UNIVERSITE DE NICE-SOPHIA ANTIPOLIS - UFR SCIENCES
Ecole doctorale Sciences et Technologies
de l'Information et de la Communication

THÈSE

présentée pour l'obtention du titre de

DOCTEUR EN SCIENCE
Specialité: Informatique

par

Nathanaël ROTA

CONTRIBUTION A LA RECONNAISSANCE DE COMPORTEMENTS HUMAINS A PARTIR DE SEQUENCES VIDEOS

Soutenue le 30 octobre 2001 devant le jury composé de:

Jean Paul RIGAULT	Professeur, Université de Nice-Sophia Antipolis	Président
Marie-Odile CORDIER	Professeur, Université de Rennes	Rapporteur
Malik GHALLAB	Directeur de Recherche, LAAS	Rapporteur
Hilary BUXTON	Professeur, University of Sussex	Examinateur
Catherine TESSIER	Ingenieur de Recherche, ONERA-CERT	Examinateur
Jean-Philippe BLANCHARD	CNCA, Responsable de la veille technologique	Examinateur
Monique THONNAT	Directeur de Recherche, INRIA	Directeur

Je tiens à remercier ici,
Malik Ghallab et Marie Odile Cordier d'avoir accepté malgré leur emploi du temps chargé, d'être rapporteurs de ce travail.
Hilary Buxton, Catherine Tessier et Jean Phillipe Blanchard d'avoir accepter de participer du jury;
Jean Paul Rigault de l'avoir presider.
Monique Thonnat qui m'a acceuilli dans son equipe et m'a toujours laissé une grande liberté pour le choix et l'organisation de mon travail.
Agnès Cortell pour avoir gerer administrativement ces travaux.
Robert Stahr, Patrick Nivet, Francois Bremond pour leurs contributions à ces travaux, ainsi que les interminables discussions scientifiques.
tous les membres de l'equipe ORION pour leur soutient amical et l'ambiance agréable à laquelle ils ont contribué (specialement Alain Boucher et Nicola Dey).
Je tiens à remercier aussi tous ceux qui m'ont supporté,
Christophe «pilote de 33» Samson, Frank «Herr Doctor Gangrainisator» Da, Laurent «lolo» Berthelot, Emmanuel «Goat Boy» Laborde, Guillaume «salviac» Rellier, Guillaume Bourdeau, Guillaume Guillard, Fabienne Terrer, Stephane Amami, Caroline Lacoste, Stephane Aboab, Maria del Mar Marcos Lopez, Monica Crubezy, Patrick Itey, Robert, Brahim, Marielaure, Radu, Oscar, Nordine, Ali, Giannella, Audrey, Delphine, Anne and co.
ma famille ...

Table des matières

Table des figures

Liste des tableaux

Notations

- A: $\bar{A}_t$ est l'ensemble des arcs représentant les relations binaires entre les concepts de la scène $\bar{A}_t = \bar{T}_t \cup \bar{R}_t$
- B: B_t est l'ensemble des blobs à t.
- C: C est le problème de groupement des blobs : $Q_t = C(B_t)$.
- D: D est le problème de reconnaissance de comportements : $(V_{t+1},R_{t+1}) = D(K,\tilde{G}_{t+1})$
- F: $\bar{F}_t$ est l'ensemble des sommets représentant les concepts datés de la scène. $\bar{F}_t = \bar{O}_t \cup \bar{P}_t \cup \bar{V}_t$
- G: $\bar{G}_t$ est le graphe d'interprétation à t. $\bar{G}_t = (\bar{F}_t,\bar{A}_t)$
- H: $\mathcal{H}_t$ est l'ensemble des humains physiques présents dans la scène à l'instant t.
- I: I le problème de diagnostic de correspondance temporelle. $\mathcal{I}$ un diagnostic de correspondance temporelle.
- K: K est l'ensemble des modèles de comportements. $K = \{M_1,...,M_m\}$.
- M: M_i est le i^{me} modèle de K. $\mathcal{M}$ est le problème de mise en correspondance. $M(\bar{G}_{t-1},Q_t) = (P_t,T_t)$,
- O: $\bar{O}_t$ est l'ensemble des sommets représentant les objets de la scène a l'instant t.
- P: $\bar{P}_t$ est l'ensemble des sommets représentant les personnes jusqu'à l'instant t
- Q: Q_t est une partition de l'ensemble des blobs à t.
- R: $\bar{R}_t$ est l'ensemble des arcs représentant la relation entre deux sommets f_1 et f_2, "f_1 à l'instant t réfère au concept représenté par f_2".
- S: S est le problème de segmenation. $B_t = S(i_t,\bar{i}_t)$. $\mathcal{S}(p,q)$ est la fonction de similarité entre un ensemble de blobs q et une personne p.
- T: $\bar{T}_t$ est l'ensemble des arcs représentant une relation temporelle entre deux sommets f_1 et f_2: "f_1, à l'instant $t-1$, est le même concept que f_2 à l'instant t"
- V: $\bar{V}_t$ est l'ensemble des sommets représentant les comportements jusqu'à t
- Z: Z est le problème de suivi de personnes : $(P_{t+1},T_{t+1}) = Z(i_{t+1},i_t,\bar{i}_t,P_t)$.
- κ: κ est l'heuristique de recherche d'une partition idéale d'un ensemble de blobs parmi l'ensemble des partitions possibles.
- Φ: Φ_i est la i^{me} fonction de correspondance.
- χ: χ est l'application de $(P^*_{t-1} \times Q^*_t)$ dans l'ensemble des diagnostics valides.
- eχ: χ est l'application de $(P^*_{t-1} \times Q^*_t)$ dans R

– e$\mathcal{M}\chi$: est la matrice representant eχ.

Chapitre 1 Introduction

Les recherches dont ce document fait état sont relatives à un programme appelé VSIS. Le but de ces recherches est l'élaboration d'un programme capable de reconnaître certains comportements humains. Ces recherches ont été menées par l'équipe TELESCOPE du GIE DYADE, (association de l'INRIA et de BULL).

Au cours de l'année 2000, le programme VSIS fut l'objet d'un inhabituel impact médiatique. Un certain nombre de publications dans des revues de vulgarisation scientifique tel que Sciences et Avenir [93] ou l'Ordinateur Individuel [66], amena VSIS à se voir décerner un des prix BIG BROTHER AWARD 2000, associant les recherches dont il était issu, à un certain nombre de contributions contraires aux libertés individuelles. La question de savoir si ces recherches avaient un réel caractère intrusif méritait d'être posée. Quel était le danger d'une telle technologie? Un premier argument, rapidement éludé, consistait peut-être à dire que le caractère intrusif de ce programme provenait de l'utilisation de vidéos de caméras de surveillance. Difficile d'imaginer que la psychose provenait de ce point, dans la mesure où ces réseaux de caméras préexistaient largement à VSIS et que sans nulle doute, personne ne s'inquiète plus d'être filmé en allant à la banque, au supermarché ou à l'aéroport. Non, plus probablement l'inquiétude avait pris sa source dans le fait que ces vidéos n'était pas analysées par un humain mais par une machine; une machine qui "voit" par la caméra, qui "suit" les personnes, qui "décide" d'alerter des opérateurs lorsqu'elle "reconnaît" certains comportements. Bref, une machine qui semble intelligente mais dont on ne sait pas comment elle marche. Turing écrivait dans un article en 1950: "Est intelligente une machine qui fait illusion et passe pour intelligente aux yeux des hommes" et Ganascia de surenchérir en 1993 dans son ouvrage "L'intelligence Artificielle" [43]: "Il ne s'agit donc pas de savoir si une machine est effectivement intelligente, si elle connaît ou si elle ressent des émotions, mais de savoir si elle peut nous apparaître comme telle". Alors, si VSIS peut susciter quelques inquiétudes parce qu'il apparaît comme une machine intelligente, du point de vue de l'intelligence artificielle, c'est un bon programme.

Par quoi est créée cette illusion d'intelligence? La plus probable explication à cette question réside dans la distance qui existe entre la nature des entrées et des sorties. Dans notre cas, les entrées étant du flux vidéo et les sorties des conclusions sémantiques sur la réalisation de comportements, la distance parait suffisamment grande pour ne pas comprendre le passage de l'un à l'autre, pour ne pas comprendre comment ça marche; pour paraître

intelligent ou dangereux. Ce document a pour but d'expliquer comment ça marche; de détailler points par points les opérations de ce programme qui vont permettre de reconnaître des comportements à partir d'un flux vidéo. En effet, ce document decrit de façon complète l'approche que nous avons suivie pour l'élaboration de VSIS; complète afin de faire disparaître toute forme de mystère et d'éviter l'écueil d'une solution basée sur une juxtapotion de solutions d'une séquence de sous-problèmes, dont la justesse apparaît souvent discutable.

En revanche, il ne sera pas question ici de raffinement de solutions existantes. Ce document n'a pas pour ambition de fournir les spécifications d'un programme prêt à être embarqué. Il ne sera ni question de contrôle de processus, d'autonomie du système ou encore de gestion d'incertitude. Tout ces problèmes étant laissés à ceux dont c'est le métier. La littérature du domaine ayant d'ailleurs déjà apportée un certain nombre de résultats [7, 40, 79, 21, 16, 22]. La littérature du domaine fera d'ailleurs l'objet du chapitre 2.

Le principe général consiste à se doter, outre d'un flux vidéo, d'un ensemble de descriptions des comportements que l'on souhaite reconnaître d'une part, et d'autre part d'une description du décor dans lequel est filmé la scène. C'est à dire un ensemble de modèles sémantiques représentant les comportements à reconnaitre et un ensemble de modèles principalement géométriques représentant les différentes parties visibles de la scène (sols, murs, portes, meubles, etc ...). Ainsi, on extrait du flux vidéo à temps fixe, les indices du mouvement des personnes, afin de calculer une description de ceux-ci. C'est à dire estimer leur taille ou leur volume et les replacer dans le décor. Avec cette description complète de la scène (personnes + décor) à un instant donné, on calcule la façon dont la scène a évolué afin d'obtenir une description spatiale et temporelle de la scène. A partir de cette description spatiale et temporelle les différents comportements prédéfinis pourront être reconnus.

Sans détailler plus avant cette approche (ceci sera fait dans le chapitre 3) disons simplement que ce problème est modélisé comme le calcul incremental d'un graphe (appelé graphe d'interprétation) dont les sommets représentent les éléments physiques (les personnes et les éléments du décor) et les éléments abstraits (les manifestations des comportements) de la scène au cours du temps. Le problème de reconnaissance de comportements consiste a mettre à jour le graphe d'interprétation à partir d'une image issue du flux vidéo, du modèle de scène et des modèles de comportements tout en minimisant les différences entre la représentation donnée par le graphe d'interprétation et le phénomène physique.

D'un point de vue algorithmique, ce problème repose sur trois problèmes aux caractéristiques très différentes. Le premier problème, dont le chapitre 4 fait l'objet, est l'extraction de description des humains à partir d'une image. Ce problème sera appelé la reconnaissance de personnes. La méthode proposée dans le chapitre 4 consiste à résoudre ce problème en

trois étapes. La première étape a pour but de calculer une image idéale de la scène vide; c'est à dire une image ayant les mêmes conditions que l'image courante du flux mais dépourvue de la projection des humains. La seconde étape a pour objectif d'extraire sous forme structurée les pixels issus de la différence entre l'image idéale de la scène vide et l'image courante. La structure adoptée est une structure de région connexe de points appelés blobs. Enfin la dernière étape a pour double but, d'une part d'identifier les blobs relatifs à une quelconque forme de bruit, et d'autre part de regrouper les blobs issus de la projection de même personnes. De façon plus précise, la première étape consiste en le calcul d'une image composite à partir d'une image idéale de l'instant précédent et d'une image réelle de l'instant précédent. La seconde étape consiste en une séquence d'opérateurs arithmétiques et morphologiques: différence absolue, seuillage, érosion, dilatation, analyse des composants connexes. Enfin la troisième étape est le calcul d'une partition de l'ensemble des régions connexes vérifiant certaines propriétés. La méthode de résolution de cette étape consiste en un parcours heuristique de l'ensemble des partitions possibles dont l'heuristique est basée sur un modèle de personne hybride 2D/3D/densité. Nous verrons aussi dans ce chapitre, les avantages et inconvénients de la méthode proposée.

Le second problème est la reconstruction incrémentale de l'évolution de la scène. Ce problème sera l'objet du chapitre 5. La solution proposée pour résoudre le problème de mise en correspondance temporelle consiste en le calcul d'un diagnostic optimal de l'évolution du système entre deux frames. A partir d'un ensemble de primitives, appelées fonctions de mise en correspondance, on construit, à chaque frame, un ensemble de diagnostics possibles dont on garde le plus vraisemblable. Ce diagnostic optimal conditionne alors l'évolution de la description du système. Nous verrons aussi dans ce chapitre, les avantages et inconvénients de la méthode proposée.

Comme nous le verrons, l'un des principaux inconvénients de la méthode est de réussir à la paramétrer. Le chapitre 6 propose une méthode d'apprentissage pour la paramétrisation de l'algorithme en question. On se place pour cela dans le cadre des algorithmes génétiques. Dans ce paradigme, un jeu de paramètres est vu comme un individu appelé chromosome. Un ensemble de chromosomes est appelé une population. Le principe général d'apprentissage par algorithme génétique consiste à faire évoluer cette population jusqu'à un état suffisamment proche de la solution cherchée, c'est à dire une population contenant un chromosome associé à un jeu de paramètres efficaces. Le principe d'évolution consiste à évaluer la qualité de chaque jeu de paramètres (c'est à dire le chromosome) afin de n'en garder, pour la population suivante, qu'une combinaison des plus efficaces. De façon plus précise, notre approche consiste à se doter d'une séquence vidéo connue associée à un résultat idéal (un graphe optimal) que l'on souhaiterait obtenir par le suivi de personne et de

faire évoluer une population en comparant le résultat obtenu par les chromosomes de cette population au résultat idéal.

Enfin, le troisième problème est la reconnaissance des comportements proprement dite à partir de la description spatiale et temporelle de l'évolution de la scène. Ce problème fera l'objet du chapitre 7. Comme nous le verrons, ce problème pose deux problèmes distincts: la représentation des comportements et la reconnaissance des comportements. Nous proposerons dans le chapitre 7, un formalisme permettant de représenter des comportements comme des ensembles de prédicats booléens dont les variables sont des sommets du graphe d'interprétation. Nous proposerons en outre, dans le chapitre 7, un algorithme de reconnaissance. Le principe de cet algorithme est de convertir chaque modèle de comportement en un ensemble de modèles particuliers dont la reconnaissance peut être résolue comme un problème de satisfaction de contrainte.

Le chapitre 8 sera consacré aux résultats obtenus avec cette approche. Nous présenterons cinq applications de notre approche. Le premier cas présente, dans la section 8.1, un ensemble de modèles de comportements de base tels que "entrer", "s'approcher de", "s'arrêter", etc... dont le but est de faciliter d'élaboration d'autres bases de comportements. Le second cas présente dans la section 8.2, des résultats de notre approche pour la sécurité dans les stations de métro. Dans la même veine, le troisième exemple dans la section 8.3, est focalisé sur la sécurité en agences bancaires. Le quatrième exemple d'utilisation présenté dans la section 8.4, est relatif à une application d'aide au travail médiatisé. Le dernier exemple présenté dans la section 8.5, est une étude sur la reconnaissance de comportements basée sur l'interaction humains/objets dans les parkings.

Chapitre 2 État de l'art

Comme il en a été fait mention dans l'introduction, la reconnaissance de comportements est principalement liée à trois grands thèmes que sont la reconnaissance de personnes, le suivi de personnes et la reconnaissance de comportements. Nous verrons tour à tour comment chacun de ces thèmes a pu être envisagé dans la littérature.

2.1 Segmentation d'image

Pendant la décennie passée, beaucoup de travaux ont été effectués sur la segmentation d'images dans le cadre de l'analyse du mouvement. Pour l'ensemble de ces travaux, le but est d'extraire de chaque image d'un flux un ensemble de primitives servant d'entrées à différents algorithmes de suivi ou d'analyse. On trouve principalement deux types de primitives: les segments [19, 97] issus majoritairement d'algorithmes de détection de contours [68, 94], [99, 101] et les régions connexes de points [65].

Deux familles de méthodes à base de régions se distinguent alors. Une première famille, dont nous ne ferons pas état ici, se caractérise pas l'analyse du mouvement dans le flux. Ces méthodes sont appelées méthodes à flot optique [77, 8, 13, 46, 67, 71]

La seconde famille de méthodes se caractérise par l'extraction de primitives relatives aux changements dans l'image. Quand la première famille de méthode a pour but de calculer un ensemble de vecteurs estimant le mouvement général de l'image, la seconde famille a pour but déterminer les parties de l'image que l'on considère avoir changées.

