

Enfin, le mode global réalise la fusion décentralisée présentée dans le chapitre 6.

Gestion des pistes

La gestion des pistes comprend l’envoi régulier de l’état des pistes à la passerelle du PS, la création de nouvelles pistes et la suppression des vieilles pistes.

Dans le cadre du démonstrateur, l’état de toutes les pistes est envoyé chaque seconde au PS. Ce délai correspond à la fréquence d’envoi des détections depuis les capteurs vers l’agent de fusion mais il pourrait être allongé pour des raisons de discrétion.

Selon l’application stricte de la stratégie décentralisée, chaque détection qui n’est associée à aucune piste doit initier la création d’une piste. Cependant, suite à l’observation de pertes de données sur le réseau, ce critère a été modifié afin d’éviter la construction de pistes parasites. Le gestionnaire de pistes attend alors deux non associations successives d’une détection avant de rendre une piste effective.

Le critère de suppression d’une piste est le nombre de cycles consécutifs sans qu’aucune détection ne lui soit associée. Bien que ce critère puisse être affiné en tenant compte par exemple de l’incertitude de localisation, il présente néanmoins l’avantage d’être simple à ajuster sur base d’observations.

7.3 Installation et configuration du système

L’objectif du projet SmartMesh était de réaliser le démonstrateur d’un système de surveillance et d’en tester le comportement sur le terrain lors de scénarios. La réalisation d’un produit fini ne faisait donc pas partie du cahier des charges, mais un
protocole de déploiement suffisamment simple a du être défini afin que le système puisse être installé dans un temps raisonnable par quatre personnes.

En ce qui concerne la fusion, la configuration comprend trois étapes. Avant l’installation, il faut définir la position approximative de tous les nœuds et établir un plan de déploiement. Une fois sur le terrain, chaque nœud est installé en limitant les manipulations informatiques. Enfin, en fonction des aléas, certains paramètres tel que le plan de déploiement ou le mode de fonctionnement doivent pouvoir être modifiés depuis le poste de supervision.

Chacune de ces étapes a nécessité un effort d’ergonomie, dont certains éléments avaient été identifiés dès la rédaction du cahier des charges, mais dont d’autres ne se sont imposés qu’au cours des expériences.

7.3.1 Préparation du déploiement

Spécifications du format

Avant l’installation du système de surveillance, il faut préparer un plan de déploiement et le transmettre à tous les agents sous la forme d’un fichier. Pour simplifier cette tâche et par souci de standardisation, nous avons proposé d’utiliser le langage *KML*\(^4\). Ce langage, développé par l’OGC\(^5\), utilise le formalisme *XML* et est destiné au partage de données géo-spatiales. Le choix de ce langage nous permet d’utiliser un des nombreux logiciels suportant ce format pour géo-référencer et dessiner la géométrie des éléments. Les informations spécifiques à l’application telles que l’adresse IP des nœuds ou le type de capteurs peuvent être ajoutées grâce à une balise destinée à la personalisation des objets.

Nous avons spécifié toutes les balises nécessaires à la description des éléments en nous appuyant sur le format *KML*. Le résultat détaillé en annexe a été validé par la société Ineo utilisant le même plan pour la représentation du site.

7.3.2 Initialisation de la fusion

Lecture des données de configuration

Chaque agent de fusion lit un fichier de configuration lors de sa phase d’initialisation. Ce fichier, comprend un grand nombre d’informations, toutes optionnelles et aussi diverses que les paramètres de construction des *BBAs*, la fréquence de travail ou le nom du fichier de déploiement. Des copies de ce fichier doivent être présentes sur chaque nœud avant le démarrage de la fusion mais celles-ci peuvent être identiques.

Acquisition d’un identifiant

Pour opérer, chaque agent doit connaître la position et l’orientation des capteurs de son voisinage. Bien que ces informations soient renseignées dans le plan de déploiement, il est parfois nécessaire de les obtenir par le biais de capteurs de positionnement et d’orientation spécifiques.
ploiement, l’agent doit savoir qui il est pour pouvoir en extraire les informations qui le concerne. Il doit donc disposer d’un identifiant lui permettant de se repérer. Ce dernier peut être spécifié dans le fichier de configuration, mais cela nécessite de personnaliser la configuration de chaque nœud.

Afin d’éviter cette procédure, une deuxième méthode a été mise en place. Lorsqu’un agent ne trouve pas d’identifiant lors de son initialisation, il va attendre que l’information lui soit communiquée par un voisin. Une fois cet identifiant reçu, il peut se repérer sur le plan et initialiser à son tour ses voisins. Ce mécanisme permet ainsi de configurer tous les agents de proche en proche à partir du poste de supervision.

L’utilisation d’un seul fichier de configuration pour toutes les instances de fusion s’est avérée très pratique lors des différents essais, mais il faut noter qu’avec ce mécanisme un nœud ne remonte aucune alarme tant que le réseau ne lui a pas transmis son identifiant.

Il a aussi été envisagé d’effectuer cette initialisation à l’aide du GPS. Avec cette approche, un nœud n’a pas besoin de ses voisins mais cette solution impose de placer les nœuds à des distances mutuelles minimales pour qu’il n’y ait pas d’ambiguïté entre les positions. Cette contrainte empêche donc de placer deux nœuds au même endroit, ce qui peut s’avérer utile pour colocaliser beaucoup de capteurs. Cette solution n’a donc pas été retenue.

**Construction du voisinage**

Pour la fusion, la notion de voisinage repose sur la proximité des nœuds dans le plan de déploiement. Deux nœuds sont voisins s’ils sont distants de moins de 150 m ou si l’intersection de leurs zones de surveillance n’est pas vide. Un agent de fusion ne peut donc rechercher les nœuds voisins sur le plan de déploiement que lorsqu’il a acquis son identifiant.

**7.3.3 Configuration dynamique**

La troisième phase de la configuration de la fusion concerne la modification dynamique de certains paramètres. Ces aspects incluent les messages d’exploration du voisinage et la propagation d’informations.

**Exploration du voisinage**

Les voisins d’un nœud sont définis par des critères purement géométriques, sans tenir compte de la topologie du réseau. Deux nœuds voisins peuvent donc être incapables de communiquer soit de manière permanente, soit pour une durée limitée.

Afin de définir les nœuds joignables, chaque agent interroge régulièrement ses voisins. Une table de communication est ainsi maintenue à jour et les messages de fusion ne sont envoyés qu’aux voisins effectifs.

Cette stratégie permet de réduire le quantité de messages envoyés et constitue une première étape vers une construction dynamique et décentralisée du plan de
déploiement. Pour ce faire, il faudrait encore renoncer à la présence d’un plan de déploiement global sur chaque nœud et mettre en place un mécanisme de construction d’une vue locale du déploiement par partage d’informations entre voisins.

**Propagation d’informations**

De manière générale, la propagation d’informations ne constitue pas une obligation du cahier des charges. Cependant la possibilité de modifier certains paramètres depuis le poste de supervision s’est vite imposée comme une nécessité, en raison du temps nécessaire pour accéder à chaque nœud.

Les propriétés de routage ne permettent pas au poste de supervision de communiquer avec un nœud particulier. Il est cependant possible de propager une information à tout le réseau. Pour ce faire, un numéro de version est ajouté à chaque information afin qu’un agent puisse comparer une nouvelle donnée reçue avec l’instance locale. Il peut ainsi déterminer si son paramètre doit être mis à jour, auquel cas il transmet à son tour l’information à ses voisins. Cette stratégie permet de stopper spontanément la propagation d’une information une fois que tous les nœuds sont mis à jour.

Les numéros de versions sont également présents dans les messages de sondage décrit dans la section précédente. Ainsi, un agent peut vérifier que sa configuration est à jour, même s’il n’était pas présent sur le réseau lors de la propagation des paramètres. Lorsqu’un agent identifie une version obsolète, il demande alors à un voisin la nouvelle valeur.

### 7.4 Conclusions

Bien que le système de surveillance *SmartMesh* présenté au cours de ce chapitre ne soit que le démonstrateur d’un système opérationnel, l’intégration de la fusion décentralisée a exigé un gros effort de développement. Cet effort a aussi bien concerné la définition de diverses interfaces que l’ajout de fonctions supplémentaires nécessaires à la bonne marche des expériences. Ce travail a toutefois permis de réaliser les expériences présentées dans le prochain chapitre, qui constituent une étape indispensable à l’évaluation de la méthode.
Chapitre 8

Essais et expérimentations

Sommaire

8.1 Plan d’intégration et d’expérimentation . . . . . . . . . . . . . . 122
8.2 Cadre expérimental . . . . . . . . . . . . . . . . . . . . . . . . . . 122
8.2.1 Outils d’analyse . . . . . . . . . . . . . . . . . . . . . . . . . 122
8.2.2 Présentation des sites de déploiements . . . . . . . . . . . 124
8.3 Scénarios de déplacement de cibles . . . . . . . . . . . . . . . . 126
8.3.1 Suivi d’une piste . . . . . . . . . . . . . . . . . . . . . . . . . 126
8.3.2 Croisement de deux pistes . . . . . . . . . . . . . . . . . . 127
8.3.3 Séparation d’un groupe . . . . . . . . . . . . . . . . . . . . 130
8.4 Étude de sensibilité . . . . . . . . . . . . . . . . . . . . . . . . . . 132
8.5 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 135
8.1 Plan d’intégration et d’expérimentation

Les premiers tests sur le système de surveillance $\textit{SmartMesh}$ ont été effectués durant la phase d’intégration du projet. Ces expériences, réalisées en laboratoire, ont notamment permis de valider les différentes interfaces. Le fonctionnement des nœuds a ensuite été stabilisé au cours d’essais plus longs effectués sur le terrain et ce n’est qu’après cette phase de mise au point que les tests fonctionnels ont pu débuter.