Bien évidement, déterminer si une partie de l'image a changé ou non, consiste à comparer cette image avec une autre image (que l'on appellera l'image de référence). Ainsi en fonction de ce que l'on considère comme image de référence cette notion de changement prend un sens et des propriétés différentes.

Si l'image de référence est une image de la scène vide, on a à faire à la classe de méthodes d'extraction de fond [34, 63, 80]. Si l'image de référence est une des précédentes images du flux, on a à faire à la classe de méthodes de détection de changement.

Dans la majorité des cas, le but poursuivi est l'extraction de fond afin que les primitives extraites des images soient les projections des humains, véhicules, etc ... Ainsi le problème de l'analyse d'une image en vue de l'extraction du fond revient à comparer cette image avec

une image de référence ne décrivant que le fond, ou encore un modèle de référence du fond.

Des lors la différence entre l'ensemble des méthodes poursuivant cet objectif réside dans la quantité d'informations investies dans le modèle du fond. Plus la quantité d'information est petite (le minimum étant une image) plus le résultat est susceptible d'être bruité et plus la méthode est rapide. A l'opposé, plus la quantité d'information [51, 31, 49] est grande, plus la comparaison est fondée, mais plus la méthode est lourde.

2.2 Suivi de personne

A l'instar de la segmentation, beaucoup de travaux ont été effectués sur le suivi d'humains pour différentes applications et selon des approches très diverses comme en témoigne un certain nombre de "Surveys" tels que [1, 44, 5]. Dans tous les cas, le suivi d'humain consiste à calculer une description temporelle d'une scène composée de quelques humains filmée par une ou plusieurs caméras. Le problème de la transcription d'un flux vidéo en ensemble de descriptions temporelles a été adressé pour différentes tâches telles que la compression vidéo, les interfaces homme machine ou l'analyse de scène. Nous nous concentrerons sur le travail effectué pour l'analyse de scène. Dans ce cas ci, l'analyse de scène signifie que la description temporelle des humains dans la scène sera l'entrée d'autres algorithmes tels que le comptage d'humains [92, 53], analyse de gestes et de posture humaine [2, 45, 100], l'analyse de scènes [59], la reconnaissance d'événements [6, 69] ou la reconnaissance de scénarios [24].

2.2.1 Contexte de suivi d'humains

L'une des différences entre ces travaux réside dans le genre de scène dans lequel le suivi d'humains doit être fait. De ce point de vue, nous pouvons trouver divers environnements tels que les bureaux [36], les laboratoires [33, 2, 100], les parkings [70], les stations de métro [86], l'environnement extérieur [9, 52] ou même les stades de sport [59, 3].

Le contexte du suivi a une incidence directe sur la complexité du suivi. En effet, du contexte de suivi dépendent les conditions naturelles telles que les conditions d'illumination (adressé dans [39]) ou le caractère changement du décor (contexte de bureaux). De plus, du contexte dépend souvent le point de vue général. C'est à dire le niveau d'amplitude des modifications des objets que l'on cherche à suivre.

2.2.2 Modèles d'humain

La deuxième source des différences entre les travaux mentionnés ci-dessus est le genre de modèle employé pour représenter un humain, dont une des difficultés réside dans la nature non-rigide [56, 57].

Trois types de modèles sont utilisés: les modèles 2D, les modèles 3D et les modèles hybrides. On a proposé plusieurs approches utilisant les 2D modèles comme la classique boîte englobante, les B-splines cubiques [9, 32, 18, 10], les ensembles d'ellipses [52] ou les ensembles de rubans [85]. La deuxième catégorie de modèles d'humains est le modèle 3D, c'est à dire modèle volumétrique composé d'ensemble de volumes 3D tels que des ensembles de cylindres [45, 27, 35], des ensembles d'ellipses [75] ou des modèles non paramétriques [33]. La dernière catégorie des modèles est le modèle hybride qui est généralement un mélange des modèles de 2D et de 3D. Le fait est que, lorsque le modèle devient plus complexe (c.-à-d. quand le nombre de degrés de liberté grandit), apparier entre deux frames est plus facile, mais l'instanciation à chaque frame de tous les degrés de liberté devient plus difficile.

2.2.3 Méthode de suivi

La troisième source des différences entre les approches trouvées dans la littérature est la technique employée pour instancier les modèles. C.-à-d. différentes approches employées pour transformer le flux vidéo en ensembles de modèles instanciés. Deux familles d'approches peuvent être trouvées. La première approche consiste à reconnaître, dans l'image actuelle, un modèle instancié dans la frame précédente [85, 58, 57, 83].

Avec ce genre d'approche, la correspondance entre deux descriptions d'un humain semble être plus précise, mais la question de l'initialisation et d'arrêt des pistes est souvent ignorée. La deuxième approche consiste à reconnaître dans l'image actuelle un ensemble de modèles sans n'importe quelle autre information, et dans un second temps, mettre en correspondance les modèles instanciés avec ceux de la frame précédente [75, 52, 24]. Avec cette approche, le problème est de trouver la correspondance optimale entre deux ensembles de descriptions parmi toutes les correspondances possibles.

Dans ce contexte, notre approche traite trois genres de scènes telles que des bureaux, des stations de métro et des banques. Nous utiliserons un modèle hybride simple composé d'un rectangle 2D et d'un cylindre 3D correspondant, parce que nous croyons fortement qu'un modèle avec un nombre élevé de degrés de liberté ne peut pas être identifié avec une bonne et constante précision sous les contraintes "Real-World". En terme de techniques utilisées pour la reconnaissance, notre approche fait partie de la seconde catégorie.

2.3 Analyse de comportements

Au même titre que les travaux menés en suivi de personnes, l'analyse de comportements a été, durant la décennie passée, un domaine riche en expérimentations diverses et variées. Bien que le terme de comportements ne soit pas de rigueur dans la majorité des cas, le problème est bien le même que l'on parle de chroniques [37], d'épisodes [17], d'actions [61], d'activités [36] ou de plans [96]. Ce problème consiste à instancier un ensemble de modèles de comportements avec les éléments d'un flux. Un modèle complètement instancié est dit reconnu et correspond à une certaine interprétation.

Dès lors, les deux axes qui différencient l'ensemble des travaux menés en analyse de comportements sont, d'une part le type de représentations utilisée pour décrire les comportements, et d'autre part les techniques employées pour instancier les modèles avec les éléments du flux.

2.3.1 Type de représentation

Il nous faut tous d'abord, différencier deux types d'approches. Les approches avec lesquelles la description est faite explicitement, c'est à dire par un utilisateur, et les approches où la description est obtenue par apprentissage (souvent statistique). Nous ne considérerons ici que les approches de la première catégorie [30, 25, 82, 74].

Le premier axe de différences entre l'ensemble des travaux du domaine est le type de représentation utilisée, c'est à dire le type de modèle. Le modèle représente le comportement à reconnaître dans le cadre d'un formalisme donné. Le choix d'un formalisme contraint alors les possibilités de représentation. C'est ce que l'on appellera l'expressivité du formalisme. De façon générale, le modèle est donc une combinaison non-instanciée d'éléments de base. L'expressivité du formalisme dépend alors du type d'éléments de base et du mode de combinaison.

En ce qui concerne, le type d'éléments de bases deux catégories peuvent être trouvées. La première catégorie constitue l'ensemble des approches ayant pour entrée un flux d'objets et de personnes [96, 20, 14, 6, 4, 62]. La seconde catégorie constitue l'ensemble des méthodes ayant pour entrée un flux d'événements [37, 38, 24, 23, 48, 64, 26, 3, 54, 78]

Bien que les approches basées sur un flux d'objets semblent être plus bas niveau donc plus expressives, la majorité d'entre-elles opère une conversion du flux objets en flux d'événements.

La différence majeure d'expressivité en matière d'éléments de base de la représentation réside dans le mode logique sous-jacent. En d'autre termes, les éléments de base de la

représentation sont des prédicats d'ordre 0 (LP0) ([48, 64, 26, 78] ou des prédicats d'ordre 1 (LP1) [37, 38, 24, 3, 54]. A ce titre, il est tout à fait clair qu'un formalisme ayant comme éléments de base des prédicats d'ordre 1 est bien plus expressif qu'un formalisme reposant sur des prédicats d'ordre 0.

En ce qui concerne les différences liées au mode de combinaison des éléments de base dans un modèle, deux points sont importants. Le premier point est lié au mode de combinaison logique entre les éléments de base et le second point est lié au mode de combinaison temporel des éléments de base.

On trouve, en matière de combinaison logique, trois grandes familles de combinaisons logiques. La première famille d'approches ne permettant que des conjonctions [37, 38, 3, 54] entre les éléments d'un modèle, la famille d'approches permettant les conjonctions et la négation [24] et une troisième famille d'approches permettant les disjonctions et les conjonctions [48, 96, 23, 16, 6, 78, 61, 60]. A ce titre, il est clair que les formalismes proposant un plus grand choix d'opérateurs logiques offrent une plus grande liberté à l'utilisateur pour décrire ces comportements.

On trouve en matière de combinaison temporelle, deux types d'approches. Le premier type d'approches est basé sur une algèbre dite de points et la seconde sur une algèbre d'intervalles. En d'autre termes, le premier mode de combinaisons temporelles est basé sur des opérateurs arithmétiques classiques (<,>,=) associés à des points dans le temps [37, 48, 96, 23, 16], etc ...) et le second mode de combinaisons temporelles est basé sur une algèbre dont les primitives sont des intervalles de temps composés à l'aide d'opérateurs spécifiques dont les fondements ont été jetés par Allen [78, 3, 54] (puis étendus au raisonnement spatial dans [11, 12, 47, 28, 29]). De ce point de vue, une algèbre d'intervalles offre, dans de nombreux cas, une plus grand souplesse de représentation que les algèbres de points.

2.3.2 Type de reconnaissance

La seconde différence entre l'ensemble des méthodes proposées dans la littérature réside dans les techniques de reconnaissance utilisées. De ce point de vue, la nature des modèles à reconnaître influence le choix de techniques de reconnaissance.

Pour simplifier, disons que reconnaître un modèle basé sur prédicat d'ordre 0 associé à une algèbre classique ne nécessite pas de gestion trop lourde, alors que la reconnaissance d'un modèle basé sur des prédicats d'ordre 1 ou basé sur une algèbre d'intervalles implique une gestion plus complexe.

Dans [96], l'approche pour la reconnaissance est basée sur la mise à jour de réseau de Pétri modélisant les comportements (appelé Plan). Le passage d'un jeton d'un état à un

autre ne nécessite pas de duplication particulière. Dans la même veine, l'approche utilisée par les auteurs de [14, 70] est la mise à jour d'un automate d'état en fonction de valeur de certaines propriétés calculées sur chaque personne. La méthode, limitée aux modèles basés sur une unique personne ne semble pas nécessiter de gestion combinatoire ou de duplication particulière. Dans la même veine encore, les auteurs de [6] décrivent un approche basée sur des prédicats d'ordre 0 modifiant l'état d'un automate. Les prédicats d'ordre 0 sont utilisés aussi dans [64]. La technique employée, ici, est un système expert type CLIPS. La méthode semble efficace, bien que les auteurs restent discrets sur le comportement du moteur d'inférence dans le cas de modèles quelconques. Cette remarque pourrait être formulée aussi pour [4] décrivant un principe d'inférence basée sur un parser stochastique.

En ce qui concerne la reconnaissance de modèles basée sur de la logique d'ordre 1 ou sur de la logique d'intervalles, la technique communément admise consiste en deux temps. Une partie off-line consiste à transformer les modèles, et une partie on-line consistant à propager des contraintes sur un ensemble de modèles partiellement reconnus. Dans [38, 23, 48], les auteurs décrivent des techniques de reconnaissance de modèles basées sur des prédicats d'ordre 1 sur un algèbre classique. La partie off-line consiste alors à optimiser les modèles et la partie on-line consiste à gérer un ensemble de modèles partiellement instanciés en y propageant les contraintes des modèles en fonction des informations entrantes.

La reconnaissance de modèles basée sur des logiques d'intervalles quand elle est abordée (la plupart des travaux de ce domaine se situant au niveau de la représentation plutôt que la reconnaissance) est réalisée en traduisant les modèles dans des formalismes différents. Dans [78, 54], la partie off-line correspond à la traduction des modèles et la partie on-line consiste aussi à propager les contraintes des modèles partiellement instanciés en fonction des informations entrantes.

Pour résumer, bien que les formalismes basés sur des logiques d'ordre 0 associés à une algèbre classique soient moins expressifs la reconnaissance des modèles associés est clairement moins lourde que la reconnaissance de modèles basé la logique d'ordre 1 ou sur des algèbres d'intervalles.

Pour finir cette analyse, les techniques de reconnaissance peuvent être différenciés en fonction du niveau évaluation de la reconnaissance elle-même. On entend par évaluation toute forme de résultat sortant de la reconnaissance différent d'une réponse booléenne. La première forme d'évaluation est la prédiction. C'est à dire fournir à tout moment une évaluation de l'état de la reconnaissance d'un modèle donné. Ce problème est abordé par [38, 48]. La seconde forme d'évaluation consiste à fournir un degré de vraisemblance sur le résultat [23] ou une mesure de possibilité [48]

Dans ce contexte, notre approche se situe dans le cadre les modèles basés sur de la

logique d'ordre 1 associée à un algèbre classique dont les éléments de base sont des objets. La nouveauté de l'approche réside dans le mode de reconnaissance ne passant pas nécessairement par un niveau événements comme la plupart des approches dont les éléments de base de modèles sont des objets. De plus, cette approche, à l'opposée des approches traitant des modèles basés sur des prédicats d'ordre 1, se caractérise par l'absence de gestion d'un ensemble de modèles partiellement instanciés.

Chapitre 3 Modélisation du problème

3.1 Notations

Soit $\bar{G}_t = (\bar{F}_t, \bar{A}_t)$ un graphe appelé graphe d'interprétation montré en figure 3.1. $\bar{F}_t$ est l'ensemble des sommets représentant les concepts dates de la scène et $\bar{A}_t$ est l'ensemble des arcs représentant les relations binaires entre les concepts de la scène. On introduit sur $\bar{G}_t$, trois types de sommets notes $\bar{O}_t$, $\bar{P}_t$ et $\bar{V}_t$ et deux types d'arcs notés $\bar{T}_t$ et $\bar{R}_t$. I.e. $\bar{G}_t = (\bar{O}_t \cup \bar{P}_t \cup \bar{V}_t, \bar{T}_t \cup \bar{R}_t)$. Soit $\tilde{G}_t = (\tilde{F}_t, \tilde{A}_t)$ un graphe appelé graphe d'interprétation partiel défini par $\tilde{G}_t = (\bar{O}_t \cup \bar{P}_t \cup \bar{V}_{t-1}, \bar{T}_t \cup \bar{R}_{t-1})$. On définit $\bar{O}_t$ (resp. $\bar{P}_t$, $\bar{V}_t$, $\bar{T}_t$, $\bar{R}_t$ et $\bar{G}_t$) par $\bar{O}_t = O_0 \cup ... \cup O_t$ où O_i (resp. P_i, V_i T_i, R_i et G_i) est l'ensemble des sommets (resp. sommets, sommets, arcs, arcs et graphes) représentant les concepts à l'instant i.

3.2 Sémantique

$\bar{O}_t$ est l'ensemble des sommets représentant les objets de la scène jusqu'à l'instant t. $\bar{P}_t$ est l'ensemble des sommets représentant les personnes jusqu'à l'instant t. $\bar{V}_t$ est l'ensemble des sommets représentant les comportements jusqu'à t. $\bar{T}_t$ est l'ensemble des arcs représentant une relation temporelle entre deux sommets f_1 et f_2: "f_1, à l'instant $i-1$, est le même concept que f_2 à l'instant i". $\bar{R}_t$ est l'ensemble des arcs représentant la relation, entre deux sommets f_1 et f_2, "f_1 à l'instant i réfère au concept représenté par f_2".

On associe à chaque sommet $f \in \bar{F}_t$ huit caractéristiques appelées *attribute*

1. *name(f)* est un identificateur symbolique,
2. *time(f)* est un identificateur temporel,
3. *type(f)* est une catégorie parmi *object*, *pedestrian* et l'ensemble des types de comportements,
4. *box(f)* est une description géométrique 2D,
5. *hull(f)* est une description géométrique 3D,
6. *velocity(f)* est un vecteur vitesse 3D,
7. *properties(f)* est un ensemble de caractéristiques symboliques,
8. *references(f)* est un ensemble de couple (*name*,*time*).

Notons, que l'on parlera, par abus de langage, du *name* de f (resp. *time* de f , *type* de f , etc ...) pour parler de la valeur de l'attribut *name* (resp. *time*, *type*, etc ...) de f.