Tous les essais ont été menés selon le mode opératoire suivant. Avant la réalisation effective des expériences, un plan de déploiement et des scénarios d’intrusion sont établis avec l’ensemble des intervenants. En ce qui concerne la fusion, le dispositif est ensuite testé en trois étapes. Dans un premier temps, les scénarios sont simulés afin de vérifier leur faisabilité théorique et d’avoir une idée des résultats attendus. Les expériences sont ensuite réalisées avec le système réel et les détections sont enregistrées sur les nœuds. Enfin, les données enregistrées sont rejouées afin d’identifier d’éventuels problèmes ou comparer différentes configurations.

Dans la suite de ce chapitre, les sites de test et les outils développés pour analyser le comportement de la fusion seront d’abord présentés. Ensuite, deux paragraphes seront consacrés à l’analyse de scénarios simples et à une étude de sensibilité. Enfin, nous donnerons quelques conclusions sur la perception subjective des premiers utilisateurs du système.

8.2 Cadre expérimental

8.2.1 Outils d’analyse

Au cours du projet, trois outils d’analyse ont été développés pour étudier les résultats produits par la fusion. Il s’agit d’un visualiseur spécifique à la fusion, d’un simulateur de scénarios et d’un lecteur pour rejouer des séquences de détections enregistrées lors des essais.

Visualisation

L’interface de visualisation implantée sur le poste de supervision est destinée à un observateur chargé de la surveillance du site. Par conséquent, les informations présentées dans cette interface sont dédiées à cet utilisateur spécifique. Ainsi, certaines informations très utiles à la compréhension de la fusion ne sont pas représentées.
tées dans cette l'IHM. C’est par exemple le cas de l’orientation des capteurs ou encore des BBAs relatives à la localisation des pistes. Ce constat nous a amené à développer une interface graphique spécifique à la fusion. Celle-ci permet d’interroger la passerelle afin d’afficher les pistes et leurs BBAs sur un ordinateur tiers. Cet exécutable permet aussi de visualiser et de modifier le plan de déploiement afin d’envoyer des mises à jour à tous les nœuds grâce au mécanisme de propagation détaillé à la section 7.3.3.

Simulateur

Le simulateur de scénario est constitué des trois éléments suivants :
- un lecteur qui permet de simuler des intrusions à partir de descriptions de scénarios lues dans un fichier,
- un simulateur de capteurs qui génère des détections lorsqu’un intrus virtuel est détecté,
- une interface de visualisation semblable à celle développée pour la visualisation.

Grâce à ces composants, la fusion décentralisée est alors simulée en lançant sur des threads indépendants tous les agents renseignés dans le plan de déploiement. Parallèlement à cela, le lecteur joue un scénario d’intrusion et fournit au simulateur de capteurs les positions des cibles. Celui-ci envoie ensuite à chaque agent de fusion, les détections générées par les capteurs de son nœud. Enfin, les pistes reçues par la passerelle virtuelle sont visualisées dans l’IHM (figure 8.1).

Il est important de noter que dans cette configuration, les modules de fusion communiquent par l’hôte local. Ces simulations ne tiennent donc pas compte des pertes de données ni des délais de communication. Or, les essais nous ont montré que cette situation idéale n’est pas toujours représentative de la réalité.

---

1. Interface Homme Machine
Rejouer de scénarios

Ce troisième outil permet de rejouer les détections enregistrées lors des essais réels. Différentes configurations de la fusion peuvent donc être comparées avec des détections à la fois rigoureusement identiques et représentatives de situations réelles. Cependant, contrairement aux scénarios simulés, la comparaison de ces résultats avec une vérité terrain n’est pas possible.

Le principe du rejoue d’un scénario est comparable à une simulation pour laquelle le simulateur de capteurs est remplacé par un lecteur d’enregistrement. Tous les agents de fusion sont donc lancés sur des threads indépendants et le lecteur leur envoie les détections par l’hôte local. Notons enfin que les horloges des nœuds doivent impérativement être synchronisées avant l’enregistrement pour que les détections puissent être envoyées dans le bon ordre au moment du rejoue.

8.2.2 Présentation des sites de déploiements

Les essais ont été effectués sur deux sites de configuration différente. Le premier déploiement a été réalisé sur le site de l’INRIA Rocquencourt et le second sur le camp de la gendarmerie de Beynes.

Site de l’INRIA

Sur ce premier déploiement (figures 8.2 et 8.3), les nœuds ont été disposés autour d’une plaine de sorte qu’un seul des quatre nœuds ne puisse communiquer directement avec la passerelle. Ce déploiement est à la fois caractérisé par une configuration en étoile et par de nombreux recouvrements entre les zones surveillées par les capteurs. Le principal enjeu de la fusion a donc été d’éviter la multiplication des pistes lors des détections simultanées d’un intrus.

![Fig. 8.2: Visualisation du déploiement sur le site de l’INRIA](image-url)
de sauts d’horloges diagnostiqués sur certains nœuds. Cette phase de mise au point était cependant nécessaire pour atteindre la maturité du système.

**Fig. 8.3:** Vue du site et poste de supervision à l’INRIA

**Site de Beynes**

Pour ce deuxième déploiement (figures 8.4 et 8.5), les nœuds ont été placés le long d’une route selon une configuration linéaire. Le recouvrement spatial des zones surveillées était moins important que sur le site de l’INRIA et plusieurs nœuds devaient passer par un intermédiaire pour envoyer des messages à la passerelle. Les pertes d’information et les délais sans détection étaient donc aussi plus importants.

**Fig. 8.4:** Visualisation du déploiement sur le site de Beynes

C’est sur le site de Beynes qu’ont été réalisés la plupart des tests fonctionnels du système.
8.3 Scénarios de déplacement de cibles

Les échanges d’informations intervenant lors de la fusion décentralisée ont tout d’abord été testés lors du suivi d’une cible passant à proximité de plusieurs nœuds de détection. Ensuite, quelques scénarios faisant intervenir plusieurs cibles ont également été mis en œuvre. Dans la mesure où ceux-ci ne constituent une difficulté spécifique que si les cibles sont proches les unes des autres, ce cas de figure a notamment été testé par le croisement de deux pistes et par la séparation d’un groupe en deux.

Ces trois scénarios, illustrant à la fois les succès et les difficultés rencontrées, sont présentés dans la suite de cette section.

8.3.1 Suivi d’une piste

Le premier objectif de notre fusion consiste à suivre les déplacements d’un intrus. Pour réaliser cette tâche, la fusion exploite trois mécanismes : le regroupement des détectations initiées par une même cible, l’échange de détectations entre nœuds et le transfert de pistes d’un nœud vers un autre. Ces processus ont été testés par le passage d’un piéton aux abords de chaque nœud du déploiement de l’INRIA, et par le passage d’un véhicule sur le tronçon de route du camp de Beynes comme illustré sur la figure 8.6.

Les deux scénarios de pistage présentés ci-dessus ont, dans un premier temps, été testés en simulant des capteurs idéaux. Dans ce cas favorable, chacune des expériences a donné des résultats satisfaisants, validant de ce fait la faisabilité des essais. Plus précisément, la fusion ne génère qu’une seule piste qui suit la cible durant son parcours dans la zone surveillée et les échanges de détectations et de pistes entre les nœuds fonctionnent correctement. Comme le montre la figure 8.7, la reconstruction du trajet suivi par la piste est toutefois un peu chaotique en raison du manque de précision des capteurs.

Lors des essais réalisés sur le terrain, ce premier scénario a été joué plusieurs fois et s’est généralement déroulé comme le prévoyait la simulation. Cependant, bien que l’effet parasite ne se soit pas reproduit lors des rejeux, certains dédoublements de pistes ont eu lieu au cours des expériences. En analysant plus en détail les fichiers
Il arrive en effet qu'un message de demande de prise en charge d'une détection soit perdu. Comme déjà mentionné dans la section 6.3.1, cette perte va alors provoquer la création d'une nouvelle piste. Une fois diagnostiqué, le problème a pu être corrigé en imposant l'attente de plusieurs détections isolées consécutives avant d'initialiser une nouvelle piste mais cette correction n'a pas pu être portée sur les nœuds avant la fin du projet.

8.3.2 Croisement de deux pistes

Ce deuxième scénario comporte deux cibles parcourant le même trajet simultanément et en sens inverse. L'objectif est d'observer la manière dont la fusion répartit...
les détections entre les deux pistes et notamment d’analyser ce qui se passe lors du croisement.

Cette fois, une erreur de pistage apparaît dès le stade de la simulation. En effet, comme on peut le voir sur la figure 8.9 en notant les identifiants des pistes sur les trois images, les deux pistes ne se croisent pas et chacune retourne à son point de départ. Ce problème, ne survient pas de manière systématique et dépend de l’endroit du croisement. Il s’explique par le fait que lorsque les deux cibles se croisent en dehors d’une zone surveillée par un capteur capable de distinguer et suivre chacune des pistes, il est très difficile à la fusion de deviner si les pistes ont fait demi-tour ou s’ont croisées.

Notons toutefois que malgré cette erreur de trajectoire, aucune piste parasite n’est créée, deux pistes sont suivies durant toute la durée du scénario et après l’erreur.
du point de croisement, les détections sont attribuées à chacune des pistes de manière cohérente (sans oscillation).

**Fig. 8.10:** Essais du croisement de deux pistes

Le phénomène de duplication de pistes déjà mentionné à l’occasion du premier scénario est à nouveau observé lors des essais réels. De même, les deux pistes retournent vers leur point de départ au lieu de se croiser (figure 8.10). Malgré ces deux erreurs, les deux pistes initiées au début du scénario sont encore présentes à la fin de l’enregistrement.

**Fig. 8.11:** Rejeu du croisement de deux pistes

A nouveau, nous pouvons observer sur la figure 8.11 que le rejeu des détections se passe beaucoup mieux que la fusion réalisée en temps réel. Grâce à cette amélioration due à la disparition des problèmes de réseau, la qualité du pistage devient comparable à la simulation, à l’exception toutefois d’une petite période d’hésitation au moment du croisement des deux cibles. Cette différence s’explique par la détection tardive d’une cible par une caméra.
8.3.3 Séparation d’un groupe

Le dernier scénario présenté dans ce travail est celui de la séparation d’un groupe en deux pistes distinctes. Ce scénario s’est déroulé comme suit. Une voiture venant du sud, c’est-à-dire du bas de la carte, s’est arrêtée vers le milieu de la zone de test. Un passager est sorti et a fait le chemin inverse à pied, tandis que le véhicule a continué sa route vers le nord comme illustré sur la figure 8.12.

![Zone de séparation](image)

**Fig. 8.12:** Scenario de séparation d’un groupe

L’objectif de ce scénario est d’étudier la capacité de la fusion à initialiser une nouvelle piste à proximité d’une piste existante et à distinguer leurs trajectoires respectives.

![Simulation de la séparation de deux pistes](image)

**Fig. 8.13:** Simulation de la séparation de deux pistes

Nous pouvons noter sur la figure 8.13 que lors de la simulation, la piste nouvellement créée continue la route vers le nord plutôt que de faire demi tour. Cette erreur...
n’a toutefois rien d’étonnant puisqu’aucun des capteurs utilisés dans SmartMesh n’assure le pistage en cas de séparation d’un groupe. La fusion ne dispose donc d’aucune information lui permettant de savoir quelle branche a été suivie par la piste existante. De plus, dans la mesure où le piéton se trouvait à l’intérieur de la voiture, l’historique commun pourrait aussi être partagé par les deux pistes après la séparation. Une telle gestion de pistes n’a toutefois pas été prévue dans le cadre de ce projet.

Dans l’ensemble, ce troisième essai s’est plutôt bien déroulé (figure 8.14). En effet, bien que la nouvelle piste ait été détectée un peu tard, les trajectoires des deux pistes sont comparables aux parcours obtenus lors des simulations.

Dans la mesure où les essais ont été de qualités comparables aux simulations, il n’y a rien d’étonnant à ce que le rejeu le soit aussi (figure 8.15).
8.4 Étude de sensibilité

Suite à cette première validation, nous avons testé la sensibilité de la localisation et du pistage par rapport à quelques facteurs perturbants.

Absence de détections

Lorsqu’une cible n’est plus détectée pendant un délai trop important, la piste est détruite et une nouvelle piste est initialisée à la détection suivante. Cette absence de détection peut être due à une trop faible sensibilité des capteurs ou à un masquage temporaire de la cible.

La sensibilité du pistage à l’absence de détections est directement liée au critère de suppression des pistes. C’est en effet sur la base de ce délai que la fusion décide de faire disparaître une piste. Plus ce délai est long, plus le pistage d’une cible sera robuste aux absences de détections. Il faut toutefois noter qu’en l’absence de détections, l’incertitude de localisation augmente et avec elle le risque de mauvaises associations. De plus, la rémanence à l’écran de pistes ayant effectivement disparues nous a semblé plus gênante que la perte occasionnelle de pistes. En pratique, ce délai a été fixé à 15 secondes.

Fausses détections

Les fausses détections sont beaucoup plus délicates à traiter que l’absence de détections. En effet, bien que les pistes initiées par un capteur réputé peu fiable puissent recevoir une confiance très faible, une piste plus sûre risque en revanche d’être détournée de sa vraie trajectoire si elle est attirée par des détections erronées.

Vitesse de la cible

La qualité du suivi des pistes est également sensible à la vitesse de déplacement des cibles. C’est plus précisément la distance parcourue par la cible entre deux observations successives qui constitue le facteur limitant. Par conséquent, la vitesse critique au-delà de laquelle le pistage devient incertain est implicitement imposée par la fréquence de détection des capteurs.

Précision des capteurs

La localisation des cibles estimée par la fusion est sensible à la précision des capteurs. Bien que le recoupement d’informations redondantes permette de réduire les incertitudes, une partie du bruit d’observation sera toujours présent. L’influence de l’imprécision des capteurs a été testée en simulant un bruit gaussien sur les détections des capteurs avec des écarts-type de 15 m et 15°. Les erreurs de localisations après fusion ont ensuite été comparées aux résultats obtenus avec des capteurs idéaux (figure 8.16). Sur les deux sites, l’erreur moyenne est de 16 m avec des capteurs bruités, alors que l’erreur moyenne avec des capteurs idéaux est de 9 m sur le site de l’INRIA et de 13 m sur le site de Beynes.
Étude de sensibilité

Fig. 8.16: Erreurs de localisation sur les sites de l’INRIA (1) et Beynes (2). Influence de la précision des capteurs sur des scénarios simulés.

Décalage temporel

La fusion traite les détections provenant des capteurs à une cadence d’une seconde. Lors de ce traitement, le module doit dialoguer avec ses voisins pour échanger des informations et ensuite envoyer ses pistes à la passerelle. Il s’ensuit un retard cumulé inévitable entre la détection d’une cible et son affichage sur le poste de supervision. Ce retard reste négligeable par rapport aux délais d’intervention mais peut constituer une gêne si l’opérateur est suffisamment proche pour observer directement la scène. Ce retard a été estimé lors des simulations en comparant les erreurs moyennes de localisation pour différents décalages de temps. Il apparaît alors qu’une piste minimise l’erreur de localisation si elle représente la position de la cible deux à trois secondes auparavant (figure 8.17).

Exploitation de l’information a priori

La prise en compte de la route pour le scénario du camp de Beynes n’améliore que légèrement la localisation des pistes (±2 m). Cela s’explique par le fait que l’in-
certitude de localisation des capteurs est essentiellement longitudinale. En d’autres termes, les capteurs estiment beaucoup mieux l’orientation des détections que leur distance. Malheureusement, du fait du choix de l’orientation des capteurs parallèlement à la route, celle-ci n’apporte aucune information susceptible de corriger cette composante de l’erreur.

Il faut toutefois noter que si l’erreur métrique moyenne n’est pas améliorée, la méthode favorise toutefois le positionnement des pistes sur la route. Cette caractéristique n’est pas négligeable car l’expérience nous a enseigné qu’un observateur accorde beaucoup plus d’importance à la cohérence des positions qu’à son erreur absolue. Il est en effet visuellement plus gênant de voir une piste passer à travers un mur ou s’éloigner sans raison d’un chemin que de ne pas se trouver à la position exacte de la cible.

Erreur d’orientation des capteurs

Lors des déploiements, les capteurs ont été disposés sur le terrain en respectant approximativement le plan établi à l’avance. Ces installations ont été réalisées à vue et il y avait une différence notable entre la position réelle du capteur et sa position supposée par la fusion qui correspond à la position sur le plan de déploiement. Ces erreurs, et en particulier les erreurs d’orientations des capteurs qui sont souvent importantes, introduisent un biais systématique dans la localisation des détections.

Fig. 8.18: Erreurs de localisation avec des capteurs mals orientés : effet de la prise en compte des informations topographique

Ces erreurs étant systématiques, elles sont difficiles à corriger par le recouplement d’observations successives mais les informations a priori permettent de s’en affranchir partiellement. Ce principe a été testé en simulant le déplacement d’un véhicule sur le site de Beynes et en altérant l’orientation des capteurs d’environ 15° sur le plan de déploiement transmis à la fusion. Les erreurs de localisation, dont les histogrammes sont illustrés sur la figure 8.18 passe alors d’une moyenne de 20 m à 15 m grâce à la prise en compte des a priori topographiques.
8.5 Conclusions

L’implantation d’un module de fusion dans le démonstrateur SmartMesh a permis de tester les principes de la fusion décentralisée développés précédemment dans ce travail. Cette validation a été réalisée au moyen d’expériences menées sur le terrain, de rejeu d’enregistrements mais aussi de simulations de scénarios.

La fusion a montré un bon comportement lors de scénarios simples. Les mécanismes d’échange de détections entre agents de fusion et de transmission de pistes se déroulent généralement bien. De même, certains avantages de la représentation des incertitudes ou imprécisions au moyen de BBAs ont été mis en évidence lors de ces expériences. C’est par exemple le cas de la modélisation des formes d’incertitudes ou de la prise en compte d’a priori topographiques pour corriger la mauvaise connaissance de l’orientation des capteurs.

L’intégration de la fusion dans un système complet a aussi mis en évidence certaines fragilités. En particulier, la prise en compte des pertes d’informations est impérative pour assurer le bon fonctionnement de la fusion. De même, la fusion est sensible aux fausses détections car elles peuvent initier la création de fausses pistes ou dévier des pistes existantes de leur trajectoire réelle et de ce fait gêner leur suivi ultérieur (en empêchant leur association).

Enfin, ces essais ont aussi montré que la précision de la localisation ou le taux de bons appariements ne reflètent pas nécessairement la perception que les utilisateurs ont de la qualité du système de fusion. D’autres critères, dont l’intérêt dépend des cas d’utilisation ont ainsi pu être mis en évidence.

– La cohérence des positions relatives des pistes par rapport aux autres éléments semble plus importante que la précision absolue de la localisation. Il est ainsi préférable de placer les véhicules sur les routes et d’éviter que les piétons ne passent à travers les murs même si la localisation exacte des pistes n’est pas connue.

– La régularité des trajectoires est également un élément important. Il est préférable que les déplacements en dents de scie soient lissés, même si cela dégrade légèrement l’erreur de localisation.

– Le dédoublement des pistes semble plus gênant qu’une perte lors du suivi d’une piste.

Bien qu’ils ne remettent pas en cause les critères classiques utilisés, il serait intéressant d’intégrer ces nouveaux éléments dans les objectifs de la fusion.
Chapitre 9

Conclusions et perspectives
Notre étude se place dans le cadre applicatif des systèmes de détection multi capteurs. Plus précisément, notre objectif consiste à combiner les informations provenant de plusieurs capteurs afin d’affiner la localisation, suivre les déplacements et identifier les pistes. Outre le recoupement d’informations partiellement complémentaires et partiellement redondantes, la gestion simultanée de plusieurs cibles exige également de mettre en correspondance les nouvelles détections avec les pistes existantes.

D’un point de vue théorique, le problème a été abordé dans le contexte spécifique des fonctions de croyance. Ce choix, motivé par la richesse de représentation offert par ce formalisme, nous a amené à développer un processus de fusion comportant cinq contributions originales.