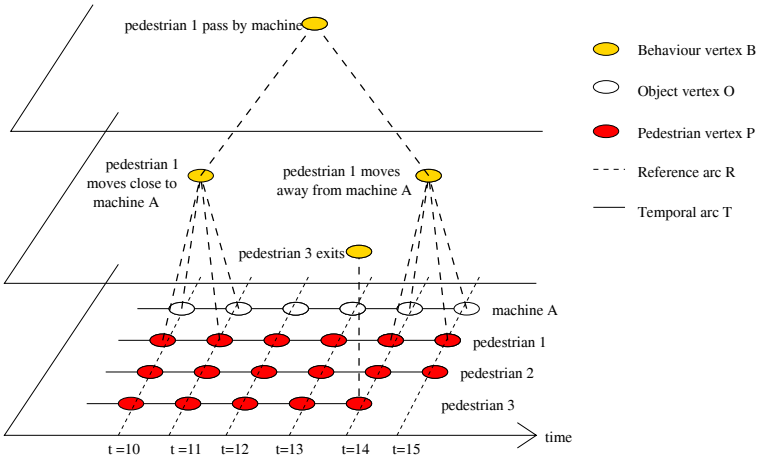


FIG. 3.1 – *Exemple de graphe d'interprétation*

3.3 Propriétés

1. $\forall f_1,f_2 \in \bar{F}_t$: $f_1 = f_2 \Leftrightarrow name(f_1) = name(f_2) \wedge time(f_1) = time(f_2)$.
2. $\forall f_1,f_2 \in \bar{F}_t$: $\exists t(f_1,f_2) \in T_t \Leftrightarrow name(f_1) = name(f_2)$, c'est à dire f_1 et f_2 représentent le même concept réel.
3. $\forall f_1,f_2 \in \bar{F}_t$: $f_1 \in references(f_2) \Leftrightarrow \exists r \in \bar{R}_t$ tel que r est un arc entre f_1 et f_2. Nous noterons $name(ref(f_1,i))$ (resp. $time(ref(f_1,i))$) le *name* (resp. le *time*) du i^{ieme} *reference* de f_1.

3.4 Notion d'individu

Soit I_t, l'ensemble des individus de $\bar{G}_t$, la partition $\bar{P}_t$ définie par :

$$\begin{aligned} I_t &= \{i_1,...,i_n\} \\ &= \{\{p_{1,\alpha},...,p_{n,\alpha}\},...,\{p_{1,\gamma},...,p_{n,\gamma}\}\} \end{aligned}$$

avec

$$\bar{P}_t = \{\{p_{1,\alpha},...,p_{1,\gamma}\} \cup ... \cup \{p_{n,\beta},...,p_{n,\delta}\}\}$$

et

$$\forall j \in [1,n] \quad \forall p_{j,\alpha} \in i_j \quad \forall p_{j,\gamma} \in i_j \quad name(p_{j,\alpha}) = name(p_{j,\gamma})$$

En d'autre termes, on définit l'ensemble des individus de $\bar{G}_t$ comme la partition dont chacun des sous-ensembles ne contient que des sommets ayant la même valeur d'attribut *name*. C'est à dire dans le cas parfait, chaque sous-ensemble correspond à un humain physique distinct. A partir de cette notion, on définit quatre mesures sur les éléments de I. Soit $start(i)$ (resp. $end(i)$) la plus petite (resp. plus grand) valeur prise par l'attribut *time* d'un élément p de i. C'est à dire que $start(i)$ (resp. $end(i)$) représente le premier instant (resp. dernier instant) ou i existe.

$$\begin{aligned} start(i) &= \min_{p \in i} time(p) \\ end(i) &= \max_{p \in i} time(p) \end{aligned}$$

Soit $t\text{-}dist(i_1,i_2)$ la mesure de distance temporelle entre deux individus i_1 et i_2. On définit $t\text{-}dist(i_1,i_2)$ par:

$$t\text{-}dist(i_1,i_2) = [\quad (start(i_1) - start(i_2))^2 + (end(i_1) - end(i_2))^2 \quad]^{\frac{1}{2}}$$

Soit $s\text{-}dist(i_1,i_2)$ la mesure de distance spatiale entre deux individus i_1 et i_2. $s\text{-}dist(i_1,i_2)$ n'est définie que si $t\text{-}dist(i_1,i_2) = 0$.

$$s\text{-}dist(i_1,i_2) = \sum_{t=start(i_1)}^{end(i_1)} dist(hull(p_{1,t}),hull(p_{2,t})) \tag{3.1}$$

avec *dist* la distance spatiale classique 3D.

3.5 Processus d'interprétation

On définit le processus global d'interprétation comme le calcul de $\bar{G}_t$ à partir de $\bar{G}_{t-1}$, un ensemble K de modèle de comportements et un triplet d'images $(i_t,i_{t-1},\bar{i}_{t-1})$ ou i_{t-1} et i_t sont deux images issues du flux vidéo à $t-1$ et t et $\bar{i}_{t-1}$ est l'image de référence à $t-1$.

Le processus d'interprétation est donc le calcul de G_t car $\bar{G}_t = \bar{G}_{t-1} \cup G_t$ et le calcul de G_t est le calcul de O_t, P_t, V_t, T_t et R_t.

Le calcul de O_t est trivial dans la mesure où le nombre des objets de la scène, ainsi que les valeurs des attributs de chacun, est constant et connu à l'instant initial.

Le calcul de P_t et T_t est appelé le suivi de personnes et le problème associé est noté Z.

$$(P_t,T_t) = Z(i_t,i_{t-1},\bar{i}_{t-1},P_{t-1})$$

Que l'on décompose en :

$$(P_t,T_t) = M(P_{t-1},C(S(i_t,i_{t-1},\bar{i}_{t-1},P_{t-1})))$$

où S est le problème de segmentation d'une image i_t en un ensemble de blobs, C est le problème de groupement des blobs et M le problème de mise en correspondance temporelle avec l'ensemble des personnes à $t-1$.
ou encore

$$(P_t,T_t) = M(P_{t-1},C(B_t))$$

où B_t est l'ensemble des blobs a t.
ou encore

$$(P_t,T_t) = M(P_{t-1},Q_t)$$

où Q_t est la partition de l'ensemble des blobs à t.

Le calcul de V_t et R_t est appelé la reconnaissance de comportements d'un base de comportements K et le problème associé est noté D et l'on a :

$$(V_t,R_t) = D(K,\tilde{G}_t)$$

que l'on décompose en :

$$(V_t,R_t) = \bigcup_{M_i \in K} P_2(M_i,\tilde{G}_t)$$

où P_2 est le problème associé au calcul des instances du modèle M_i sur $\tilde{G}_t$.

Le processus global d'interprétation peut alors être formalisé de de la façon suivante :

$$\begin{aligned}
\bar{G}_t &= \bar{G}_{t-1} \cup G_t \\
&= \bar{G}_{t-1} \cup (O_t \cup P_t \cup V_t, T_t \cup R_t) \\
&= \bar{G}_{t-1} \cup (O_t, \emptyset) \cup (P_t, T_t) \cup (V_t, R_t) \\
&= \tilde{G}_t \cup (V_t, R_t) \\
&= \tilde{G}_t \cup D(K, \tilde{G}_t) \\
&= \tilde{G}_t \cup \textstyle\bigcup_{M_i \in K} P_2(M_i, \tilde{G}_t)
\end{aligned}$$

avec

$$\begin{aligned}
\tilde{G}_t &= \bar{G}_{t-1} \cup (O_t, \emptyset) \cup (P_t, T_t) \\
&= \bar{G}_{t-1} \cup (O_t, \emptyset) \cup Z(i_t, i_{t-1}, \bar{i}_{t-1}, P_{t-1}) \\
&= \bar{G}_{t-1} \cup (O_t, \emptyset) \cup M(P_{t-1}, Q_t) \\
&= \bar{G}_{t-1} \cup (O_t, \emptyset) \cup M(P_{t-1}, C(B_t)) \\
&= \bar{G}_{t-1} \cup (O_t, \emptyset) \cup M(P_{t-1}, C(S(i_t, i_{t-1}, \bar{i}_{t-1}, P_{t-1})))
\end{aligned}$$

avec

$$O_t = O_0$$

3.6 Caractérisation de la solution du suivi de personnes

Pour caractériser la solution du suivi de personnes, nous noterons "humain" le phénomène physique que l'on souhaite décrire et $\mathcal{H}_t$ l'ensemble des humains présents dans la scène à l'instant t.

On définit en outre l'opérateur "correspond", noté $\equiv$, entre un humain h et un sommet $p \in P_t$ par: $h \equiv p$ **si et seulement si** la distance entre $hull(p)$ et le plus petit cylindre englobant l'humain h est minimale.

On dit que (P_t, T_t) est la solution de $Z(i_{t+1}, i_t, \bar{i}_t, P_t)$ **si et seulement si** :

1. $\forall p_{i,t} \in P_t : \exists\,!\ h \in \mathcal{H}_t$ tel que $h \equiv p_{i,t}$. En d'autres termes, pour chaque élément $p_{i,t} \in P_t$ il existe un humain de la scène à l'instant t correspondant à la représentation donnée par $p_{i,t}$.
2. $\forall h \in \mathcal{H}_t : \exists\,!\ p_{i,t} \in P_t$ tel que $h \equiv p_{i,t}$. En d'autres termes, pour chaque humain de la scène à l'instant t, il existe un sommet $p_{i,t} \in P_t$ correspondant à cet humain.
3. $\forall t(x,y) \in T_t$, $x \in P_{t-1}$, $y \in P_t : \exists h \in \mathcal{H}_{t-1} \cap \mathcal{H}_t$ tel que $x \equiv h \wedge y \equiv h$. En d'autres termes, pour chaque relation $t(x,y) \in T_t$, la $x \in P_{t-1}$ et $y \in P_t$ correspond au même humain h, x à l'instant $t-1$ et y à l'instant t.
4. $\forall h \in \mathcal{H}_{t-1} \cap \mathcal{H}_t : \exists t(x,y) \in T_t$ tel que $x \in P_{t-1} \wedge y \in P_t \wedge x \equiv h \wedge y \equiv h$. En d'autres termes, pour chaque humain présent dans la scène à l'instant $t-1$ et à l'instant t, il

existe une relation binaire $t(x,y) \in T_t$ tel que x correspond à cet humain à l'instant $t-1$ et y correspond à ce même humain à l'instant t

3.7 Caractérisation de la solution de la reconnaissance de comportements

De la même façon que nous avons défini la notion d'humain pour caractériser la solution du suivi de personnes, nous définissons la notion de "comportement" comme étant le phénomène physique (bien qu'abstrait) dont nous souhaitons reconnaître les manifestations. On note $\mathcal{C}_t$ l'ensemble des manifestations de comportements dans la scène à l'instant t.

Bien que la notion de manifestation d'un comportement est plus ambigüe que la notion d'humain, dans la mesure où celle-ci depend de l'interprétation qu'on en donne, nous considérons ici que cette interpretation ne depend que d'une unique expertise $\mathcal{E}$.

Ainsi on étend l'opérateur "correspond", noté $\equiv$, entre un comportement c et un sommet $v \in V_t$ par: $c \equiv v$ **si et seulement si** selon l'expertise $\mathcal{E}$, v est une manifestation du comportement c.

On dit que (V_t,V_t) est la solution de $D(K,\tilde{G}_t)$ **si et seulement si** :

1. $\forall v_{i,t} \in V_t : \exists\,!\ c \in \mathcal{C}_t$ tel que $c \equiv v_{i,t}$. En d'autres termes, pour chaque élément $v_{i,t} \in V_t$ il existe une manifestation d'un comportement à l'instant t correspondant à la représentation donnée par $v_{i,t}$.
2. $\forall c \in \mathcal{C}_t : \exists\,!\ v_{i,t} \in V_t$ tel que $c \equiv v_{i,t}$. En d'autres termes, pour chaque manifestation d'un comportement donné à l'instant t, il existe un sommet $v_{i,t} \in V_t$ correspondant à cet manifestation

3.8 Conclusion

Nous avons présenté dans ce chapite la modélisation du problème que nous souhaitons résoudre. Ce problème se présente comme un calcul incrémental de graphes dont les sommets représentent les concepts impliqués (personnes, objets et comportements).

L'accent a été mis dans ce chapitre sur d'une part la présentation unifiée du problème complet (de l'images aux comportements) tendant par la même de voir ce problème non pas comme une séquence de sous-problèmes, mais comme un problème à part entière. D'autre part, l'accent a été mis sur l'identification de problématiques distinctes comme la segmentation d'images, le groupement des blobs, la mise en correspondance temporelle ou la reconnaissance de comportements dont les spécificités sont clairement différentes.

Chapitre 4 Reconnaissance de personnes

Ce chapitre a pour objectif de détailler la méthode utilisée pour résoudre le problème de reconnaissance de personnes. Le but de cette méthode, abordée dans [86] et [89], est d'identifier dans chacune des images du flux, les pixels susceptibles d'être les projections des humains de la scène.

La difficulté de ce problème réside dans l'existence de plusieurs niveaux de bruit dans les images du flux. Le premier niveau de bruit est celui du bruit purement image souvent associé à un bruit blanc gaussien. Le second niveau de bruit est associé aux conditions physiques, c'est à dire les changements d'illumination, les ombres, les reflets, etc ... Enfin, un troisième niveau de bruit que l'on peut qualifiée de bruit structurel vient s'ajouter aux deux premières sources de perturbations. Cette source de bruit est liée à l'ensemble des éléments mobiles ou changeants non-humains de la scène. C'est à dire d'une part les éléments changeants du décor et d'autre part les objets rapportés de l'extérieur de la scène.

La méthode proposée dans ce chapitre consiste à résoudre ce problème en trois étapes. La première étape a pour but de calculer une image idéale de la scène vide; c'est à dire une image ayant les mêmes conditions que l'image courante du flux mais dépourvue de la projection des humains. La seconde étape a pour objectif extraire sous forme structurée les pixels issus de la différence entre l'image idéale de la scène vide et l'image courante. La structure adoptée est une structure de région connexe de points appelés blobs. Enfin la dernière étape a pour double but, d'une part d'identifier les blobs relatifs à une quelconque forme de bruit, et d'autre part de regrouper les blobs issus de la projection de même personnes.

De façon plus précise, la première étape consiste en le calcul d'une image composite à partir d'une image idéale de l'instant précédent et d'une image réelle de l'instant précédent. La seconde étape consiste en une séquence d'opérateurs arithmétiques et morphologiques: différence absolue, seuillage, érosion, dilatation, analyse des composants connexes. Enfin la troisième étape est le calcul d'une partition de l'ensemble des régions connexes vérifiant certaines propriétés. La méthode de résolution de cette étape consiste en un parcours heuristique de l'ensemble des partitions possibles dont l'heuristique est basée sur un modèle de personne hybride 2D/3D/densité.

Après avoir détaillé, dans la section 4.1, la problématique et la solution choisi pour la segmentation, nous définirons, dans la section 4.2, la notion et l'utilisation du modèle de

scène. la section 4.3 aura pour objet le problème de groupement des blobs, ainsi que la solution retenue. La section 4.4 proposera quelques résultats obtenus par notre approche.

4.1 Problème de segmentation

4.1.1 Définition du problème de segmentation

Soient i_t (resp. i_{t-1}) l'image du flux vidéo à l'instant t (resp. $t-1$), $\bar{i}_{t-1}$ est l'image de référence à $t-1$ et P_{t-1} l'ensemble des sommets de G_{t-1} représentant les humains présents dans la scène à $t-1$. On définit le problème de segmentation S comme le calcul d'un ensemble de régions connexes de pixels, noté B_t. On dit que $B_t = \{b_{1,t},...,b_{k,t}\}$ est la solution de $S(i_t,i_{t-1},\bar{i}_{t-1},P_{t-1})$ **si et seulement si** $\forall i \in [1,k] \quad \forall$ pixel $i_t(x,y) \in b_{i,t}$, $i_t(x,y)$ est la projection sur le plan image d'une partie d'un humain.

4.1.2 Résoudre le problème de segmentation

On propose de résoudre le problème de segmentation en appliquant une analyse des régions connexes à une image binaire obtenue par différence seuillée entre l'image i_t et une image de référence $\bar{i}_t$ calculée à partir de $\bar{i}_{t-1}$,i_{t-1} et P_{t-1} de la façon suivente:

$$
\begin{aligned}
B_t &= CC(TH_\alpha(DIFF(i_t,\bar{i}_t))) \\
\bar{i}_t(x,y) &= \begin{cases} \bar{i}_{t-1}(x,y) & \text{SI } \exists f \in P_{t-1} \text{ tel que } i_{t-1}(x,y) \in box(f) \\ i_{t-1}(x,y) & \text{SINON} \end{cases}
\end{aligned}
$$

où CC est l'opérateur d'analyse des régions connexes, TH_α est l'opérateur de seuillage binaire d'une image par rapport à un seuil α et $DIFF$ l'opérateur de différence absolue de deux images.