Représentation des incertitudes par des BBA

La première contribution concerne la représentation crédibiliste des incertitudes de localisation. Chaque détection est modélisée par une BBA définie sur un pavage de la scène. Les difficultés liées à la taille de ce cadre de discernement sont contournées en conditionnant les détections sur une petite région de la scène et en ajoutant un facteur d’échelle par grossissement. Avant d’être combinées, deux BBAs sont alors ramenées sur un même cadre de discernement par conditionnement et raffinement.

Prise en compte des a priori topographiques

Le modèle de représentation permet également d’intégrer facilement des a priori topographiques. Ceux-ci sont alors modélisés par des BBAs décrivant la croyance a priori qu’une cible soit détectée par un capteur dans une région spécifique de la scène. Ces BBAs sont ensuite combinées aux détections pour en affiner la localisation.

Un critère d’association s’appuyant sur la représentation de la localisation

La troisième contribution concerne la définition d’un critère d’association entre les pistes et les détections à partir de la même représentation crédibiliste des localisations. Ce critère, maximisant la probabilité pignistique jointe des associations, permet de réaliser de manière cohérente l’ensemble des traitements relatifs à la fusion sans avoir à définir un nouveau cadre de discernement.

Exploitation de la décomposition canonique

La représentation des BBAs au moyen de leur décomposition canonique a également été étudiée. Cette alternative présente l’avantage de permettre la combinaison prudente des détections successives d’un même capteur. De plus, une méthode adaptée de simplification des BBAs (nécessaire en raison du caractère récursif des traitements) a également été proposée. Il s’agit de la quatrième contribution théorique de ce travail.
Architecture décentralisée

Enfin, la dernière contribution a été de définir une architecture décentralisée pour la réalisation de l’ensemble des traitements. Chaque nœud du système collabore alors avec ses voisins afin que les informations envoyées au poste de supervision forment un ensemble complet et cohérent.

Validation expérimentale

La validation du système de fusion a constitué une part importante de ce travail. Celle-ci a été réalisée à la fois au moyen de simulations et d’essais menés sur le terrain, ces deux approches étant complémentaires.

Les simulations ont permis de tester indépendamment les différents aspects de la fusion. Certains choix ont ainsi pu être justifiés en comparant les performances de différentes solutions envisageables. Par exemple, la prise en compte des a priori topographiques lors de scénarios simulés a montré une diminution de l’erreur de localisation. De même, le critère d’association a donné de bons résultats pour l’application qui nous concerne. Enfin, les simulations ont grandement simplifié la paramétrisation des BBAs de détections.

Parallèlement aux simulations, la fusion a été testée lors de scénarios réels grâce à l’implantation d’un module dans le système de détection SmartMesh. Ces expériences ont été nécessaires d’une part pour quantifier de manière réaliste les erreurs relatives à chaque capteur mais aussi pour intégrer dans le plan de validation les difficultés liées aux interfaces avec les autres composants. C’est par exemple lors de ces essais que la nécessité d’être robuste face aux pertes du réseau a été mise en évidence.

Les essais ont aussi montré que du point de vue d’un observateur, la perception du bon fonctionnement de la fusion repose davantage sur la cohérence visuelle que sur des critères objectifs. Il est par exemple important que les pistes se déplacent le long des routes et que leur déplacement soit lissé. De même, le dédoublement des pistes semble visuellement plus gênant que la perte lors de leur suivi.

Il est enfin apparu lors des essais que la discrétion des nœuds semble être un argument clé pour les applications militaires.

Perspectives

Parmi les perspectives de ce travail, certaines concernent un point spécifique alors que d’autres, plus générales, ont émergé suite aux observations et discussions lors des essais.

Un premier point concerne l’association entre les pistes et les détections. Cette association, effectuée en minimisant un critère global, répartit de manière irréversible les nouvelles détections entre les pistes. Cette stratégie, motivée par l’aspect décentralisé de l’application, pourrait être améliorée en maintenant plusieurs hypothèses
d’associations selon une approche MHT. La difficulté est alors d’implanter, au moins partiellement, cette méthode selon une architecture décentralisée.

Un second point porte sur la simplification des BBAs. Nous avons, à ce sujet, proposé une méthode permettant de répartir la masse de l’élément supprimé sur un ensemble d’éléments focaux. Cette répartition, effectuée au moyen de la décomposition canonique, n’est possible que si la BBA est séparable. Cette condition étant vérifiée dans notre cas, nous n’avons pas poussé plus en avant cette analyse. Il serait toutefois intéressant d’étudier plus en détail le cas de BBAs non séparables.

Une troisième piste intéressante concerne les contraintes ergonomiques. À l’instar des a priori topographiques, d’autres informations pourraient aussi être intégrées au pistage sous forme de BBAs. En particulier, la régularisation des trajectoires sans compromettre la qualité du pistage est un point qui mériterait d’être étudié.

Une quatrième perspective, plus générale, concerne l’exploitation des BBAs. Dans ce travail, la BBA de localisation n’est utilisée que pour estimer la position des pistes et pour évaluer la croyance d’une association. Or, une analyse plus fine des structures de croyances peut contenir d’autres informations. Elle pourrait, par exemple mettre en évidence des contradictions au moins partielles entre les capteurs et peut-être ainsi diagnostiquer des dysfonctionnements. Cette analyse pourrait également être élargie aux fausses détections.

Enfin, la contrainte de discrétion est aussi un enjeu important. Cette discrétion imposerait une nouvelle stratégie de collaboration entre les nœuds. Ce n’est alors plus une fusion entre les pistes et les nouvelles détections qui doit être réalisée mais une fusion entre des pistes locales maintenues sur chaque nœud. Dans ce cas cependant, la gestion des incohérences peut devenir complexe.
Chapitre 10

Annexes

Sommaire

10.1 Détection d’intrusions par caméra visible . . . . . . . . . . . . . . 142
10.1.1 Objectifs et hypothèses de travail . . . . . . . . . . . . . . 142
10.1.2 Détection du changement . . . . . . . . . . . . . . . . . . . 142
10.1.3 Pistage . . . . . . . . . . . . . . . . . . . . . . . . . . . . 148
10.1.4 Organisation et fonctionnalités de l’application . . . . . . 151
10.2 Spécification des plans de déploiement . . . . . . . . . . . . . . 152
10.2.1 Format KML . . . . . . . . . . . . . . . . . . . . . . . . . . 152
10.2.2 Contenu d’un fichier de déploiement . . . . . . . . . . . . . 158
10.2.3 Données topographiques . . . . . . . . . . . . . . . . . . . . 160
10.1 Détection d’intrusions par caméra visible

Le démonstrateur SmartMesh sur lequel la fusion décentralisée a été testée est équipé de quatre types de capteurs différents. Parmi ceux-ci, les capteurs acoustiques, les capteurs passifs infrarouges et les caméras thermiques non refroidies ont été développés par des sociétés partenaires du projet. Les traitements mis en œuvre n’ont donc pas été rendu public et ces capteurs ont été considérés par la fusion comme des boîtes noires capables de détecter des intrusions et de communiquer selon une interface définie.

En revanche, la méthode de traitement des images visibles est connue puisque c’est nous qui l’avons développée en marge du module de fusion décentralisée. Bien que ce travail soit indépendant, nous avons choisi de le décrire brièvement ici afin d’illustrer les traitements en amont fournisant les détections traitées par la fusion.

10.1.1 Objectifs et hypothèses de travail

La méthode que nous allons décrire a pour objectif de détecter et suivre des objets mobiles à partir d’images provenant d’une caméra supposée fixe. Elle est en outre destinée à être exploitée dans un contexte de surveillance de site. Cette spécificité aura des conséquences à la fois sur les caractéristiques des images d’entrées et sur les objectifs de performance du système.

En effet, puisqu’il s’agit de détecter des intrusions, nous pouvons raisonnablement nous attendre à ce que celles-ci soient relativement rares. Le nombre d’objets à suivre simultanément sera donc peu élevé, ce qui simplifie considérablement le pistage. Cependant, pour que le système soit exploitable, il est impératif que le nombre de fausses alarmes soit extrêmement faible et que les intrusions soient efficacement détectées. La qualité de la détection sera donc privilégiée au pistage, ce dernier étant généralement plus critique lorsqu’il s’agit de suivre un grand nombre d’objets.

La suite de cette annexe sera divisée en trois parties. La première abordera le problème de la détection, la deuxième évoquera le pistage des objets détectés et enfin, la troisième partie détaillera très rapidement les différentes fonctionnalités de l’exécutable.

10.1.2 Détection du changement

De manière générale, toutes les techniques de détection développées et testées pour ce projet suivent la même démarche constituée des deux étapes suivantes :

– mise à jour d’une image de fond modélisant la scène statique, sans objet en mouvement,
– détection des différences entre l’image de fond et l’image courante.

Chacune des solutions testées pour ces deux étapes va être détaillée dans la suite de cette section mais notons déjà que ces traitements sont indépendants et que n’importe quel choix parmi les solutions proposées pour chaque étape constitue un système opérationnel.
Mise à jour de l'image de fond

Dans des conditions idéales de fonctionnement, l'image de fond, également appelée background, pourrait être constituée d'une image acquise au démarrage du système de détection. Si la luminosité est parfaitement stable et si aucun objet en mouvement ne vient se placer de manière permanente dans le champ de la caméra, cette image constituerait alors une référence tout à fait satisfaisante. Malheureusement, en pratique aucune de ces deux conditions n’est vérifiée et il est nécessaire d’assimiler progressivement les changements dans l’image de référence.

Filtre sigma-delta. Ce filtre est un classique dont le succès s’explique à la fois par son efficacité et par sa simplicité. Nous en proposons ici une version légèrement modifiée permettant de séparer les étapes de mise à jour et de détection du changement. Chaque pixel $i$ du background est alors modélisé par une moyenne $m_i$ et un bruit $s_i$ évoluant au cours du temps par assimilation des nouvelles observations $p_t^i$ selon la démarche qui suit.

- Mise à jour de $m_i$ :
  - si $m_i > p_t^i + \delta$ alors $m_i = m_i - \delta$,
  - si $m_i < p_t^i - \delta$ alors $m_i = m_i + \delta$,
  - sinon $m_i = p_t^i$.
- Mise à jour de $s_i$ :
  - si $s_i > |p_t^i - m_i|$ alors $s_i = s_i - 1$,
  - si $s_i < |p_t^i - m_i|$ alors $s_i = s_i + 1$,

où le paramètre $\delta$ exprime la réactivité du filtre face au changement.

Notons que si la moyenne $m_i$ reste constante au cours du temps, $s_i$ tendra vers la valeur médiane des différences absolues $|p_t^i - m_i|$. Par conséquent, si les observations $p_t^i$ se comportent comme un bruit blanc gaussien autour de la moyenne $m_i$, $s_i$ est lié à l’écart-type $\sigma_i$ de ce bruit par la relation :

\[
s_i \simeq 0.68\sigma_i \tag{10.1}
\]

Ce modèle, bien que très simple, est relativement robuste et ne dépend que du paramètre $\delta$ la plupart du temps fixé à 1. Il souffre toutefois d’un inconvénient majeur : afin d’être effectués rapidement, les calculs sont généralement réalisés en octet. Dans ces conditions, la plus petite valeur possible pour $\delta$ est 1, ce qui implique une assimilation très rapide du changement. Par exemple, si la caméra fonctionne à une fréquence de 25 Hz, un objet détecté qui s’arrête provisoirement de bouger sera assimilé dans l’image de fond (et donc plus détecté) après environ deux secondes.

Modèle codebook. Dans ce deuxième modèle, chaque pixel $i$ est modélisé par plusieurs triplets $(m_{ij}, s_{ij}, c_{ij})$ ainsi qu’un triplet $(m_i, s_i, c_i)$ pour l’apprentissage. La mise à jour à l’aide de l’observation $p_t^i$ est alors effectuée comme suit.

- Recherche du triplet $j$ dont la moyenne $m_{ij}$ minimise $|m_{ij} - p_t^i|$ :
  - si $|m_{ij} - p_t^i| < \gamma s_{ij}$, alors $m_{ij}$ et $s_{ij}$ sont mis à jour de manière identique à un filtre sigma-delta et $c_{ij} = t$,
– si $|m_i - p_i^t| < \gamma s_i$, alors mise à jour du triplet d’apprentissage comme dans un filtre sigma-delta et $c_i = c_i + 1$,
– sinon réinitialisation du triplet d’apprentissage : $m_i = p_i^t$, $s_i = s_{init}$ et $c_i = 1$.
– Parcours des triplets $(m_j^t, s_j^t, c_j^t)$. Si $c_j^t < t - dt$, le triplet $j$ est supprimé.
– Si $c_i > c_{min}$, le triplet d’apprentissage est ajouté au modèle.

Ce modèle nécessite alors trois paramètres :
– $\gamma$, le seuil de tolérance de la différence entre une observation $p_i^t$ et un triplet du modèle,
– $dt$ le délai maximum au delà duquel un triplet non observé est supprimé,
– $c_{min}$ le nombre minimum d’observations avant qu’un triplet en cours d’apprentissage ne soit ajouté au modèle.

Ce modèle permet de gérer des images dans lesquelles certaines zones ne peuvent être modélisées par une seule valeur. C’est par exemple souvent le cas de la végétation dont la couleur peut varier brusquement avec le vent ou la luminosité. Malheureusement ce traitement plus lourd a un coût en terme de temps de calcul.

**Modèle codebook simplifié.** Ce troisième modèle est une version simplifiée du modèle par codebook. L’idée consiste à séparer le modèle de background et les pixels en cours d’apprentissage. L’objectif de cette approche est de décorrélérer la réactivité $\delta$ et le délai d’assimilation. Chaque pixel $i$ est alors modélisé par un seul couple $(m_i, s_i)$ et un triplet d’apprentissage $(m_i^a, s_i^a, c_i)$. Le modèle est mis à jour de la même manière que dans le paragraphe précédent et la paire $(m_i, s_i)$ est remplacée par $(m_i^a, s_i^a)$ dès que $c_i^a > c_{min}$.

En raison du bon compromis qu’elle assure et suite aux différents essais réalisés, c’est cette technique qui a été utilisée pour SmartMesh.

**Détection d’anomalies.**

Cette deuxième étape concerne la détection des zones de changement entre l’image courante et le modèle de background.

**Changement au niveau pixel.** Puisque nous disposons d’un modèle du fond et d’une valeur courante pour chaque pixel, nous pouvons évaluer par seuillage le changement pour chaque pixel. Le bruit $s_i$ étant proportionnel à l’écart-type, les pixels de changement à l’instant $t$ sont ceux vérifiant l’inégalité :

$$|m_i - p_i^t| > ks_i$$

où $k$ est le paramètre fixant le seuil de détection.

Il faut toutefois noter que l’ensemble des pixels de l’image peut représenter plusieurs millions de tests. Si les observations se comportent comme des réalisations indépendantes de distributions normales de moyennes $m_i$ et d’écart-types $s_i$, il faut s’attendre à ce que certains pixels ne correspondant pas à du changement se trouvent à quatre ou cinq fois l’écart-type. Il est donc difficile de choisir un seuil $k$ permettant d’éviter les fausses alarmes tout en détectant le changement à cette échelle d’analyse.
En pratique, le taux de fausses détections est significativement amélioré en filtrant les zones de changement en fonction de leur taille. Ce traitement est réalisé en segmentant le changement en régions connexes et en supprimant les éléments trop petits. Il repose sur l’hypothèse que le changement concerne des objets ayant une taille importante par rapport à la résolution de l’image et que les fausses détections sont spatialement décorrélées.

Ce gain de robustesse se paie donc par une taille minimale des zones de changement détectables.

Changement par fenêtre. La seconde approche consiste à comparer l’image courante et le background sur des fenêtres de taille supérieure au pixel. La localisation des changements sera alors limitée par la taille de la fenêtre d’analyse mais contrairement au filtrage imposant une taille minimale aux détections, des zones de changement de taille inférieure aux fenêtres pourront être détectées. En effet, un changement radiométrique local mais significatif peut modifier la statistique de toute la fenêtre de sorte qu’il soit détecté.

Le modèle proposé est le suivant :

à chaque pas de temps, la valeur d’un pixel sans changement est une variable aléatoire $X$ qui suit une distribution normale $N_i(\mu_i, \sigma_i)$ de moyenne $\mu_i = m_i + \Delta$ et d’écart-type $\sigma_i \simeq s_i/0.68$ où $\Delta$ est le changement moyen observé sur toute l’image. La somme des différences quadratiques $\sum_{i \in W} (p^i - m_i)^2$ sur une fenêtre $W$ de $n$ pixels suivra alors une loi $\chi^2$.

En notant $Y_i$ la variable aléatoire de distribution normale centrée réduite telle que $X_i = \sigma_i Y_i + \mu_i$, la moyenne $\mu$ de la somme quadratique vaut alors :

$$\mu = E \left[ \sum_{i \in W} (\sigma_i Y_i + \Delta)^2 \right] = \sum_{i \in W} \left( \sigma_i^2 E[Y_i^2] + 2\Delta \sigma_i E[Y_i] + E[\Delta^2] \right) = n\Delta^2 + \sum_{i \in W} \sigma_i^2$$

et sa variance $\sigma^2$ :

$$\sigma^2 = E \left[ \left( \sum_{i \in W} (\sigma_i Y_i + \Delta)^2 \right)^2 \right] - \left( n\Delta^2 + \sum_{i \in W} \sigma_i^2 \right)^2$$

Si toutes les observations sont mutuellement indépendantes, alors

$$\sigma^2 = \sum_{i,j \in W} \sigma_i^2 \sigma_j^2 E[Y_i^2 Y_j^2] + 4\Delta^2 \sum_{i,j \in W} \sigma_i \sigma_j E[Y_i Y_j]$$

(10.5)
\(E[Y_i^2 Y_j^2] = 1,\ E[Y_i^2 Y_j] = 0,\ E[Y_i Y_j] = 0\ \ \ \forall i, j \in W | i \neq j\) 
\(E[Y_i^2 Y_j^2] = 3,\ E[Y_i^2 Y_j] = 0,\ E[Y_i Y_j] = 0\ \ \ \forall i, j \in W | i = j\)  
(10.6)  
(10.7)

Par conséquent, la variance devient :

\[
\sigma^2 = 2 \sum_{i \in W} \sigma_i^4 + 4\Delta^2 \sum_{i \in W} \sigma_i^2
\]

(10.8)

Si en revanche, les observations d’une fenêtre sont toutes mutuellement corrélées, alors

\(E[Y_i^2 Y_j^2] = 3,\ E[Y_i^2 Y_j] = 0,\ E[Y_i Y_j] = 0\ \ \ \forall i, j \in W\)

(10.9)

et par conséquent :

\[
\sigma^2 = 2 \left( \sum_{i \in W} \sigma_i^2 \right)^2 + 4\Delta^2 \left( \sum_{i \in W} \sigma_i \right)^2
\]

(10.10)

Pour essayer d’évaluer lequel de ces deux modèles correspond le mieux à la réalité, nous avons calculé la variance de \(\sum_{i \in W} (p_i - m_i)^2\) sur des séquences d’images sans changement et pour différentes tailles de fenêtres. Ces variances, calculées en faisant l’hypothèse d’ergodicité, ont ensuite été comparées aux variances estimées selon les modèles avec et sans corrélations. Les résultats présentés sur la figure 10.1 tendent à montrer que le modèle faisant l’hypothèse d’une corrélation totale approche mieux la variance effective malgré une légère surestimation.