En d'autres termes, $\bar{i}_t(x,y)$ est composée des pixels de l'image réelle précedente si ces pixels ne sont pas des pixels de personnes et composée des pixels de l'image de référence precedente dans le cas contraire.

Notons que :

- Afin d'obtenir des régions connexes avec une morphologie plus lisse, on applique à l'image binaire un couple d'opérateurs érosion/dilatation définis dans [50]. On a alors :

$$B_t = CC(DIL(ERO(TH_\alpha(DIFF(i_{t+1},\bar{i}_{t+1})))))$$

où DIL est l'opérateur de dilatation et ERO l'opérateur d'érosion.

– Afin de contrôler l'influence de i_{t-1} sur le calcul de $\bar{i}_t$, on pondère son intensité par un pourcentage de l'intensité de $\bar{i}_{t-1}$. On a alors :

$$\bar{i}_t(x,y) = \begin{cases} \bar{i}_{t-1}(x,y) & \text{SI } \exists f \in P_{t-1} \text{ tel que } i_{t-1}(x,y) \in box(f) \\ \beta.i_{t-1}(x,y) + (1-\beta).\bar{i}_{t-1}(x,y) & \text{SINON} \end{cases}$$

4.1.3 Analyse de la solution du le problème de segmentation

L'influence de la valeur de seuil α est prépondérante sur la qualité de la segmentation. Plus la valeur de α est faible, plus le nombre et la taille des blobs est faible. En d'autres termes, avec une valeur de α faible, le nombre de faux blobs (blobs non issus d'humain) est faible. En revanche, plus la valeur de α est faible, plus le nombre de blobs ratés (blobs non détectés) est grand. *A contrario*, avec une valeur de α forte, le nombre de blobs ratés est faible et le nombre de faux blobs fort.

La valeur β influence la qualité des résultats en terme de résistance aux changements de conditions dans la scène. En effet, avec $\beta \to 0$, le niveau d'intégration est fort et donc la méthode est très réactive aux changements de conditions. En revanche, avec $\beta \to 0$, la méthode devient sensible aux erreurs commises sur P_t. Avec $\beta \to 1$, la méthode devient plus sensible aux changements de conditions de la scène, mais plus indépendante aux erreurs commises sur P_t.

La complexité de la solution proposée au problème de segmentation dépend du nombre de parcours des images impliquées. Toute les images impliquées ayant la même taille, nous noterons s la taille commune de toute ces images. Les opérateurs CC, DIL, ERO ne requièrent qu'un parcours de l'image. L'opérateur $DIFF$ associé à TH_α requière un parcours pour chacune des images impliquées et le calcul de $\bar{i}_t$ nécessite un seul parcours. La complexité de la solution est donc $5s$

4.2 Modèle de scène

Nous avons détaillé, dans la section précédente, la solution retenue pour la segmentation d'image dont les résultats doivent maintenant être repris par l'algorithme de regroupement des blobs dont le but est de calculer un ensemble de descriptions des personnes dans la scène. Ce point soulève le problème du passage de l'information 2D à de l'information 3D, abordé en outre dans [90]. Le principe de modèle de scène assure ce passage. L'idée d'information *a priori* sur l'environnement n'est pas une idée neuve en vision [15, 95], mais on tente ici, d'une part d'utiliser cette information le plus tôt possible et, d'autre part d'intégrer ce concept au reste du problème.

4.2.1 Définition du modèle de scène

On appelle O_t le modèle de la scène à t. Comme il l'a déjà été dit dans le chapitre 3, le nombre de sommets ainsi que les valeurs des attributs est considéré ici constant. On a alors $O_t = O_0$. En d'autres termes, le modèle de scène est connu à l'instant initial. On considérera le modèle de scène comme une information *a priori* et exacte.

4.2.2 Utilisation du modèle de scène

On définit deux fonctions liées à O_t: $\mathcal{P}(O_t,x,y)$ la projection d'un point du référentiel image dans le référentiel scène, $\mathcal{M}(O_t,x,y,m)$ le calcul d'une mesure de distance 3D à partir d'une mesure de distance 2D.

$$
\begin{aligned}
\mathcal{P}(O_t,x,y) &= (X,Y,Z) \in \Delta(O_t,x,y) \text{ tel que } d_{cam}(X,Y,Z) = \underset{(X,Y,Z)\in\Delta(O_t,x,y)}{MIN} d_{cam}(X,Y,Z) \\
d_{cam}(X,Y,Z) &= [(X-X_c)^2 + (Y-Y_c)^2 + (Z-Z_c)^2]^{\frac{1}{2}} \\
\Delta(O_t,x,y) &= \{(X,Y,Z) \in (\Delta(x,y) \cap \bigcup_{o_i \in O_t} (hull(o_i)))\} \\
\Delta(x,y) &= \{(X,Y,Z) \text{ tel que } PROJ(X,Y,Z) = (x,y)\}
\end{aligned}
$$

avec (X_c,Y_c,Z_c) les coordonnées d'un point focal de la caméra dans le référentiel scène et $PROJ$ la projection d'un point du référentiel scène dans le référentiel image [41, 84].

En d'autres termes, le point (X,Y,Z) = $\mathcal{P}(O_t,x,y)$ est défini comme étant le point le plus proche de la caméra parmi l'ensemble des points d'intersection entre l'union des *hull*

des éléments de O_t et la droite Δ définit l'ensemble des points de la scène qui se projettent en (x,y) dans l'image.

$$\begin{aligned}
\mathcal{M}(O_t,x,y,m) &= [(X_0 - X_p)^2 + (Y_0 - Y_p)^2 + (Z_0 - Z_p)^2]^{\frac{1}{2}} \\
(X_0,Y_0,Z_0) &= \mathcal{P}(O_t,x,y) \\
(X_p,Y_p,Z_p) &= \mathcal{P}(O_t,x_p,y_p) \text{ tel que } [(y_0 - y_p)^2 + (y_0 - y_p)^2]^{\frac{1}{2}} = m \text{ et } (X_p,Y_p,Z_p) \in \Pi(x,y)
\end{aligned}$$

où $\Pi(x,y)$ est le plan parallèle au plan image passant par $\mathcal{P}$ (O_t,x,y).

En d'autres termes, on calcule une mesure 3D m associée à un point 2D p_1 par la mesure de distance entre deux points P_1 et P_2 tel que P_1 est le projeté de p_1 dans la scène, P_2 est le projeté de p_2 dans le plan parallèle au plan image passant par P_1 avec p_2 un point du plan image dont la distance avec p_1 est m.

On étend la fonction $\mathcal{P}(O_t,x,y)$ à la projection d'un rectangle quelconque de hauteur h et de largeur w dont le milieu du point bas est p_0 de coordonnées (x,y) noté $\mathcal{P}(O_t,x,y,w,h)$. On pose que $\mathcal{P}(O_t,x,y,w,h)$ est un cylindre de largeur $\mathcal{M}(O_t,x,y,w)$ et de hauteur $\mathcal{M}(O_t,x,y,h)$ dont le centre de la base est $\mathcal{P}(O_t,x,y)$.

La figure 4.1 illustre le calcul de $\mathcal{P}(O_t,x,y,w,h)$. La droite $\Delta(x,y)$, composée des points de la scène se projetant en (x,y) est représentée par une portion de droite en noir épais. On a:

$$\begin{aligned}
\Delta(O_t,x,y) &= \{P1,P2,P3,P4\} \\
\mathcal{P}(O_t,x,y) &= P1
\end{aligned}$$

Ainsi, à partir de la donnée du modèle de scène, le passage de l'information 2D à de l'information 3D est organisé de façon non-ambiguë.

4.3 Problème de groupement

4.3.1 Problème de groupement

Soit $B_t = \{b_{1,t},...,b_{k,t}\}$ l'ensemble des blobs à t. Chaque blob est défini par un vecteur $(x_{2D},y_{2D},w_{2D},h_{2D},a_{2D}) \in \mathbb{R}^5$, où x est la valeur moyenne des coordonnées x_{2D} et y_{2D} le minimum des valeur des coordonnées y_{2D} de tous les pixels composant le blob et w_{2D} la largeur, h_{2D} la hauteur et a_{2D} le nombre de pixel du blob.

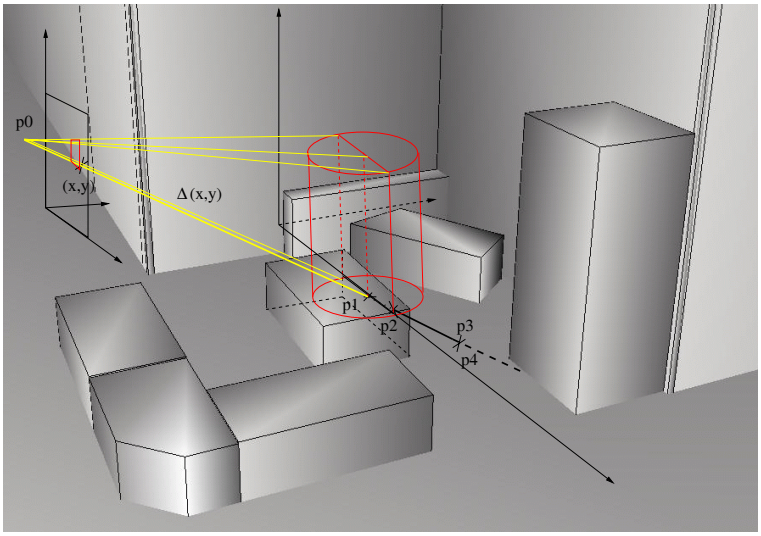


FIG. 4.1 – *Exemple d'application de $\mathcal{P}(O_t,x,y,w,h)$. O_t est le modèle de scène du "coin café" dont les éléments sont représentés en dégradés de gris. (X_c,Y_c,Z_c) est représenté par une croix notée P0. Le référentiel image est représenté par un rectangle noir en haut à gauche. La droite $\Delta(x,y)$ est représentée par une portion de droite en jaune épais. Les points P1, P2, P3 et P4 sont les points d'intersection entre $\Delta(x,y)$ et le modèle de scène (P1 et P2 sont deux points d'intersection avec la table centrale et P3 et P4 sont deux points d'intersection avec le sol). Le rectangle rouge à l'intérieur du plan image représente un rectangle de hauteur h et de largeur w dont le milieu du point bas est p_0 de coordonnées (x,y) et le cylindre rouge au centre de la scène représente $\mathcal{P}(O_t,x,y,w,h)$*

Définissons le problème de regroupement de blobs comme un problème de partitionnement. On dit que la partition $Q_t = \{q_{1,t},...,q_{k,t}\} = \{\{b_{\alpha,t},...,b_{\beta,t}\},...,\{b_{\delta,t},...,b_{\gamma,t}\}\}$ est la solution de $C(B_t)$ **si et seulement si** :

1. **SI** 2 blobs $b_{i,t}$ et $b_{j,t}$ proviennent de la projection du même humain **ET** $b_{i,t} \in q_{\lambda,t}$ **ALORS** $b_{j,t} \in q_{\lambda,t}$.
2. **SI** un blob $b_{i,t}$ provient d'une quelconque forme de bruit (bruit de l'image, bruit de la scène, ombres, etc...) **ET** $b_{i,t} \in q_{\lambda,t}$ **ALORS** $|q_{\lambda,t}| = 1$

Avec le premier point, on souhaite regrouper tous les blobs provenant de la projection d'un humain particulier dans le même sous-ensemble de la partition. Avec le second point, on souhaite isoler les blobs de bruit dans des sous-ensemble distincts de la partition.

4.3.2 Résoudre le problème de groupement

On propose de résoudre $C(B_t)$ par un parcours heuristique de l'ensemble de toute les partitions de B_t possibles. On définit ce parcours heuristique par:

$$\begin{cases} e_0, \text{ l'état initial du parcours, est la partition composée de } k \text{ singletons} \\ e_{i+1} \in A(e_i) \text{ tel que } \kappa(e_{i+1}) = MIN_{e_k \in A(e_i)}(\kappa(e_k)) \wedge \kappa(e_{i+1}) < \kappa(e_i) \\ e_f, \text{ l'état final du parcours} \end{cases}$$

I.e. e_f, l'état final, est l'etat a partir duquel il n'est pas possible de trouver une autre partition parmi $A(e_f)$ ayant une meilleure évaluation.

e_{i+1} est la partition qui minimise l'heuristique κ parmi l'ensemble de toutes les partitions admissibles $A(e_i)$. $A(e_i)$ est l'ensemble des partitions obtenues à partir de e_i en permutant un unique blob d'un sous-ensemble vers un autre. La valeur de l'heuristique κ est défini par:

$$\begin{aligned} \kappa(e_i) &= \sum_{q_{k,t} \in e_i} f(q_{k,t}) \\ f(q_{k,t}) &= \begin{cases} \lambda & \text{SI} \quad q_{k,t} = \emptyset \\ \frac{\Delta_m(q_{k,t})}{d(q_{k,t})} & \text{SI} \quad q_{k,t} \neq \emptyset \end{cases} \end{aligned}$$

où $\Delta_m(q_{k,t})$ est la distance dans $\mathbb{R}^4$ d'un ensemble de blobs $q_{k,t}$ à un *a priori* modèle d'humain défini par $(w^m_{2D},h^m_{2D},W^m_{3D},H^m_{3D})$ et $d(q_{k,t})$ est la densité de pixels mobiles de $q_{k,t}$:

$$\begin{aligned}\Delta_m(q_{k,t}) &= \big[(w^m_{2D} - w_{2D}(q_{k,t}))^2 + (h^m_{2D} - h_{2D}(q_{k,t}))^2 \\ &\quad +(W^m_{3D} - W_{3D}(q_{k,t}))^2 + (H^m_{3D} - H_{3D}(q_{k,t}))^2\big]^{1/2} \\ d(q_{k,t}) &= \frac{a_{2D}(q_{k,t})}{w_{2D}(q_{k,t}).h_{2D}(q_{k,t})}\end{aligned}$$

Finalement, $C(B_t) = \{q_{k,t} \in e_f \quad | \quad \Delta_m(q_{k,t}) < \Delta_{max}\}$, où Δ_{max} est le seuil définissant la distance admissible maximale au modèle d'humain *a priori*. En d'autre termes, la partition finale e_f est filtrée pour ne conserver que les sous-ensembles dont la distance au modèle est inferieure à Δ_{max}. Nous verrons plus tard l'influence de la valeur de Δ_{max} sur l'état final du parcours.

Notons que :

- Le modèle d'humain *a priori* est obtenu par régression par l'algorithme d'un ensemble d'apprentissage des nuées dynamiques, à k vecteurs dans $\mathbb{R}^4 (k \simeq 10)$.
- $d(q_{k,t})$ est raffinée par une fonction gamma, afin de contrôler l'influence de la densité dans le calcul de l'heuristique κ En définitive,

$$d(q_{k,t}) = e^{\frac{1}{\gamma} ln(\frac{a_{2D}(q_{k,t})}{w_{2D}(q_{k,t}) h_{2D}(q_{k,t})})}$$

- λ n'est pas constant sur le parcours heuristique; en fait sa valeur est calculée à chaque e_{i+1} comme suit:

$$\lambda = \frac{1}{k} \sum_{q_{k,t} \in e_i} \Delta_m(q_{k,t})$$

4.3.3 *Analyse de la solution*

L'influence de Δ_{max} sur l'exactitude des résultats est importante. Plus Δ_{max} est grand, plus le nombre de sous-ensembles de l'état final sera grand. A l'opposé, plus Δ_{max} est petit, plus le nombre de sous-ensembles de l'état final est réduit.

En d'autres termes, si Δ_{max} est grand un sous-ensembles, même loin du modèle d'humain, sera considérées en tant qu'humain. Ainsi le nombre d'humains non reconnus (reconnaissance ratée) sera petit et le nombre d'entités non-humaines considérés comme humains (fausse reconnaissance) sera fort. Inversement, si Δ_{max} est petit alors le nombre de reconnaissances ratées sera haut et le nombre de fausses reconnaissances sera petit.

L'influence de la valeur de γ dans la fonction gamma de la densité est également importante. Plus γ est petit, plus l'influence de la densité sur l'heuristique κ est grande et plus le

nombre de sous-ensembles avec plus d'un blob sera élevé dans l'état final. On montre qu'à la limite ($\gamma \to 0$), l'état final est égal à l'état initial ($e_0 = e_f$).

Plus γ est grand, plus l'influence de la densité sur l'heuristique est faible et plus le nombre de sous-ensembles vides sera élevé dans l'état final. On montre qu'à la limite ($\gamma \to \infty$), l'état final est une partition composée $b - 1$ de sous-ensembles vides et d'un sous-ensemble avec tous les blobs.

La complexité dépend de deux points. Le premier point est le coût du passage d'un état e_i au prochain e_{i+1} et le deuxième point est le nombre f d'états e_i jusqu'à l'état final e_f. Le coût du passage d'un état au prochain est $|B_t|^2 = b^2$. Ainsi la complexité de notre approche est fb^2. Nous ne pouvons pas strictement montrer que notre approche converge dans tous les cas possibles.

Nous pouvons seulement prouver un certain nombre de propriétés qui nous permettent de dire que cette heuristique préfère certaines bonnes partitions à d'autres partitions moins bonnes. La figure 4.2 illustre un de ces critères de convergence locale.

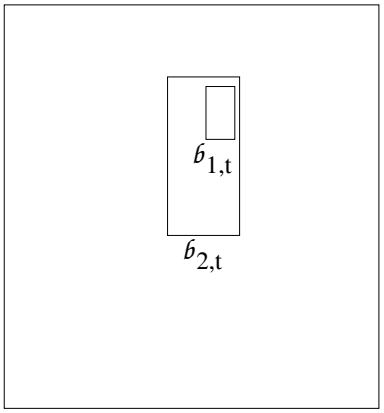


FIG. 4.2 – *Illustration d'un cas de convergence locale avec* $B_t = \{b_{1,t}, b_{2,t}\}$. *Dans ce cas, la partition* $\{\{\},\{b_{1,t},b_{2,t}\}\}$ *est toujours préférée à* $\{\{b_{1,t}\},\{b_{2,t}\}\}$ *car* $\kappa(\{\{\},\{b_{1,t},b_{2,t}\}\}) > \kappa(\{\{b_{1,t}\},\{b_{2,t}\}\})$

4.4 Résultats

Nous présentons dans cette section quelques résultats de notre approche. Dans table 4.1, nous présentons les résultats de la méthode par rapport à la caractérisation de la solution donnée en 4.3. C.-a-d. nous considérons comme erreur chaque blob mal classé. Si un blob correspondant à un bruit est dans un sous-ensemble de la partition correspondant à un

humain, ce blob est mal classé. Si un blob correspondant à une certaine partie d'un humain n'est pas dans le sous-ensemble de la partition correspondant au reste de cet humain, ce blob est mal classé.

référence de la vidéo	nombre de frames	nombre de blobs	nombre de mal classés	% de mal classés
st1-23	190	951	17	1.78
c02-2	535	4268	136	3.18
mc2-17	197	6073	290	4.77
va2-7	55	451	25	5.54
va2-4	340	5888	364	6.18
B008	570	5840	365	6.25
c07-2	137	590	49	8.30
mc1-22	153	1140	106	9.29
va2-6	322	2363	275	11.63
TOTAL	2499	27564	1627	5.90

TAB. 4.1 – *Résultats de la méthode en terme de mauvaises classifications*

Nous pouvons donner deux explications à ces erreurs. Un humain entrant ou quittant la scène est seulement partiellement visible et ne correspond pas à notre *a priori* modèle d'humain. La deuxième source d'erreurs est l'ensemble des petits blobs d'ombres qui sont souvent détectés prés d'un humain. Les blobs correspondants sont souvent mal classés même si ces fausses classifications ne changent pas vraiment la description de l'humain correspondant.

Les tables 4.2 et 4.3 présentent les résultats de la méthode en terme de ce que l'expert attend. C'est à dire une description des humains présents dans la scène à chaque instant. Cette description peut être jugée à deux niveaux. Le premier niveau est le nombre de sous-ensemble non vide de la partition finale (c-a-d le nombre d'humains). Le deuxième niveau est la précision de la description de chaque humain. Le premier niveau est très important parce qu'un faux nombre de personnes (sous-ensembles non vides) dans la partition finale peut sérieusement altérer les résultats de l'étape de mise en correspondance temporelle.

La table 4.2 présente les résultats en termes de reconnaissances ratées (faux négatifs); c'est à dire quand il n'y a aucun sous-ensemble dans la partition correspondant à un humain donné. La table 4.3 présente les résultats en termes de fausse reconnaissance (faux positif); c'est à dire quand il n'y a aucune correspondance humaine à un sous-ensemble donné de la partition

référence de la vidéo	nombre d'images	nombre d'humains	nombre de ratés	% de raté
c02-2	535	906	3	0.33
st1-23	190	180	1	0.55
c07-2	137	120	1	0.83
mc2-17	197	354	4	1.12
mc1-22	153	217	5	2.30
B008	570	782	21	2.68
va2-7	55	138	4	2.89
va2-6	322	345	16	4.63
va2-4	340	304	65	21.38
TOTAL	2499	3346	120	3.58

TAB. 4.2 – *Résultats de la méthode en termes de reconnaissances ratées*

La seule source d'erreur causant des reconnaissances ratées est quand un sous-ensemble correspondant à un humain donné est trop loin de notre modèle. C'est souvent le cas quand un sous-ensemble correspond à plus d'une personne (groupes, occlusion, etc.).

référence de la vidéo	nombre d'images	nombre humains	nombre de faux	% de faux
mc2-17	197	354	0	0.00
B008	570	782	0	0.00
va2-7	55	138	0	0.00
c02-2	535	906	1	0.11
va2-6	322	345	1	0.28
st1-23	190	180	1	0.55
c07-2	137	120	1	0.83
va2-4	340	304	10	3.28
mc1-22	153	217	10	4.60
TOTAL	2499	3346	24	0.71

TAB. 4.3 – *Résultats de la methode en termes de fausses reconnaissances*

La cause principale de fausses reconnaissance est un excès de blobs correspondant à du bruit. Nous pouvons voir que les résultats en termes de fausses reconnaissance sont meilleurs que ceux en termes de reconnaissances ratées. Ce point est important dans la mesure où les problèmes provoqués par les fausses reconnaissances et par les reconnaissances ratées sont très différents pour le problème de mise en correspondance temporelle et la méthode

décrite dans le chapitre suivant est plus robuste aux reconnaissances ratées qu'aux fausses reconnaissance.

4.5 Conclusion

Nous avons présenté, dans ce chapitre, la méthode utilisée pour la reconnaissance de personnes. Cette méthode est basé sur la recherche d'une partition de l'ensemble des blobs obtenus par différence d'images.

L'accent a été mis dans cette méthode sur l'idée de réunir, dans le cadre générique de la recherche heuristique, l'ensemble des informations susceptibles de s'affranchir des contraintes "Real-World", plutôt que d'essayer de résoudre le problème sur un plan purement image.

L'information utilisé est principalement de deux natures: de la connaissance (information statique) et des déductions (information dynamique).

La connaissance mise en jeu est intégré à l'heuristique de recherche de la partition de l'ensemble des blobs. C'est d'une part le modèle *a priori* de la scène, d'autre part un modèle hybride d'humain 2D/3D/densité.

Les déductions utilisés sont les résultats globaux du suivi de personne. Cette information est intégré au calcul de l'image de la scène idéalement vide.

Le premier avantage de l'approche est le faible coût en temps de calcul dans la mesure où cette méthode ne nécessite pas la gestion d'un modèle complexe de la scène vide. Le second avantage de la méthode est la robustesse due à l'utilisation d'information externe.

L'inconvénient majeur de la méthode est sa fragilité par rapport au choix des valeurs des deux principaux paramètres que sont α, la valeur de seuillage, et β, la valeur du pourcentage d'intégration.

Chapitre 5 Mise en correspondance temporelle

L'objectif de ce chapitre est de détailler la méthode utilisée pour résoudre le problème de la mise en correspondance temporelle.

Ces travaux ont été réalisés avec la collaboration de R. Stahr.

Le but de cette méthode est de fournir une description temporelle de l'évolution des humains de la scène. C'est à dire fournir pour chaque humain, un ensemble de descriptions de son état (position, vitesse, taille, etc ...). En d'autres termes, l'objectif de cette méthode est de fournir à chaque instant, c'est à dire à chaque image, un ensemble de descriptions des humains présents ainsi que les relations qui les lient aux descriptions de l'instant précédent.

Les deux difficultés majeures de la résolution de ce problème sont d'une part, le caractère lacunaire des informations en entrée et d'autre part la nature intrinsèque du problème. En effet, les données d'entrée de ce problème étant le résultat de la reconnaissance de personnes, ces informations ne supposent pas qu'à un unique ensemble de blobs corresponde un unique humain. En effet, un ensemble de blobs issus du groupement des blobs peut être la projection de plusieurs humains. De plus ces données peuvent être erronées: certains humains peuvent ne pas être détectés. Au même titre qu'à une détection peut ne correspondre aucun humain.

La seconde difficulté du problème réside dans sa nature même. Si ce problème peut se ramener à savoir si à un humain donné à un instant donné correspond un autre humain à l'instant d'avant, il faut ajouter à cela un ensemble de cas particuliers, rendant ce simple calcul de similarité fragile voir inefficace. Le cas des entrées/sorties d'humains dans la scène ainsi que les problèmes d'occultations sont des exemples de cas particuliers.

La solution proposée pour résoudre le problème de mise en correspondance temporelle consiste en le calcul d'un diagnostic optimal de l'évolution du système entre deux frames. A partir d'un ensemble de primitives, appelées fonctions de mise en correspondance, on construit, à chaque frame, un ensemble de diagnostics possibles dont on garde le plus vraisemblable. Ce diagnostic optimal conditionne alors l'évolution de la description du système.

Après avoir défini, dans la section 5.1, la notion de fonction de mise en correspondance et, dans la section 5.3 la notion de diagnostic, nous présenterons les détails de la méthode retenue comme solution du problème de mise en correspondance, dans la section 5.6. Des résultats de la méthode seront détaillés dans la section 5.9.

5.1 Mise en correspondance temporelle

Définissons la notion de fonction de correspondance Φ_j comme une fonction de soit P_{t-1}, soit Q_t, soit $P_{t-1} \times Q_t$ dans $P_t \times T_t$ représentant une certaine configuration impliquant un humain $h \in \mathcal{H}_{t-1} \cap \mathcal{H}_t$. Les ensembles Q_t,P_{t-1},P_t et T_t sont définis par:

$Q_t = \{q_{1,t},...,q_{m,t}\} = C(B_t)$ est la solution du problème de groupement de blobs (voir chapitre 4) (i.e. une partition de l'ensemble des blobs). On notera m le cardinal de Q_t. Dans le même esprit que pour les éléments de P_t, on associe à chaque $q \in Q_t$ deux formes de représentations *box*(q) et *hull*(q) où *box*(q) est un rectangle dans le référentiel image défini par $x_{2D}(q)$,$y_{2D}(q)$,$w_{2D}(q)$ et $h_{2D}(q)$ et *hull*(q) est un cylindre dans le référentiel scène obtenu par projection $\mathcal{P}$ (O_t,$x_{2D}(q)$,$y_{2D}(q)$,$w_{2D}(q)$,$h_{2D}(q)$ (voir 4.2) . On étend de plus, la définition de l'opérateur $\equiv$, décrit dans la section 3 pour un sommet $p \in \bar{P}_t$, à un sous-ensemble $q \in Q_t$ à ceci prêt, que l'on permet que deux humains h_1,$h_2 \in \mathcal{H}_t$ puissent correspondre à un même ensemble de blobs $q \in Q_t$.

P_t (resp. P_{t-1}) est l'ensemble des sommet de $\bar{G}_t$ de type *person* à l'instant t (resp. $t-1$). On notera n le cardinal de P_{t-1}.

$T_t = \{t(p_{\alpha,t-1},p_{\beta,t}),...,t(p_{\gamma,t-1},p_{\delta,t})\}$ est l'ensemble des arcs de $\bar{G}_t$ entre un élément de P_{t-1} et un élément de P_t.

Nous définissons dans la suite un ensemble de 7 fonctions Φ_j correspondant aux configurations possibles de correspondance temporelle. Dans chacun de ces 7 cas, nous spécifierons le résultat de la fonction Φ_j comme un couple $(p_{k,t},t(p_{i,t-1},p_{k,t}))$ avec $p_{k,t} \in P_t$ et $t(p_{i,t-1},p_{k,t}) \in T_t$. Nous spécifierons, en outre, les conditions d'application de chaque fonction Φ_j, en terme de correspondance entre le phénomène physique et les différents types de représentations dont nous disposons. Nous spécifierons enfin, les moyens d'évaluer dans quelles mesures certaines conditions sont vérifiées ou non. On notera $e(\Phi_j) \in [0,1]$ l'évaluation de Φ_j.

Dans le cas général, $\Phi_l(p_{i,t-1},q_{j,t}) = (p_{k,t},t(p_{i,t-1},p_{k,t}))$ tel que:

Les 7 fonctions de correspondance sont: MATCH, LOST, NOISE, ENTRY, EXIT, HIDDEN et APPEARS et sont définies comme suit:

- Φ_1: MATCH$(p_{i,t-1},q_{j,t}) = (p_{k,t},t(p_{i,t-1},p_{k,t}))$ **si et seulement si** $p_{i,t-1}$ et $q_{j,t}$ correspondent au même humain à l'instant $t-1$ et t. On détermine si les conditions de ce cas sont vérifiées par le calcul de similarité entre $p_{i,t-1}$ et $q_{j,t}$: $e(\text{MATCH}(p_{i,t-1},q_{j,t})) = \mathcal{S}$ $(p_{i,t-1},q_{j,t})$ où $\mathcal{S}$ est la fonction de similarité détaillée dans la section 5.2.

MATCH$(p_{i,t-1},q_{j,t}) = (p_{k,t},t(p_{i,t-1},p_{k,t}))$ **si et seulement si**
$\exists h \in \mathcal{H}_{t-1} \cap \mathcal{H}_t$ tel que $p_{i,t-1} \equiv h \wedge q_{j,t} \equiv h$

$$
\begin{array}{lcl}
name(p_{k,t}) & = & \begin{cases} name(p_{i,t-1}) & \text{SI} \quad t(p_{i,t-1},p_{k,t}) \in T_t \\ \text{un nouveau } name & \text{SI} \quad t(p_{i,t-1},p_{k,t}) \notin T_t \end{cases} \\
time(p_{k,t}) & = & t \\
type(p_{k,t}) & = & person \\
box(p_{k,t}) & = & box(q_{l,t}) \\
hull(p_{k,t}) & = & hull(q_{l,t}) \\
velocity(p_{k,t}) & = & \text{un vecteur vitesse 3D estimé} \\
properties(p_{k,t}) & = & \emptyset \\
references(p_{k,t}) & = & \emptyset
\end{array}
$$

- Φ_2: LOST$(p_{i,t-1}) = (p_{k,t}, t(p_{i,t-1},p_{k,t}))$ **si et seulement si** il n'existe aucun $q_{j,t} \in Q_t$ correspondant à l'humain décrit par $p_{i,t-1}$. La signification de $p_{k,t}$ est expliquée à la fin de la section. Le cas est toujours possible alors $e(\text{LOST}(p_{i,t-1})) = 1$.

$$\boxed{\begin{array}{c}\text{LOST}(p_{i,t-1}) = (p_{k,t}, t(p_{i,t-1},p_{k,t})) \textbf{ si et seulement si} \\ \exists h \in \mathcal{H}_{t-1} \cap \mathcal{H}_t \quad \text{tel que} \quad p_{i,t-1} \equiv h \wedge \forall q \in Q_t : q \not\equiv h\end{array}}$$

- Φ_3: NOISE$(q_{j,t}) = (\emptyset,\emptyset)$ **si et seulement si** $q_{j,t}$ ne correspond à aucun humain. Ce cas est toujours possible alors $e(\text{NOISE}(q_{j,t})) = 1$

$$\boxed{\text{NOISE}(q_{j,t}) = (\emptyset,\emptyset) \textbf{ si et seulement si } \forall h \in \mathcal{H}_t : q_{j,t} \not\equiv h}$$

- Φ_4: EXIT$(p_{i,t-1}) = (\emptyset,\emptyset)$ **si et seulement si** l'humain correspondant au sommet $p_{i,t-1}$ ayant quitté la scène à l'instant t, il n'existe aucun $q_{j,t}$ correspondant. Les conditions d'application de cette fonction sont évaluées en comparant la localisation de $p_{i,t-1}$ dans la scène avec un ensemble de zones prédéfinies de la scène. En d'autres termes, $e(\text{EXIT}(p_{i,t-1})) = 1$ si le prédicat IsInIO$(p_{i,t-1})$ = TRUE.

$$\boxed{\begin{array}{c}\text{EXIT}(p_{i,t-1}) = (\emptyset,\emptyset) \textbf{ si et seulement si} \\ \exists h \in \mathcal{H}_{t-1} \setminus \mathcal{H}_t \quad \text{tel que} \quad p_{i,t-1} \equiv h\end{array}}$$

- Φ_5: ENTRY$(q_{j,t}) = (p_{k,t},\emptyset)$ **si et seulement si** l'humain correspondant à $q_{j,t}$ n'était pas présent dans la scène à l'instant $t-1$. On évalue les conditions de ce cas en comparant la localisation de $q_{j,t}$ avec un ensemble de zones prédéfinies de la scène. En d'autres termes, $e(\text{ENTRY}(q_{j,t})) = 1$ si le prédicat IsInIO$(q_{j,t})$ =TRUE.