Fig. 10.1: Comparaison de la variance (bleu) sur des fenêtres de tailles croissantes avec le modèle sous hypothèse de corrélation totale (jaune) et indépendance des pixels (rouge).
Si la taille \( n \) des fenêtres est suffisamment grande, la somme quadratique tend rapidement vers une loi normale. Les fenêtres de changement peuvent alors être détectées par un seuillage adaptatif de la forme :

\[
\sum_{i \in W} (p_t^i - m_i)^2 > \mu + k\sigma
\]  
(10.11)

En injectant les moyennes et variances estimées plus haut, cette inégalité devient :

\[
\sum_{i \in W} (p_t^i - m_i)^2 > n\Delta^2 + \sum_{i \in W} \sigma_i^2 + k\sqrt{2 \left( \sum_{i \in W} \sigma_i^2 \right)^2 + 4\Delta^2 \left( \sum_{i \in W} \sigma_i \right)^2}
\]  
(10.12)

ou encore, avec l’approximation \( \sigma_i^2 \approx 2s_i^2 \) :

\[
\sum_{i \in W} (p_t^i - m_i)^2 > n\Delta^2 + 2 \sum_{i \in W} s_i^2 + k\sqrt{8} \sqrt{\left( \sum_{i \in W} s_i^2 \right)^2 + \Delta^2 \left( \sum_{i \in W} s_i \right)^2}
\]  
(10.13)

Afin d’éviter le calcul des \( \sum_{i \in W} s_i \), nous pouvons encore faire l’approximation suivante :

\[
\sum_{i \in W} (p_t^i - m_i)^2 > n\Delta^2 + 2 \sum_{i \in W} s_i^2 + k\sqrt{8} \sqrt{\sum_{i \in W} s_i^2 + \left( n\Delta^2 + \sum_{i \in W} s_i^2 \right)}
\]  
(10.14)

**Image intégrale.** Afin d’affiner l’estimation de la taille et de la position des zones de changement, la détection proposée dans la section précédente peut être réalisée sur des fenêtres se recouvrant partiellement. Dans ce cas, il n’est pas nécessaire de calculer les sommes \( \sum_{i \in W} \) et \( \sum_{i \in W} (p_t^i - m_i)^2 \) sur chaque fenêtre. Ces calculs peuvent en effet être effectués de manière efficace à l’aide des images intégrales \( I_{s_2} \) et \( I_{(p_t^i - m_i)^2} \). Dans ce cas, les sommes sur une fenêtre délimitée par \( W = (x_{\min}, y_{\min}, x_{\max}, y_{\max}) \) sont calculées comme suit :

\[
\sum_{i \in W} A = I_A(x_{\min}, y_{\min}) + I_A(x_{\max}, y_{\max}) - I_A(x_{\min}, y_{\max}) - I_A(x_{\max}, y_{\min})
\]  
(10.15)

Une fois les images intégrales calculées, cette approche permet donc de tester rapidement un grand nombre de fenêtres qui se chevauchent. En segmentant le résultat en régions connexes, la taille et la localisation des détections seront alors évaluées de manière plus précises.

C’est cette approche qui a été retenue pour la détection dans les images de *SmartMesh*.

**Seuillage par hystérèsis.** Dans les trois méthodes présentées ci-dessus, le détection est effectuée par un seuillage sur les pixels ou sur des fenêtres d’évaluation. Par soucis de clarté, le seuil \( k \) a été considéré comme constant jusqu’ici mais la méthode implantée permet en réalité de procéder par seuillage à hystérèsis afin d’affiner la détection.
10.1.3 Pistage

Le pistage des détections consiste à estimer au cours du temps le déplacement des objets détectés. Dans le contexte de la surveillance, le pistage sert avant tout à filtrer certaines détections qui peuvent être considérées comme des fausses alarmes du point de vue de l’utilisateur. Par exemple, des objets bougeant sous l’action du vent reflètent bien un changement dans l’image. Il est donc normal qu’ils soient détectés comme tels bien qu’il ne s’agisse pas pour autant d’intrusions. Le pistage constitue alors un moyen simple et efficace de filtrer ces fausses alarmes en ne communiquant à l’utilisateur que les pistes s’étant déplacées dans l’image d’une distance minimum.

Le pistage permet également d’estimer la vitesse et la trajectoire des cibles ainsi que leur position en cas de masquage temporaire.

Association entre les détections et les pistes

Le pistage développé pour le projet SmartMesh n’exploite pas l’information contenue dans l’image. Il se contente d’associer les détections et les pistes uniquement sur un critère de proximité spatiale. Cette approche est un peu simpliste mais elle présente l’avantage d’être extrêmement simple et rapide.

Recherche du plus proche voisin. Cette association consiste à associer chaque piste à la détection la plus proche. Si aucune détection ne se trouve à une distance inférieure à une limite, la piste n’est alors pas associée.

Cette démarche très simple mérite tout de même deux remarques :
– plusieurs pistes peuvent être associées à la même détection,
– la distance considérée est la distance euclidienne entre les centres des boîtes englobantes.

Cette approche s’est avérée peu efficace car il arrive fréquemment qu’une cible soit détectée en plusieurs morceaux. Dans ce cas, s’il n’y a qu’un seul de ces morceaux qui est associé à la piste et les autres détections risquent d’initialiser des pistes parasites.

Recherche de tous les voisins proches. Il est possible d’associer chaque piste à toutes les détections se trouvant dans son voisinage. S’il y a beaucoup de détections et de pistes, cette approche risque d’associer chaque détection à plusieurs pistes, mais dans un contexte de surveillance de site, où les détections sont éparse, les résultats sont acceptables.

Pour cette approche, il est préférable de considérer la plus courte distance séparant les bords des boîtes englobantes. Ainsi, une piste sera au moins associée à toutes les détections ayant une intersection non nulle avec celle-ci.

Cette technique permet d’éviter l’initialisation de pistes parasites, mais si deux pistes sont trop proches l’une de l’autre, elles seront toutes les deux associées aux deux détections courantes. La mise à jour des pistes sera alors dégradée.
Minimisation de la distance totale. Cette troisième approche cherche à minimiser la distance totale séparant les pistes et les détections. Après avoir construit la matrice des distances séparant chaque piste de chaque détection, le problème peut être résolu par l'algorithme hongrois. Avec cette technique, chaque détection et chaque piste sont au plus associées à un seul élément. La critique concernant la recherche du plus proche voisin reste donc valable. Cependant, en observant après traitement la matrice des coûts, on peut associer les éléments isolés au voisin le plus proche, si celui-ci est situé à une distance inférieure au seuil. Nous pouvons ainsi favoriser les associations 1-1 tout en garantissant que chaque élément ayant un voisin proche sera associé.

Il faut encore définir la distance entre les pistes et les détections. L'expérience nous a montré que la distance minimale entre bords de boîtes était souvent un bon critère, mais il arrive que cette distance soit nulle pour plusieurs voisins. C'est alors la distance entre les centres qui est préférable. Pour combiner ces deux cas de figure, le coût d'association $c_{ij}$ utilisé est construit comme suit :

$$c_{ij} = d_e(i,j) + (1 + d_{c}^{\text{max}})d_{e}(i,j)$$  \hspace{1cm} (10.16)

où
- $d_e(i,j)$ est la distance entre le centre de la détection $i$ et le centre de la piste $j$,
- $d_{c}^{\text{max}}$ est la plus grande distance entre centres,
- $d_e(i,j)$ est la distance entre les bords.

Mise à jour des pistes

Une fois l'association effectuée, l'état des pistes peut être mis à jour à l'aide des détections. Ce traitement concerne deux grandes catégories de variables constituant l'état d'une piste. Il s'agit d'une part des variables décrivant la piste telle qu'elle apparaît à l'image (position, boîte englobante, ...) et d'autre part des variables décrivant les caractéristiques physiques de la piste telles que sa hauteur, sa distance à la caméra ou l'estimation de sa vitesse de déplacement.

Traitement des associations  Lorsqu'une piste est associée à une seule détection, les paramètres de la piste sont simplement mis à jour en remplaçant les anciennes valeurs de position et distance par la nouvelle détection. Un filtrage de type Kalman aurait pu être introduit, mais ce développement n'a pas été une priorité pour ce projet.

Lorsque une piste n'est associée à aucune détection, la piste n'est pas détruite immédiatement, mais un compteur est initialisé. Après $n$ non détections, la piste est détruite. Si au contraire, une piste est associée à plusieurs détections, la nouvelle boîte englobante de la piste sera la boîte qui englobe toutes les détections. À nouveau, un compteur est initialisé, et après $n$ regroupements, la piste est séparée en plusieurs morceaux. Enfin, si plusieurs pistes sont associées à une seule détection, la détection sera partagée par toutes les pistes et un compteur également initialisé. Après $n$ associations consécutives, les pistes sont regroupées.
Estimation des grandeurs physiques  La taille et la localisation des pistes est estimée en faisant l’hypothèse que les pistes touchent le sol et que ce dernier est plan. Nous supposons également que l’horizon est parallèle aux lignes de l’image et que la caméra peut être décrite par un modèle pinhole (figure 10.2). Dans ce cas, les coordonnées polaires \((\alpha, d)\) d’un point \((x, y)\) de l’image sont données par :

\[
\alpha = \atan \left( \frac{x - x_c}{f} \right) \tag{10.17}
\]

\[
d = h \cotan \left( \atan \left( \frac{y - y_c}{f} \right) + \delta \right) \tag{10.18}
\]

où
- \(f\) est la distance focale en pixel,
- \(h\) est la hauteur de la caméra,
- \(\delta\) est l’angle de tangage.

Fig. 10.2: Estimation de l’orientation et de la distance d’une cible

De même, la hauteur \(h_e\) et la largeur \(w_e\) de la cible peuvent être approchées par :

\[
h_e = h - d \tan \left( \atan \left( \frac{y_h - y_c}{f} \right) + \delta \right) \tag{10.19}
\]

\[
w_e = d \frac{\Delta x}{f} \tag{10.20}
\]

où
- \(d, \alpha\) sont les coordonnées calculées précédemment,
- \(y_h\) est la position du haut de la détection,
- \(\Delta x\) est la largeur de la détection.