$$\boxed{\begin{array}{c}\text{ENTRY}(q_{j,t}) = (p_{k,t},\emptyset) \textbf{ si et seulement si} \\ \exists h \in \mathcal{H}_t \setminus \mathcal{H}_{t-1} \quad \text{tel que} \quad q_{j,t} \equiv h\end{array}}$$

- Φ_6: HIDDEN$(p_{i,t-1},q_{j,t}) = (p_{k,t}, t(p_{i,t-1},p_{k,t}))$ **si et seulement si** l'humain correspondant au sommet $p_{i,t-1}$ est occulté par un autre humain décrit par $q_{j,t}$. Notons qu'une instance de HIDDEN$(p_{i,t-1},q_{j,t})$ est donc forcement accompagnée d'une instance de MATCH$(p_{i',t-1},q_{j,t})$ correspondant à l'humain occultant. Les deux sommets $p_{i,t-1}$ et $p_{i',t-1}$ sont alors mis en correspondance avec le même ensemble de blobs

$q_{j,t}$ de l'instant t et cette configuration perdurera tant que l'un des deux humains occultera le second. On évalue les conditions de ce cas en comparant la localisation de $p_{i,t-1}$ par rapport à $q_{j,t}$. En d'autres termes, $e(\text{HIDDEN}(p_{i,t-1},q_{j,t})) = 1$ si le prédicat $\text{OCCLUDES}(p_{i,t-1},q_{j,t})) = \text{TRUE}$.

$$\text{HIDDEN}(p_{i,t-1},q_{j,t}) = (p_{k,t},t(p_{i,t-1},p_{k,t})) \textbf{ si et seulement si}$$
$$\exists h_1,h_2 \in \mathcal{H}_{t-1} \cap \mathcal{H}_t \text{ tel que}$$
$$p_{i,t-1} \equiv h_1 \wedge p_{i',t-1} \equiv h_2 \quad \wedge \quad q_{j,t} \equiv h_1 \wedge q_{j,t} \equiv h_2$$

– Φ_7: $\text{APPEARS}(p_{i,t-1},q_{j,t}) = (p_{k,t},\emptyset)$ **si et seulement si** l'humain correspondant à l'ensemble de blobs $q_{j,t}$ était caché à l'instant $t-1$ par un autre humain décrit par $p_{i,t-1}$. Notons que ce cas ne s'applique que lorsque deux humains entrent dans la scène au même instant; l'un des deux occultant le second. Ce cas est évalué en comparant les localisations de $p_{i,t-1}$ et $q_{j,t}$. En d'autres termes, $e(\text{APPEARS}(p_{i,t-1},q_{j,t})) = 1$ si le prédicat $\text{OCCLUDES}(p_{i,t-1},q_{j,t})) = \text{TRUE}$.

$$\text{APPEARS}(p_{i,t-1},q_{j,t}) = (p_{k,t},\emptyset) \textbf{ si et seulement si}$$
$$\exists h \in \mathcal{H}_{t-1} \cap \mathcal{H}_t \text{ tel que } q_{j,t} \equiv h \wedge \forall p \in P_{t-1} : p \not\equiv h$$

Note:

Nous avons dit que les attributs *hull* et *box* de $p_{k,t}$ résultant d'une fonction de correspondance étaient calculés à partir des attributs de l'ensemble de blobs $q \in Q_t$. Hors, dans le cas de $\text{LOST}(\Phi_2)$, il n'y a aucun q. Dans ce cas, les attributs *hull* et *box* de $p_{k,t}$ doivent être extrapolés à partir de *hull* et *box* de $p_{i,t-1}$. Plusieurs stratégies sont alors possibles. Une stratégie, appelé stratégie stationnaire, consiste à garder les attributs de $p_{i,t-1}$, c'est à dire, $hull(p_{k,t}) = hull(p_{i,t-1})$ et $box(p_{k,t}) = box(p_{i,t-1})$. Une autre stratégie, appelée stratégie prédictive, consiste à estimer les $hull(p_{k,t})$ (resp. $box(p_{k,t})$) de $p_{k,t}$ par filtrage de Kalman des précédents $hull(p_{k,t-j})$ (resp. $box(p_{k,t-j})$).

Le prédicat $\text{IsInIO}(p_{i,t-1})$ est défini par rapport aux éléments de O_t et le prédicat $\text{OCCLUDES}(p_{i,t-1},q_{j,t})$ est défini en fonction de la localisation relative de $p_{i,t-1},q_{j,t}$ et la caméra.

5.2 *Calcul de similarité*

On définit la fonction de similarité $\mathcal{S}$ entre un ensemble de blobs $q \in Q_t$ et un sommet p par:

$$\begin{aligned}\mathcal{S}(p,q) &= \sum_i \omega_{i,2D} e^{\frac{-d_{2D}(box(q),D_i(box(p)))}{\sigma_{i,2D}}} + \sum_i \omega_{i,3D} e^{\frac{-d_{3D}(hull(q),D_i(hull(p)))}{\sigma_{i,3D}}} \\ \sum_i \omega_i &= 1\end{aligned}$$

où d_{2D} (resp. d_{3D}) est la distance 2D (resp. distance 3D) entre $box(q)$ (resp. $hull(q)$) et $D_i(box(p))$ (resp. $D_i(hull(p))$) où D_i est le i^{th} mode d'estimation (moyen, médian, Kalman [73, 76, 81, 98, 102], ...).

5.3 Problème de diagnostic

Soit $I(Q_t,P_{t-1})$ le problème défini par $I(Q_t,P_{t-1}) = \{\Phi_\alpha(p_{\beta,t-1},q_{\gamma,t}),..., \Phi_\delta(p_{\mu,t-1})\}$ tel que $\Phi_\alpha(p_{\beta,t-1},q_{\gamma,t}) \cup ... \cup \Phi_\delta(p_{\mu,t-1}) = M(Q_t,P_{t-1})$. En d'autre termes, le problème I est de trouver un ensemble d'applications des fonctions de $\{\Phi_1,...,\Phi_7\}$ sur les éléments de P_{t-1} et Q_t tel que l'union des résultats de ces applications est la solution de $M(Q_t,P_{t-1})$. Nous appellerons "diagnostic" tout ensemble d'applications des fonctions $\{\Phi_1,...,\Phi_7\}$ sur les éléments de P_{t-1} et Q_t.

Nous définissons alors l'évaluation d'un diagnostic $\mathcal{I}_t = \{\Phi_\alpha(p_{\beta,t-1},q_{\gamma,t}),..., \Phi_\delta(p_{\mu,t-1})\}$ comme la somme pondérée des évaluations des applications des fonctions de correspondance de $\mathcal{I}_t$. C'est à dire que:

$$e(\mathcal{I}_t) = \lambda_\alpha e(\Phi_\alpha(p_{\beta,t-1},q_{\gamma,t})) + ... + \lambda_\delta e(\Phi_\delta(p_{\mu,t-1}))$$

où $\{\lambda_1,...,\lambda_7\}$ représente les préférences associées à chaque cas, E.g. On préfèrera toujours obtenir un maximum de MATCH possible (i.e. $\forall k \neq 1 : \lambda_1 > \lambda_k$) et l'on préfèrera toujours obtenir le moins de LOST et NOISE possible (i.e. $\forall k \in \{1,2,3,4,5\} : \lambda_6 < \lambda_k \wedge \lambda_7 < \lambda_k$).

La figure 5.1 est un exemple d'un problème $I(Q_t,P_{t-1})$ à l'instant $t = 11$ avec $Q_t = \{q_{1,11},q_{2,11},q_{3,11}\}$ et $P_{t-1} = \{p_{1,10},p_{2,10},p_{3,10}\}$. Dans ce cas:

$$
\begin{aligned}
I(Q_t,P_{t-1}) &= \{\text{MATCH}(p_{3,10},q_{3,11}),\text{MATCH}(p_{1,10},q_{2,11}),\\
&\quad \text{HIDDEN}(p_{2,10},q_{2,11}),\text{ENTRY}(q_{1,11})\}\\
\text{MATCH}(p_{3,10},q_{3,11}) &= (p_{1,11},t(p_{3,10},p_{1,11}))\\
\text{MATCH}(p_{1,10},q_{2,11}) &= (p_{2,11},t(p_{1,10},p_{2,11}))\\
\text{HIDDEN}(p_{2,10},q_{2,11}) &= (p_{3,11},t(p_{2,10},p_{3,11}))\\
\text{ENTRY}(q_{1,11}) &= (p_{4,11},\emptyset)
\end{aligned}
$$

I.e. la solution du problème de mise en correspondance temporelle est $M(Q_t,P_{t-1}) = (P_t,T_t)$ avec $P_t = \{p_{1,11},p_{2,11},p_{3,11},p_{4,11}\}$ et $T_t = \{t(p_{3,10},p_{1,11}),t(p_{1,10},p_{2,11}),t(p_{2,10},p_{3,11})\}$.

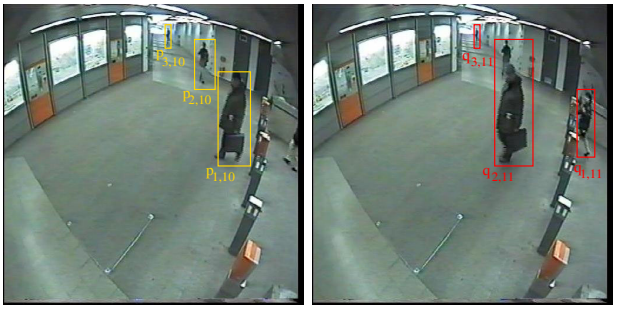


FIG. 5.1 – *Exemple de problème* $I(Q_t,P_{t-1})$ *à* $t = 11$

Nous définissons maintenant la notion d'un diagnostic valide ou invalide. Nous disons qu'un diagnostic est invalide quand deux fonctions de correspondance ou plus ont des conditions incompatibles. Par exemple {EXIT(a), HIDDEN(a)} est un diagnostic invalide parce que la première fonction s'applique quand le sommet $a \equiv h$ avec h non présent dans la scène à t et la deuxième fonction s'applique quand le sommet $a \equiv h$ avec h présent (mais occulté) dans la scène à t. Nous définissons la notion d'un diagnostic valide comme n'importe quel diagnostic non-invalide.

5.4 Théorème 1

Soit $\mathcal{I}_t$ la solution de $I(Q_t,P_{t-1})$. $\mathcal{I}_t$ est un diagnostic valide **si et seulement si** $\forall a,c \in P_{t-1}$,$a \neq c$ et $\forall b,d \in Q_t,b \neq d$, aucun des cas suivants n'existe

1. $\mathcal{I}_t = \{...,\text{MATCH}(a,b),\text{MATCH}(a,d),...\}$

2. $\mathcal{I}_t = \{...,\text{MATCH}(a,b),\text{HIDDEN}(a,d),...\}$
3. $\mathcal{I}_t = \{...,\text{MATCH}(a,b),\text{LOST}(a),...\}$
4. $\mathcal{I}_t = \{...,\text{MATCH}(a,b),\text{EXIT}(a),...\}$
5. $\mathcal{I}_t = \{...,\text{HIDDEN}(a,b),\text{LOST}(a),...\}$
6. $\mathcal{I}_t = \{...,\text{HIDDEN}(a,b),\text{EXIT}(a),...\}$
7. $\mathcal{I}_t = \{...,\text{LOST}(a),\text{EXIT}(a),...\}$
8. $\mathcal{I}_t = \{...,\text{MATCH}(a,b),\text{MATCH}(c,b),...\}$
9. $\mathcal{I}_t = \{...,\text{MATCH}(a,b),\text{APPEARS}(c,b),...\}$
10. $\mathcal{I}_t = \{...,\text{MATCH}(a,b),\text{NOISE}(a),...\}$
11. $\mathcal{I}_t = \{...,\text{MATCH}(a,b),\text{ENTRY}(a),...\}$
12. $\mathcal{I}_t = \{...,\text{APPEARS}(a,b),\text{NOISE}(b),...\}$
13. $\mathcal{I}_t = \{...,\text{APPEARS}(a,b),\text{ENTRY}(b),...\}$
14. $\mathcal{I}_t = \{...,\text{NOISE}(b),\text{ENTRY}(b),...\}$

5.5 Preuve

On peut prouver ce théorème en trois points:

1. Premièrement on prouve que chacun des 14 cas est un diagnostic invalide.
2. Puis on montre que le nombre d'un diagnostic valide est fini.
3. Enfin on montre que seuls ces 14 cas sont des diagnostics invalides.

Nous ne démontrerons, pour la première étape, que le dernier cas de figure, c'est à dire $\mathcal{I}_t = \{...,\text{NOISE}(b),\text{ENTRY}(b),...\}$. Les preuves des 13 autres cas sont similaires.

A partir de la définition des fonctions de correspondance NOISE et ENTRY, on sait que NOISE(b) ne peut s'appliquer que si $\forall h \in \mathcal{H}_t : b \not\equiv h$ et ENTRY(b) et NOISE(b) ne peut s'appliquer que si $\exists h \in \mathcal{H}_t \setminus \mathcal{H}_{t-1}$ tel que $b \equiv h$. Ces deux conditions sont à l'évidence incompatibles, donc NOISE(b) et ENTRY(b) ne peuvent être appliquées en même temps. En d'autre termes, $\mathcal{I}_t = \{...,\text{NOISE}(b),\text{ENTRY}(b),...\}$ est un diagnostic invalide.

Pour prouver le second point, on remarquera que pour un $p \in P_{t-1}$ donné (resp. $q \in Q_t$), p ne peut être impliqué dans deux fonctions de correspondance parmi MATCH, HIDDEN, LOST ou EXIT (resp. MATCH, APPEARS, NOISE ou ENTRY) dans le même diagnostic. Alors, a étant le nombre d'instances de MATCH, n le cardinal de P_{t-1} et m celui de Q_t, le cardinal d'un diagnostic valide est $a + (m - a) + (n - a)$. Si $a = 0$, le cardinal d'un diagnostic valide est $n + m$. Si $a =$MIN(m,n), le cardinal d'un diagnostic valide est

$n+m-\text{MIN}(n,m) = \text{MAX}(n,m)$. Donc, pour une valeur quelconque de a, le cardinal d'un diagnostic valide $\mathcal{I} = I(Q_t, P_{t-1})$ est borné par l'intervale $[\text{MAX}(m,n), m+n]$.

La dernière étape de la preuve est effectuée en utilisant le deuxième point. Le fait que le cardinal d'un diagnostic valide soit borné implique que le nombre de diagnostics est fini donc connu. Nous ne voulons pas énumérer tous les diagnostics valides, mais nous certifions que seulement les 14 cas précédents sont invalides.

5.6 Résoudre le problème de diagnostic

Dans cette sous-section, nous proposons un schéma numérique pour résoudre le problème de diagnostic. La solution du problème de diagnostic est le diagnostic ayant la meilleure évaluation $\mathcal{I}$ parmi tous les diagnostics valides. En d'autres termes, le problème de diagnostic est de trouver, parmi tous les diagnostics valides, le diagnostic

$$\begin{aligned}\mathcal{I} = \{&\Phi_{\alpha_1}(p_{\beta_1,t-1}, q_{\gamma_1,t}), \ldots, \Phi_{\alpha_\rho}(p_{\beta_\rho,t-1}, q_{\gamma_\rho,t}),\\ &\Phi_{\zeta_1}(q_{\xi_1,t}), \ldots, \Phi_{\zeta_\sigma}(q_{\xi_\sigma,t}),\\ &\Phi_{\delta_1}(p_{\mu_1,t-1}), \ldots, \Phi_{\delta_\omega}(p_{\mu_\omega,t-1})\}\end{aligned}$$

tel que l'évaluation

$$\begin{aligned}&\lambda_{\alpha_1} e(\Phi_{\alpha_1}(p_{\beta_1,t-1}, q_{\gamma_1,t})) + \ldots + \lambda_{\alpha_\rho} e(\Phi_{\alpha_\rho}(p_{\beta_\rho,t-1}, q_{\gamma_\rho,t})) +\\ &\lambda_{\zeta_1} e(\Phi_{\zeta_1}(q_{\xi_1,t})) + \ldots + \lambda_{\zeta_\sigma} e(\Phi_{\zeta_\sigma}(q_{\xi_\sigma,t})) +\\ &\lambda_{\delta_1} e(\Phi_{\delta_1}(p_{\mu_1,t-1})) + \ldots + \lambda_{\delta_\omega} e(\Phi_{\delta_\omega}(p_{\mu_\omega,t-1}))\end{aligned}$$

soit maximale.