L’évaluation de ces paramètres suppose que la hauteur \(h\) et l’angle de tangage \(\delta\) de la caméra soient connus. Cette hauteur peut être facilement estimée lors de l’installation, mais il n’en va pas de même pour le tangage. C’est pour cette raison qu’une méthode d’estimation automatique a été mise en place. La technique développée propose de passer par l’estimation de la hauteur de l’horizon dans l’image. Si cette hauteur \(h_z\) est connue, l’angle de tangage peut en effet être facilement déduit par la relation :

\[
\delta = \atan \left( \frac{h_z - y_c}{f} \right) \tag{10.21}
\]
La hauteur de l’horizon est alors estimée en observant des cibles de taille constante qui s’éloignent de la caméra comme illustré sur la figure 10.3. Les détections de ces cibles vont se rapprocher progressivement de la hauteur de l’horizon et leurs hauteurs vont également tendre vers zéro. Plus précisément, les coordonnées verticales $y_i$ et les tailles $h_i$ des détections sont liées par une relation linéaire du type :

$$y_i = ah_i + b$$  \hspace{1cm} (10.22)

où le paramètre $b$ correspond à la position lorsque la taille est nulle. Il s’agit donc de la hauteur de l’horizon $h_z$.

La hauteur de l’horizon peut donc être estimée en cherchant l’intersection de la droite approchant les positions des détections d’une cible en fonction de la hauteur. Celle-ci se calcule par un estimateur de moindres carrés :

$$h_z = \frac{1}{n} \sum_i y_i - \frac{S_{by}}{nS_{hh}} \sum_i h_i$$ \hspace{1cm} (10.23)

avec :

$$S_{by} = \sum_i y_i h_i - \frac{1}{n} \sum_i y_i \sum_i h_i$$ \hspace{1cm} (10.24)

$$S_{hh} = \sum_i h_i^2 - \frac{1}{n} \left( \sum_i h_i \right)^2$$ \hspace{1cm} (10.25)

10.1.4 Organisation et fonctionnalités de l’application

L’application développée pour le projet SmartMesh est organisée de la manière suivante : une interface charge une nouvelle image à cadence régulière et la place dans un buffer circulaire. L’algorithme de détection copie localement l’image la plus récente disponible et la traite. Si toutes les images disponibles ont été traitées, il attend jusqu’à ce qu’une nouvelle image soit placée dans le buffer. Il en va de même pour le suivi des pistes et pour chacune des sorties produites. Actuellement, deux interfaces d’entrées, à spécifier lors de la compilation, sont disponibles : un lecteur...
de fichier vidéo et une interface avec la caméra achetée pour le projet. Plusieurs sorties paramétrables dans le fichier de configuration peuvent être produites : un affichage de la vidéo avec différentes surcharges, un enregistrement de vidéo et un enregistrement d’images. Les enregistrements peuvent commencer lors de la première acquisition d’images ou démarrer et s’arrêter automatiquement lors de détections d’intrusions. Dans ce cas, il est également possible de spécifier une avance de phase pour que l’enregistrement commence légèrement avant le déclenchement de l’alarme. Enfin, deux fréquences de travail peuvent être spécifiées : une fréquence de scruitation ne monopolisant pas toutes les ressources lorsqu’aucune anomalie n’est détectée et une fréquence plus élevée se déclenchant en cas de détection pour tenter de suivre la piste et remonter une alarme le cas échéant.

10.2 Spécification des plans de déploiement

Lors de la phase d’initialisation du système *SmartMesh*, chaque agent de fusion doit récupérer un certain nombre d’informations sur la configuration du déploiement. Ces données étant partagées avec le pupitre de supervision, il a été nécessaire de spécifier le format et le contenu des données.

10.2.1 Format KML

Par souci de standardisation, nous avons proposé de décrire les données en *KML*. Ce langage, développé par l’*OGC*\(^1\), utilise le formalisme *XML* et est destiné au partage de données géo-spatiales.

Toutes les balises spécifiées par ce format ne sont pas utiles pour le projet *SmartMesh*. De plus, certaines informations n’ont d’intérêt que pour le module de fusion ou le poste de supervision. Dans cette section, nous présenterons uniquement les éléments exploités par la fusion. Si d’autres éléments sont rencontrés, le contenu se trouvant entre la balise ouvrante et la balise fermante est ignoré, même si certaines parties sont compréhensibles par la fusion.

L’ensemble des classes reconnues sont présentées sur la figure 10.4, les éléments encadrés étant des classes abstraites. Aucun élément de ces classes encadrées ne peut donc être instancié directement dans un fichier *KML*.

Détailons à présent chacune de ces classes.

Object

Il s’agit d’une classe abstraite. Cette classe regroupe les éléments *KML identifiables*. C’est-à-dire les éléments pour lesquels il est possible de spécifier un attribut *id* qui permet de les identifier. Cet attribut peut être utilisé pour mettre à jour les éléments en rechargeant un fichier.

\(^1\) *Open GIS Consortium*
Un Document un est conteneur regroupant des StyleSelector, des Feature et des Folder. La classe StyleSelector, servant à décrire des styles de présentation, n’a pas d’intérêt pour la fusion et ne sera pas abordée dans ce document. Les deux autres types d’éléments seront décrits plus loin.

Exemple

```xml
<?xml version= "1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    ...
  </Document>
</kml>
```

Folder

Les éléments Folder servent à organiser la hiérarchie des Feature.

Exemple

```xml
<?xml version= "1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Folder>
      ...
    </Folder>
  </Document>
</kml>
```
**Feature**

Il s’agit d’une classe abstraite. Un *Feature* regroupe un certain nombre d’éléments spécifiant des propriétés relatives à l’objet dérivé (nom, auteur, format, couleur,...).

**Éléments spécifiques :**
- `<ExtendedData>` Permet d’ajouter des propriétés personnalisées aux objets. Ces données peuvent être typées ou non.
- `<name>` Nom de l’objet. Il ne s’agit pas d’un identifiant, mais d’un nom à l’usage des utilisateurs, pouvant par exemple apparaître à l’affichage.
- `<description>` Description textuelle de l’objet.

**Placemark**

Un *Placemark* est un *Feature* contenant 0 ou 1 élément de type *Geometry*. Tous les objets représentables sur une carte sont décrits par un élément de ce type.

**Exemple**

```xml
<Placemark>
  <name> Point </name>
  <description> Une description du point </description>
  <Point>
    <coordinates> 90.86948943473118,48.25450093195546,0 </coordinates>
  </Point>
</Placemark>
```

**ExtendedData**

Élément de données personnalisées d’un *Feature*. Cet élément permet d’ajouter des éléments non spécifiés par la norme *KML*.

**Exemple**

```xml
<Placemark>
  <name>Objet personnalisé</name>
  <ExtendedData>
    ...
  </ExtendedData>
</Placemark>
```

**Geometry**

Il s’agit d’une classe abstraite, aucun élément de ce type de peut être instancié directement dans un fichier *KML*. Sans élément spécifique, cette classe a pour seule fonction de désigner les objets géométriques au moyen d’un seul nom.
Point

Localisation géographique définie par une latitude, une longitude et une altitude (optionnelle).

Éléments spécifiques
- `<extrude>` spécifie si le point est relié au sol par une droite,
- `<altitudeMode>` spécifie la manière dont l’altitude doit être interprétée :
  - `clampToGround` (par défaut) l’altitude est ignorée,
  - `relativeToGround` l’altitude désigne la hauteur du point par rapport au sol,
  - `absolute` l’altitude désigne la hauteur du point par rapport au niveau de la mer.
- `<coordinates>` (nécessaire) un seul n-uplet de flottants décrivant la localisation au moyen de la longitude, latitude et altitude séparé par des virgules et sans espace.

Exemple

```
<Placemark>
  <name> Point </name>
  <description> Une description du point </description>
  <Point>
    <coordinates> 90.86948943473118,48.25450093195546,0 </coordinates>
  </Point>
</Placemark>
```

LinearRing

Ensemble de segments de droite connectés et fermés. Typiquement, cette classe sert à décrire les frontières d’un polygone.

Éléments spécifiques
- `<extrude>` spécifie si la ligne est extrudée jusqu’au sol, formant alors une clôture fermée,
- `<altitudeMode>` spécifie la manière dont l’altitude doit être interprétée :
  - `clampToGround` (par défaut) l’altitude est ignorée,
  - `relativeToGround` l’altitude désigne la hauteur du point par rapport au sol,
  - `absolute` l’altitude désigne la hauteur du point par rapport au niveau de la mer.
- `<coordinates>` (nécessaire) quatre n-uplets ou plus. Chacun des n-uplets est un triplet de flottants décrivant la localisation au moyen de la longitude, latitude et altitude (en option) et séparé par des virgules. La dernière coordonnée doit être identique à la première.
Exemple

<table>
<thead>
<tr>
<th>LinearRing</th>
</tr>
</thead>
</table>
| <coordinates>
| -122.365662,37.826988,10
| -122.365202,37.826302,10
| -122.364581,37.82655,10
| -122.365038,37.827237,10
| -122.365662,37.826988,10 |
| </coordinates>
| </LinearRing>

LineString

Ensemble de segments de droite connectés.

Éléments spécifiques

- `<extrude>` spécifie si la ligne est extrudée jusqu’au sol, formant alors une clôture.
- `<altitudeMode>` spécifie la manière dont l’altitude doit être interprétée :
  - `clampToGround` (par défaut) l’altitude est ignorée,
  - `relativeToGround` l’altitude désigne la hauteur du point par rapport au sol,
  - `absolute` l’altitude désigne la hauteur du point par rapport au niveau de la mer.
- `<coordinates>` (nécessaire) deux n-uplets ou plus. Chacun des n-uplets est un triplet de flottants décrivant la localisation au moyen de la longitude, latitude et altitude (option) séparé par des virgules.

Exemple

<table>
<thead>
<tr>
<th>LineString</th>
</tr>
</thead>
</table>
| <extrude>1</extrude>
| <coordinates>
| -122.365662,37.826988,10
| -122.365202,37.826302,10
| -122.364581,37.82655,10 |
| </coordinates>
| </LineString>

MultiGeometry

Regroupement de plusieurs éléments de type Geometry, le nombre d’éléments pouvant être égal à 0.
Polygone

Un Polygone est défini par une frontière extérieure et par une ou plusieurs frontières intérieures. Chacune de ces frontières est constituée d’un LinearRing. Les coordonnées du polygone doivent être spécifiées dans le sens horloger.

Éléments spécifiques

- `<extrude>` spécifie si le polygone est extrudée jusqu’au sol, formant alors un volume,
- `<altitudeMode>` spécifie la manière dont l’altitude doit être interprétée :
  - `clampToGround` (par défaut) l’altitude est ignorée,
  - `relativeToGround` l’altitude désigne la hauteur du point par rapport au sol,
  - `absolute` l’altitude désigne la hauteur du point par rapport au niveau de la mer.
- `<outerBoundaryIs>` (nécessaire) frontière extérieure du polygone, contenant un LinearRing
- `<innerBoundaryIs>` frontière intérieure du polygone, contenant un LinearRing.

Un Polygone peut contenir plusieurs `innerBoundaryIs` créant plusieurs *trous* dans le polygone.

Exemple

```xml
<Polygon id="ID">
  <extrude>0</extrude>
  <altitudeMode>clampToGround</altitudeMode>
  <outerBoundaryIs>
    <LinearRing>
      <coordinates>...</coordinates>
    </LinearRing>
  </outerBoundaryIs>
  <innerBoundaryIs>
    <LinearRing>
      <coordinates>...</coordinates>
    </LinearRing>
  </innerBoundaryIs>
</Polygon>
```
10.2.2 Contenu d’un fichier de déploiement

La fusion doit disposer de la localisation, de l’adresse IP et de l’identifiant de chaque nœud du système. De plus, pour chacun des capteurs, la fusion doit également connaître l’identifiant, le type du capteur ainsi que le nœud propriétaire. Deux capteurs peuvent éventuellement avoir le même identifiant à condition qu’ils ne soient pas situés sur le même nœud. Par ailleurs, même si le format KML accepte n’importe quelle chaîne de caractère comme identifiant, seul les nombres en fin de chaîne seront gardés. Par exemple, une caméra d’identifiant KML "Camera1" aura comme identifiant "1". Cet identifiant sera, entre autre, inclus à toutes les détections provenant du capteur.

La plupart des informations relatives aux capteurs dont la fusion à besoin ne sont pas directement spécifiées par KML. Ces données peuvent alors être insérées à l’aide du mécanisme d’ajout de données personnalisables.

Fig. 10.5: Diagramme des éléments de description du système de surveillance

Node

Un nœud du réseau peut être partiellement décrit par un Placemark dont la géométrie est de type Point et dont l’identifiant est spécifié. Le seul paramètre manquant est alors l’adresse IP.

Élément spécifique

- `<sh:ipaddress>` (nécessaire) chaîne de caractères définissant l’adresse IP du nœud.

Exemple

```
<Placemark id="1">
  <name>Node</name>
  <Point>
    <coordinates>2.166361,48.698235</coordinates>
  </Point>
  <ExtendedData xmlns:prefix="sh">
    <sh:Node>
      <sh:ipaddress>129.175.0.0</sh:ipaddress>
    </sh:Node>
  </ExtendedData>
</Placemark>
```
Sensor

Il s’agit d’une classe abstraite. Sensor regroupe tous les capteurs qu’un nœud peut gérer. La géométrie des capteurs peut comprendre un point, si ce dernier est déporté par rapport au nœud. Elle peut également contenir un polygone définissant la représentation graphique de la zone d’inspection du capteur. Les deux éléments peuvent être ajoutés au moyen d’un MultiGeometry.

Élément spécifique
– <sh:nodeid> (nécessaire) identifiant du nœud d’appartenance.

Camera

Cet élément regroupe les caméras visibles et IR.

Éléments spécifiques
– <sh:cameratype> (nécessaire) type de caméra (visible par défaut) :
  – visible
  – ir
– <sh:orientation> (nécessaire) orientation du capteur par rapport au nord.
  L’orientation correspond au milieu du secteur surveillé,
– <sh:aperture> (nécessaire) ouverture angulaire du champ de la caméra,
– <sh:range> (nécessaire) portée du capteur.

Exemple

```xml
<Placemark id="2">
  <name>Camera</name>
  <Point>
    <coordinates>2.166361,48.698235</coordinates>
  </Point>
  <ExtendedData xmlns:prefix="sh">
    <sh:Camera>
      <sh:nodeid>1</sh:nodeid>
      <sh:cameratype>ir</sh:cameratype>
      <sh:orientation>80</sh:orientation>
      <sh:aperture>70</sh:aperture>
      <sh:range>100</sh:range>
    </sh:Camera>
  </ExtendedData>
</Placemark>
```

Micro

Élément définissant un capteur acoustique.

Élément spécifique
– <sh:range> (nécessaire) portée du capteur.
### Exemple

```xml
<Placemark id="2">
  <name>Sensor</name>
  <Point>
    <coordinates>2.166361,48.698235</coordinates>
  </Point>
  <ExtendedData xmlns:prefix="sh">
    <sh:Micro>
      <sh:nodeid>1</sh:nodeid>
      <sh:range>100</sh:range>
    </sh:Micro>
  </ExtendedData>
</Placemark>
```

### Pir

Élément définissant un PIR.

**Éléments spécifiques**

- `<sh:orientation>` (nécessaire) orientation du capteur par rapport au nord. L’orientation correspond au milieu du secteur surveillé,
- `<sh:aperture>` (nécessaire) ouverture angulaire du champ du capteur,
- `<sh:range>` (nécessaire) portée du capteur.

### Exemple

```xml
<Placemark id="2">
  <name>Sensor</name>
  <Point>
    <coordinates>2.166361,48.698235</coordinates>
  </Point>
  <ExtendedData xmlns:prefix="sh">
    <sh:Pir>
      <sh:nodeid>1</sh:nodeid>
      <sh:orientation>80</sh:orientation>
      <sh:aperture>70</sh:aperture>
      <sh:range>100</sh:range>
    </sh:Pir>
  </ExtendedData>
</Placemark>
```

### 10.2.3 Données topographiques

Pour pouvoir être exploitées, les données topographiques doivent contenir une forme géométrique ainsi qu’un type spécifiant ce que représente la forme. Enfin, un identifiant unique est associé à chaque objet afin de pouvoir suivre d’éventuelles mises à jour.
Spécification des plans de déploiement

**Topographic Object**

Un *Topographic Object* est une classe abstraite désignant tous les éléments topographiques du plan de déploiement.

**Road**

Une *Road* décrit une zone favorable à la localisation et au déplacement des détections. Le terme favorable signifie qu’en cas de doute, une détection sera prioritairement placée sur une zone de type route. Une route n’affecte en rien la visibilité des capteurs. Un ou plusieurs types de détections peuvent également être associés à une route. Dans ce cas, seules les localisations de ces types seront affectées par la présence de la route. La géométrie d’une route peut être décrite par un polygone ou une polyligne associée à une largeur. De plus, la route sera toujours positionnée sur le sol, quelles que soient les valeurs de l’altitude et de *altitudeMode*.

**Éléments spécifiques**

- `<sh:width>` largeur de la route,
- `<sh:detectiontype>` type des détections affecté par la route :
  - vehicle
  - pedestrian
  - all

**Exemple**

```xml
<Placemark id="3">
  <name>Road</name>
  <LineString>
    <coordinates>...</coordinates>
  </LineString>
  <ExtendedData xmlns:prefix="sh">
    <sh:Road>
      <sh:detectiontype>vehicle</sh:detectiontype>
      <sh:type>road</sh:type>
      <sh:width>5</sh:width>
    </sh:Road>
  </ExtendedData>
</Placemark>
```
Annexes

Fence

Un Fence est à la fois une zone de masquage des capteurs et un obstacle empêchant le passage des cibles. Ces deux caractéristiques, associées à des paramètres de transparence et de perméabilité permettent de modéliser des murs, barrières et diverses clôtures. La géométrie d’un Fence est décrite par une polyligne associée à une hauteur optionnelle. La hauteur ne sera prise en compte que si altitudeMode est de type relativeToGround (la valeur absolute est ignorée).

Éléments spécifiques

- `<sh:permeability>` perméabilité des parois (1 par défaut). Valeur comprise entre 0 et 10. La valeur 0 signifie qu’il est impossible qu’une détection traverse l’obstacle et 10 signifie que l’obstacle ne constitue en rien une barrière physique,
- `<sh:transparency>` transparence des murs (0 par défaut). La valeur 0 signifie que l’obstacle est opaque et la valeur 10 qu’il est transparent.

Exemple

```xml
<Placemark id="3">
  <name>Fence</name>
  <LineString>
    <coordinates>...</coordinates>
  </LineString>
  <ExtendedData xmlns:prefix="sh">
    <sh:Fence>
      <sh:permeability>0</sh:permeability>
    </sh:Fence>
  </ExtendedData>
</Placemark>
```

Building

Un Building décrit un bâtiment. Tout comme les Fence, les bâtiments sont caractérisés par un paramètre de transparence et un paramètre de perméabilité. En revanche, la géométrie des Buildings est décrite par un polygone associé à une valeur d’extrusion optionnelle. La valeur d’extrusion ne sera prise en compte que si altitudeMode soit de type relativeToGround (la valeur absolute est ignorée).

Éléments spécifiques

- `<sh:permeability>` perméabilité du bâtiment (1 par défaut). Valeur comprise entre 0 et 10. La valeur 0 signifie qu’il est impossible qu’une détection traverse l’obstacle et 10 signifie que l’obstacle ne constitue en rien une barrière physique,
- `<sh:transparency>` transparence du bâtiment (0 par défaut). Valeur comprise entre 0 et 10. La valeur 0 signifie que l’obstacle est opaque et la valeur 10 qu’il est transparent.
Exemple

<Placemark id="3">
  <name>Building</name>
  <Polygon>
    <outerBoundaryIs>
      <LinearRing>...</LinearRing>
    </outerBoundaryIs>
    <Polygon>
      <ExtendedData xmlns:prefix="sh">
        <sh:Building>
          <sh:permeability>0</sh:permeability>
        </sh:Building>
      </ExtendedData>
    </Polygon>
  </Polygon>
</Placemark>
Bibliographie