Soit χ l'application de $(P^*_{t-1} \times Q^*_t)$ dans l'ensemble de tous les diagnostics tel que :

SI $n > m$ $(Card(P_{t-1}) > Card(Q_t))$

$$\begin{aligned}P^*_{t-1} &= \{p_1, \ldots, p_n, q^*_1, \ldots, q^*_m, q^{**}_1, \ldots, q^{**}_m, q^{***}_1, \ldots, q^{***}_m, q^{****}_1, \ldots, q^{****}_{2(n-m)}\}\\ Q^*_t &= \{q_1, \ldots, q_m, p^*_1, \ldots, p^*_n, p^{**}_1, \ldots, p^{**}_n, p^{***}_1, \ldots, p^{***}_n\}\end{aligned}$$

SI $n < m$ $(Card(P_{t-1}) < Card(Q_t))$

$$\begin{aligned}P^*_{t-1} &= \{p_1, \ldots, p_n, q^*_1, \ldots, q^*_m, q^{**}_1, \ldots, q^{**}_m, q^{***}_1, \ldots, q^{***}_m\}\\ Q^*_t &= \{q_1, \ldots, q_m, p^*_1, \ldots, p^*_n, p^{**}_1, \ldots, p^{**}_n, p^{***}_1, \ldots, p^{***}_n, p^{****}_1, \ldots, p^{****}_{2(m-n)}\}\end{aligned}$$

SI $m = n$ ($Card(P_{t-1}) = Card(Q_t)$)

$$\begin{aligned} P^*_{t-1} &= \{p_1,\ldots,p_n,q^*_1,\ldots,q^*_m,q^{**}_1,\ldots,q^{**}_m,q^{***}_1,\ldots,q^{***}_m\} \\ Q^*_t &= \{q_1,\ldots,q_m,p^*_1,\ldots,p^*_n,p^{**}_1,\ldots,p^{**}_n,p^{***}_1,\ldots,p^{***}_n\} \end{aligned}$$

ET

$$\chi(x,y) = \begin{cases} \text{MATCH}(p_i,q_j) & \text{SI} & x = p_i & \text{ET} & y = q_j \\ \text{HIDDEN}(p_i,q_j) & \text{SI} & x = p_i & \text{ET} & y = p^*_i \\ \text{APPEARS}(p_i,q_j) & \text{SI} & x = q^*_j & \text{ET} & y = q_j \\ \text{LOST}(p_i) & \text{SI} & x = p_i & \text{ET} & y = p^{**}_i \\ \text{NOISE}(q_j) & \text{SI} & x = q^{**}_j & \text{ET} & y = q_j \\ \text{EXIT}(p_i) & \text{SI} & x = p_i & \text{ET} & y = p^{***}_i \\ \text{ENTRY}(q_j) & \text{SI} & x = q^{***}_j & \text{ET} & y = q_j \\ \emptyset & \text{SINON} & & & \end{cases}$$

où les p_i sont les éléments de P_{t-1}, les q_j sont les éléments Q_t, les q^*_i,q^{**}_j,q^{***}_j,q^{****}_j, p^*_i,p^{**}_i,p^{***}_i,p^{****}_i sont des artefacts de notation.
On représente l'application χ par la matrice $\mathcal{M}_\chi$ montrée en figure 5.2.
De la même manière, soit $e\chi$ l'application de $(P^*_{t-1} \times Q^*_t)$ dans $\mathbb{R}$ tel que $e\chi(x,y) = e(\chi(x,y))$. On représente cette application $e\chi$ par la matrice $e\mathcal{M}_\chi$.

Soit $d\chi$ l'application qui associe à toutes bijections f de P^*_{t-1} dans Q^*_t un diagnostic $d\chi(f)$ défini par:

$$d\chi(f) = \{\chi(x,f(x)) \quad \forall x \in P^*_{t-1}\}$$

5.7 Théorème 2

$d\chi(f)$ est un diagnostic valide pour toute bijection f de P^*_{t-1} dans Q^*_t **ET** à tout diagnostic valide $\mathcal{I}$ correspond une bijection f de P^*_{t-1} dans Q^*_t tel que $d\chi(f) = \mathcal{I}$.

5.8 Preuve

f est une bijection est équivalent à dire qu'il n'existe pas un élément a de P^*_{t-1} tel que $f(a) = b$ et $f(a) = d$ ET il n'existe pas d'élément b de Q^*_t tel que $f(c) = b$ et $f(c) = b$.

Dans le premier cas, nous n'aurons pas de diagnostic comme {..., MATCH(a,b), MATCH(a,d),...} ou {..., MATCH(a,b), HIDDEN(a,d),...} ou {..., MATCH(a,b), LOST(a),...} ou {..., MATCH(a,b),

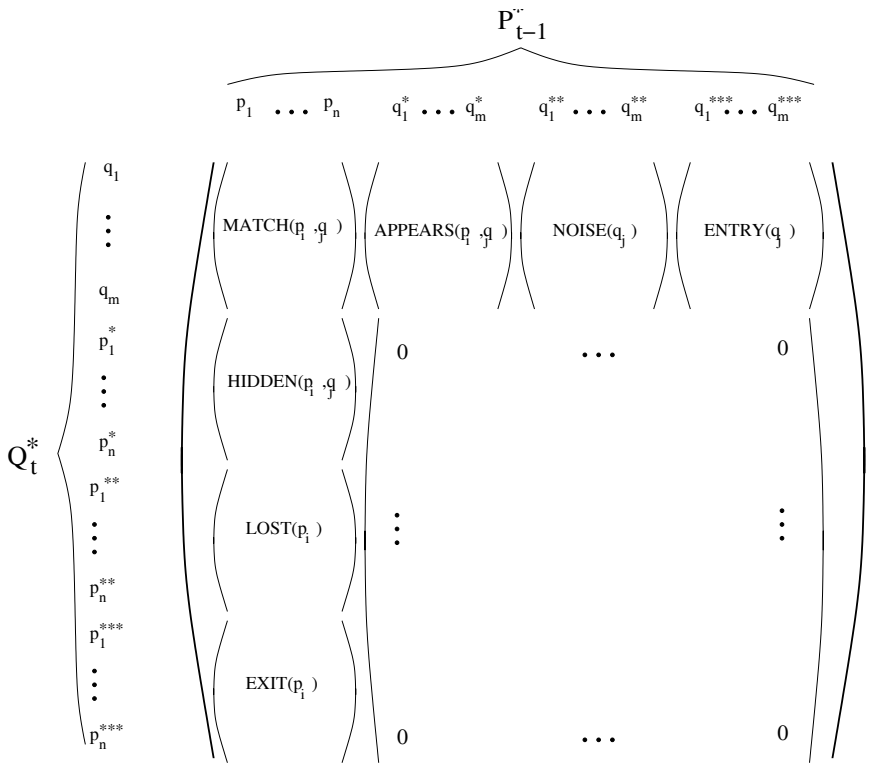


FIG. 5.2 – $\mathcal{M}_\chi$: *La représentation matricielle de l'application* χ *(cas où* $m = n$

EXIT(a),...} ou {..., HIDDEN(a,b), LOST(a),...} ou {..., HIDDEN(a,b), EXIT(a),...} ou {..., LOST(a), EXIT(a),...}. Tous ces diagnostics sont invalides (cf. théorème 1). Dans le second cas, nous n'aurons pas de diagnostic comme: {..., MATCH(a,b), MATCH(c,b),...} , ou {..., MATCH(a,b), APPEARS(c,b),...} ou {..., MATCH(a,b), NOISE(a),...} ou {..., MATCH(a,b), ENTRY(a),...} ou {..., APPEARS(a,b), NOISE(b),...} ou {..., APPEARS(a,b), ENTRY(b),...} ou {...,NOISE(b),ENTRY(b),...}. Tous ces diagnostics sont aussi invalides (cf. théorème 1). C'est à dire que f est une bijection est équivalent au fait que $d\chi(f)$ est un diagnostic valide.

Finalement, le diagnostic valide le mieux évalué est trouvé en cherchant la bijection f dans l'ensemble $\mathcal{B}$ de toutes les bijections de P^*_{t-1} dans Q^*_t tel que :

$$e(d\chi(f)) = \max_{b \in \mathcal{B}} e(d\chi(b))$$

Les lignes et les colonnes de $e\mathcal{M}_\chi$ constituées que de zéros sont enlevées par paire Ainsi si la dimension de cette matrice carrée est d, il y a $d!$ bijections possibles. La bijection optimale est obtenue par une méthode de "Branch and Bound" qui étudie, en général, un nombre de cas bien moindres.

Par exemple, dans le cas présenté en figure 5.1, la matrice $e\mathcal{M}_\chi$ donne:

$$
\begin{array}{c}
 \\
q_1 \\
q_2 \\
q_3 \\
p_1^* \\
p_2^* \\
p_3^* \\
p_1^{**} \\
p_2^{**} \\
p_3^{**} \\
p_1^{***} \\
p_2^{***} \\
p_3^{***}
\end{array}
\begin{array}{c}
\begin{array}{ccccccccccccc}
p_1 & p_2 & p_3 & q_1^* & q_2^* & q_3^* & q_1^{**} & q_2^{**} & q_3^{**} & q_1^{***} & q_2^{***} & q_3^{***}
\end{array} \\
\left(\begin{array}{ccccccccccccc}
10 & 1 & 1 & 0 & 0 & 0 & 10 & 0 & 0 & 50 & 0 & 0 \\
70 & 1 & 1 & 0 & 25 & 0 & 0 & 10 & 0 & 0 & 0 & 0 \\
1 & 5 & 80 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 50 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)
\end{array}
$$

avec $\lambda_1 = 100$, $\lambda_2 = \lambda_3 = 25$, $\lambda_4 = \lambda_5 = 10$ et $\lambda_6 = \lambda_7 = 50$. Après avoir enlever les paires de colonnes/lignes de zéro $e\mathcal{M}_\chi$ devient:

$$
\begin{array}{c}
 \\
q_1 \\
q_2 \\
q_3 \\
p_2^* \\
p_1^{**} \\
p_2^{**} \\
p_3^{**} \\
p_3^{***}
\end{array}
\begin{array}{c}
\begin{array}{cccccccc}
p_1 & p_2 & p_3 & q_2^* & q_1^{**} & q_2^{**} & q_3^{**} & q_1^{***}
\end{array} \\
\left(\begin{array}{cccccccc}
10 & 1 & 1 & 0 & 10 & 0 & 0 & 50 \\
70 & 1 & 1 & 25 & 0 & 10 & 0 & 0 \\
1 & 5 & 80 & 0 & 0 & 0 & 10 & 0 \\
0 & 25 & 0 & 0 & 0 & 0 & 0 & 0 \\
10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 50 & 0 & 0 & 0 & 0 & 0
\end{array}\right)
\end{array}
$$

La bijection optimale est alors $e\chi(p_1,q_2)+ e\chi(p_2,p_2^*)+ e\chi(p_3,q_3)+ e\chi(q_1^{***},q_1)+$ quelques termes égals à 0, avec

$$\begin{aligned} e\chi(p_1,q_2) &= e(\text{MATCH}(p_1,q_2)) \\ e\chi(p_2,p_2^*) &= e(\text{HIDDEN}(p_2,q_2)) \\ e\chi(p_3,q_3) &= e(\text{MATCH}(p_3,q_3)) \\ e\chi(q_1^{***},q_1) &= e(\text{ENTRY}(q_1)) \end{aligned}$$

Ce qui donne:

$$\begin{aligned} \mathcal{I} = \{&\text{MATCH}(p_3,q_3),\text{MATCH}(p_1,q_2), \\ &\text{HIDDEN}(p_2,q_2),\text{ENTRY}(q_1)\} \end{aligned}$$

Ce qui est (en passant) le diagnostic que l'on souhaitait trouver.

5.9 Résultats

Nous présentons dans cette section quelques résultats de notre approche. Dans table 5.1, nous présentons pour chaque séquence vidéo de test les erreurs. Nous considérons comme erreur chaque diagnostic faux, c.-a-d toute frame où le diagnostic ne correspond pas à la vérité de terrain (Ground Truth) qui est un diagnostic fabriqué à la main. Nous distinguons deux types d'erreurs (deux types de diagnostic faux). Le premier type est constitué par les personnes d'une frame (OFP) correspondant à un sommet p du graphe d'interprétation $\bar{G}_t$ relié à aucun autre, c'est à dire sans arc. Ce genre d'erreur n'est pas vraiment un problème, dans la mesure ou elle peut être facilement résolue. Il en va autrement pour le deuxième type d'erreurs, qui peuvent être considérées en tant que vraies erreurs, en ceci qu'elles changent la structure du graphe $\bar{G}_t$.

Video id.	frames	nombre de OFP	nombre de faux	Faux diagnostics
st1-23	190	0	1	frm. 162: NOISE as APPEAR
c02-2	535	17	2	frm. 290: HIDDEN as LOST
				frm. 74: NOISE as ENTER
mc2-17	197	1	1	frm. 18: HIDDEN as EXIT
va2-7	55	0	2	frm. 29: LOST as EXIT
				frm. 33: LOST as EXIT
va2-4	340	6	0	
B008	570	0	0	
c07-2	137	1	1	frm. 99: NOISE as ENTER
mc1-22	153	0	1	frm. 79: NOISE as ENTER
va2-6	322	0	0	

TAB. 5.1 – *Résultats de la méthode en terme de faux diagnostics*

Notons que même s'il n'y a aucun faux diagnostic sur la vidéo VA2-4, la trajectoire des personnes est suffisamment altérée pour que nous ne puissions pas considérer ce résultat particulier comme bon.

Dans la suite, nous détaillerons trois séquences vidéos ayant lieu dans différents environnements, afin d'illustrer les deux erreurs commises, mais aussi les difficultés gérées. Chaque figure se compose de deux parties: à gauche est présentée l'image d'entrée issue du flux vidéo et à droite est présentée notre reconstruction. Les différentes parties de l'environnement ($o_{i,t} \in O_t$) sont représentées en gris, en vert et en orange. Les différentes personnes à t ($p_{j,t} \in P_t$) sont représentées par les cylindres bruns et les pistes ($t \in \bar{T}_t$) sont représentées par les lignes rouges.

Les figures 5.3, 5.4, 5.5, 5.6 et 5.7 illustrent les résultats de notre approche dans une station de métro (vidéo id: VA2-7), les figures 5.8, 5.9, 5.10, 5.11 et 5.12 illustrent les résultats dans une agence bancaire (vidéo id: MC2-17) et les figures 5.13, 5.14, 5.15, 5.16 et 5.17 illustrent des résultats obtenus dans un bureau (vidéo id: C02-2).

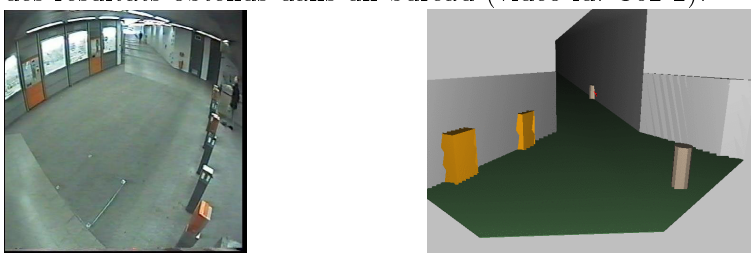


FIG. 5.3 – *Sur le quai du métro de Nuremberg, il y a deux humains h_1 et h_2. h_1 est au fond du couloir et h_2 vient d'entrer par la droite.*

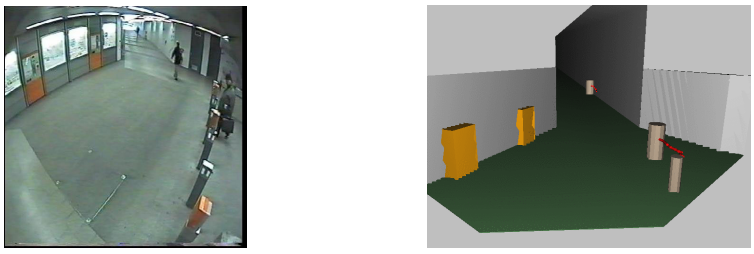


FIG. 5.4 – *Un nouvel humain h_3 entre a son tour par la droite.*

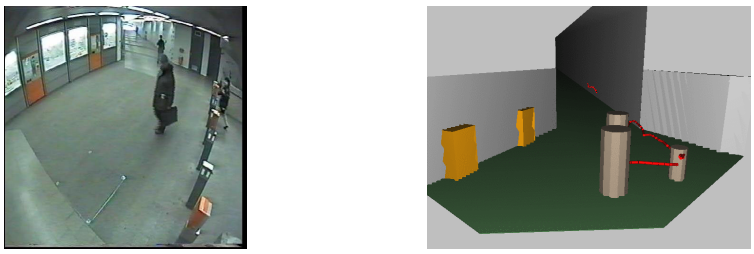


FIG. 5.5 – *Un autre humain h_4 entre par la droite. A cet instant, h_3 occulte h_2, mais l'on peut observer sur la reconstruction que h_2 n'est pas perdu. En revanche, On s'apercoit aussi que h_1 est perdu car il n'est pas detecté et se trouve dans une zone d'entrées/sorties.*

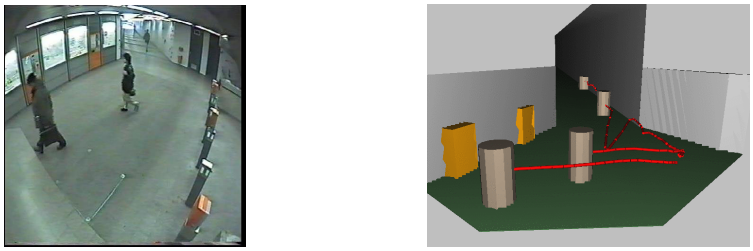


FIG. 5.6 – *Même si, h_2 n'a pas été perdu sa trajectoire est partiellement corrompue. h_1 est maintenant détecté à nouveau et un nouveau* name *lui a été attribué.*

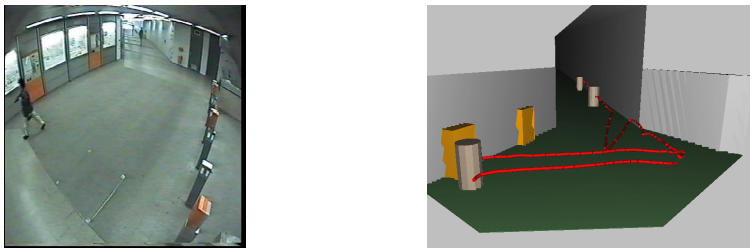


FIG. 5.7 – *h_3 et h_4 sortent de la scène sans aucun problème.*

FIG. 5.8 – *Dans une agence de la Caisse régionnale de la Brie, 2 guichetiers entrent pour s'asseoir à leur poste. h_1 (au premier plan) occulte h_2 (au second plan). Il n'y a plus de détection associable à h_2. Celui ci sera alors considéré comme sorti.*

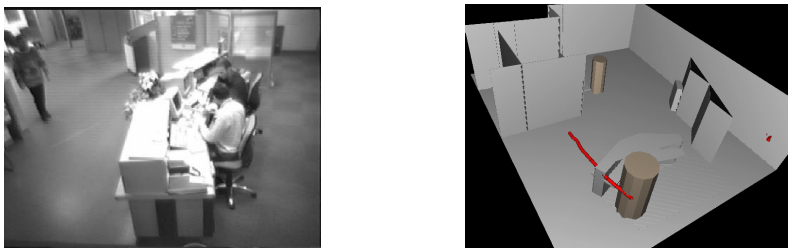


FIG. 5.9 – *h_3 entre dans la scène et est correctement détecté.*

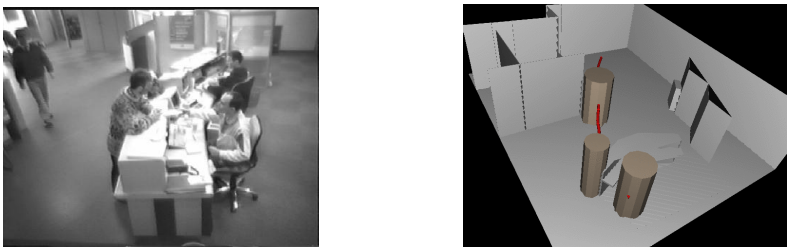


FIG. 5.10 – *h_4 entre dans la scène. On peut observer à cet instant une erreur typique sur la localisation au sol de h_3 causée par sa propre ombre.*

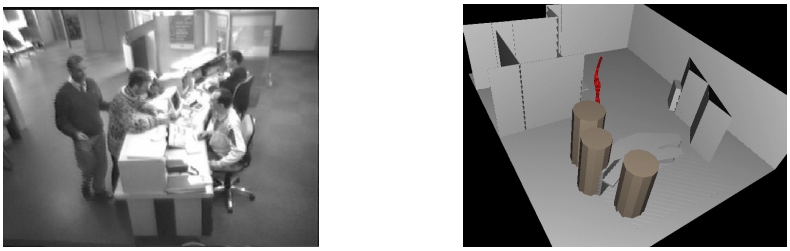


FIG. 5.11 – *h_3 et h_4 entrent en contact , i.e. qu'il n'existe plus qu'un seul ensemble de blobs $q \in Q_t$ pour représenter deux humains, mais l'on peut s'apercevoir que chacun d'eux (h_3 et h_4) continue à être considérés comme deux personnes.*

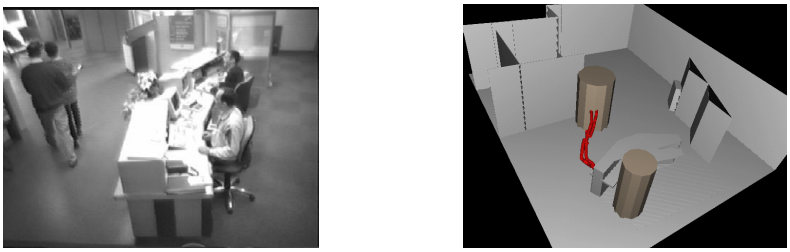


FIG. 5.12 – *Ce manque de détection (un unique ensemble de blobs pour deux humains) persiste dans le temps, mais h_3 et h_4 sont toujours considérés comme deux humains. (superposés sur la reconstruction).*

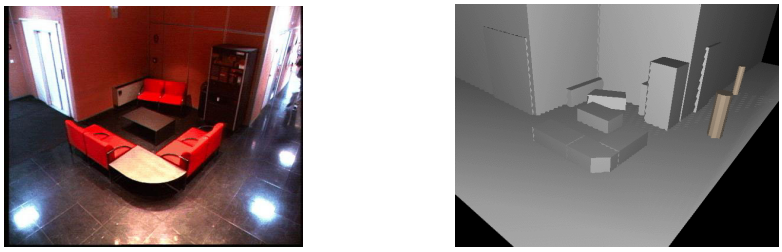


FIG. 5.13 – *h_1 entre dans la scène par la droite. Un mélange de réflexions et d'ombres au sol sont reconnues comme étant humain.*

FIG. 5.14 – h_1 *et* h_2 *sont correctement localisés, mais on peut observer sur la reconstruction que la porte de l'ascenseur (à gauche) est reconnue comme étant un humain.*

FIG. 5.15 – h_2 *occulte* h_1*. L'un comme l'autre continuent à être correctement détectés.*

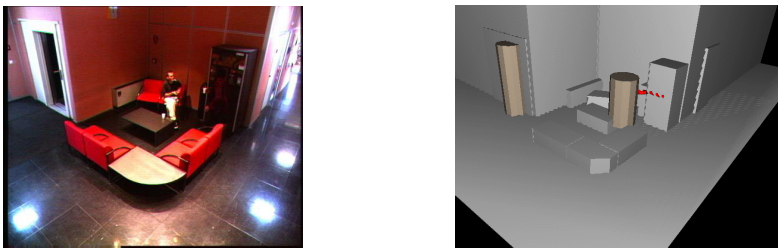


FIG. 5.16 – h_2 *étant très similaire au fond, n'est maintenant plus détecté. Il est perdue à cet instant.*

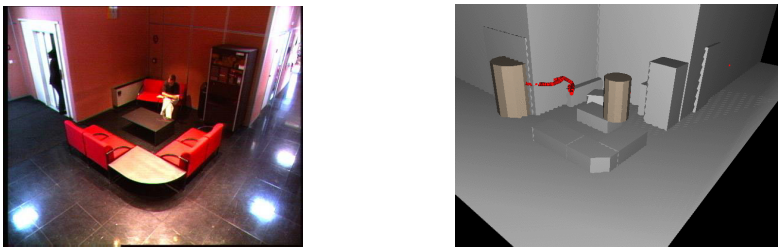


FIG. 5.17 – h_2 *quitte la scène par l'ascenseur.* h_1 *finit son café et sa cigarette.*

5.10 Conclusion

Nous avons présenté dans ce chapitre, la méthode utilisée pour le suivi de personnes. Cette méthode est basée sur la recherche du diagnostic le plus vraisemblable de l'évolution de la scène entre deux images.

L'accent a été mis dans cette méthode sur la gestion simultanée des problèmes d'occultations statiques et dynamiques, des problèmes d'entrées/sorties, des problèmes de bruit et du problème de calcul de similarité entre deux personnes.

Le premier avantage de l'approche est la gestion unifiée de l'ensemble de problèmes réels inhérents au suivi de personnes. En effet, si l'on compare avec l'ensemble des méthodes proposées dans la littérature, rares sont les méthodes qui proposent une gestion complète du problème. La majorité des méthodes se focalisent sur le problème du calcul de similarité (en gérant le bruit). Quelques méthodes prennent en compte le problème des occultations. Quant aux (très rares) méthodes gérant l'ensemble des problèmes, celles-ci opèrent de façon séquentielle.

Le second avantage de la méthode est sa rapidité (voir 8.6).

Les inconvénients de la méthode sont de deux ordres. Le premier inconvénient réside dans le choix d'un ensemble des valeurs des poids de la fonction d'évaluation d'un diagnostic. En particulier, le choix de la valeur de λ_1 est problématique car celui-ci peut être vu comme un seuil de similarité minimum. C'est à dire que de sa valeur dépend la quasi-totalité des résultats de suivi de personnes.

Le second inconvénient de la méthode réside dans le mode évaluation naïf des fonctions de mise en correspondance HIDDEN et APPEARS. Bien que celles-ci aient donné des résultats satisfaisants sur les tests effectués, il est envisageable de trouver des formes plus raffinées d'évaluation de ces deux cas.

Le troisième inconvénient est la relative simplicité du calcul de similarité. Là encore, bien que celui-ci ait donné des résultats satisfaisants, il est tout à fait envisageable de trouver un mode de calcul de similarité plus riche.

Ce point est directement lié au choix délibéré d'utiliser un modèle de personne simple (rectangle 2D/ cylindre 3D) afin de garantir une reconnaissance suffisamment robuste.

L'enrichissement du modèle de personne par des descripteurs de couleur ou de forme, si tant est qu'il soit robuste, permettrait l'élaboration d'un calcul de similarité plus sophistiqué.

Chapitre 6 Apprentissage de paramètres pour le suivi de personnes

Le but de ce chapitre est de détailler une méthode permettant de déterminer l'ensemble des paramètres de l'algorithme de suivi de personnes proposé dans les chapitres 4 et 5.

Ces travaux ont été réalisés avec la collaboration de P. Nivet dans la cadre de son stage de DEA.

La première difficulté de ce problème est que les valeurs des paramètres ne peuvent par être calculées *stricto census* (il n'existe pas de forme analytique pour le calcul des valeurs de paramètres idéaux) et donc le choix d'un ensemble de paramètres doit être fait en connaissance des algorithmes proposés. Ceci établi, déterminer un ensemble de paramètres, sans connaître précisément les algorithmes en question, consiste alors à choisir un ensemble de paramètres et à l'améliorer en évaluant son impact sur la qualité des résultats qu'il obtient.

La difficulté réelle de l'apprentissage de paramètres pour le suivi de personnes est donc de définir un mode d'amélioration efficace capable de transformer un jeu de paramètres quelconques en un jeu de paramètres utilisables.

On se place pour cela dans le cadre des algorithmes génétiques. Dans ce paradigme, un jeu de paramètres est vu comme un individu appelé chromosome. Un ensemble de chromosomes est appelé une population. Le principe général d'apprentissage par algorithme génétique consiste à faire évoluer cette population jusqu'à un état suffisamment proche de la solution cherchée, c'est à dire une population contenant un chromosome associé à un jeu de paramètres efficaces. Le principe d'évolution consiste à évaluer la qualité de chaque jeu de paramètres (c'est à dire le chromosome) afin de n'en garder, pour la population suivante, qu'une combinaison des plus efficaces. On parlera, par abus de langage, du résultat d'un chromosome pour parler du résultat du suivi de personnes par la méthode proposée paramétrée par le jeu de paramètres décrits par le chromosome.

De façon plus précise, notre approche consiste à se doter d'une séquence vidéo connue associée à un résultat idéal (un graphe optimal) que l'on souhaiterait obtenir par le suivi de personne et de faire évoluer une population en comparant le résultat obtenu par les chromosomes de cette population au résultat idéal.

Nous détaillerons, dans la section 6.1, les principes de l'évaluation d'une population et les principes d'évolution. Les résultats obtenus par cette méthode sont commentés dans la

section 6.5.

6.1 Méthodes et protocoles

6.2 Structure générale

- Soit $\mathrm{POP}_g = \{Cr_{i,g} \quad \forall i \in [1,k]\}$ l'ensemble des chromosomes $Cr_{i,g}$ formant la population POP_g à la génération g, on écrira POP_0 pour désigner la population initiale et POP_f pour la population finale.
- Soit $Cr_{i,g} = (P_0(Cr_{i,g}),...,P_n(Cr_{i,g}))$ la séquence de n gènes $P_j(Cr_{i,g})$ du chromosome $Cr_{i,g}$.

$$(P_0(Cr_{i,t}),...,P_n(Cr_{i,g})) = (\alpha,\beta,\gamma,\delta_{max},\lambda,\nu,R,K,A,p,w,h,P,W,H)$$

avec

1. α est la valeur du seuil de l'opérateur TH_α (voir 4.1).
2. β est le niveau d'intégration de l'image courante dans l'image de référence (voir 4.1).
3. γ est la pondération de la densité dans l'heuristique κ de groupement des blobs (voir 4.3).
4. δ_{max} est la distance maximale admissible d'un ensemble de blobs au modèle *a priori* d'humains (voir 4.3).
5. λ est la meta-valeur déterminant les poids $\lambda_1,...\lambda_7$ des préférences des fonctions de mise en correspondance (voir 5.6).
6. ν est un paramètre d'évaluation de la fonction de mise en correspondance LOST (voir 5.1).
7. R,K et A sont les poids des modes de filtrage intervenant dans le calcul de similarité (voir 5.1).
8. p,w et h sont les poids des mesures 2D intervenant dans le calcul de similarité (voir 5.1).
9. P,W et H sont les poids des mesures 3D intervenant dans le calcul de similarité (voir 5.1).

La structure générale de l'algorithme génétique est donc:

CONSTRUIRE POP_0
ÉVALUER POP_0
TANT QUE $\nexists \quad Cr_{i,g} \in \mathrm{POP}_g$ tel que $eval(Cr_{i,g}) > e_{opt} - \varepsilon$.
    CONSTRUIRE POP_{g+1}

ÉVALUER POP_{g+1}

avec $eval(Cr_{i,g})$ le résultat de l'évaluation du chromosome $Cr_{i,g}$, e_{opt} l'évaluation optimale et ε la tolérance de l'algorithme à l'évaluation optimale

En d'autres termes, tant qu'il n'existe pas dans la population courante de chromosome dont l'évaluation est suffisante, on construit une nouvelle population et on l'évalue.

Nous verrons dans la section 6.3 les techniques mises en oeuvre pour évaluer la qualité d'un chromosome et dans la section 6.4 les principes de construction d'une population à partir d'une autre.

6.3 Evaluation d'une population

L'évaluation consiste à attribuer un score à chaque chromosome en comparant le graphe calculé avec le graphe optimal. Plus le graphe à évaluer ressemble au graphe optimal, noté $\bar{G}^{opt}$, plus son score se rapproche du score maximum e_{opt}.

L'évaluation d'un chromosome est faite en deux étapes: la première étape consiste à obtenir un graphe résultant du suivi de personnes. La seconde étape consiste à mesurer la similarité entre ce graphe et le graphe optimal et l'on notera :

$$\bar{G}^{i,g} = g(Cr_{i,g})$$

où $\bar{G}^{i,g}$ est le graphe résultat du suivi de personnes paramétré par le chromosome $Cr_{i,g}$

La seconde étape est appelée la *fitness* et est définie par:

$$e_{i,g} = f(\bar{G}^{i,g},\bar{G}^{opt})$$

avec $e_{i,g}$ le score du graphe $\bar{G}^{i,g}$, $\bar{G}^{opt}$ le graphe optimal et f la fonction de *fitness*.

Après avoir calculé le graphe correspondant au chromosome à évaluer, il est possible de le comparer au graphe optimal. Comparer un graphe quelconque à un graphe optimal consiste à un trouver une mesure de similarité entre deux graphes. On définit la fonction de *fitness* comme une somme pondérée de quatre termes fonctionnels estimant la similarité de deux graphes.

$$f(\bar{G}^{i,g},\bar{G}^{opt}) = \mathbf{P}_{\alpha}.f_1(\bar{G}^{i,g},\bar{G}^{opt}) + \mathbf{P}_{\beta}.f_2(\bar{G}^{i,g},\bar{G}^{opt}) + \mathbf{P}_{\gamma}.f_3(\bar{G}^{i,g},\bar{G}^{opt}) + \mathbf{P}_{\lambda}.f_4(\bar{G}^{i,g},\bar{G}^{opt})$$

avec f_1,...,f_4 les quatre termes fonctionnels détaillés dans la suite.

6.3.1 Similarité des personnes

f_1 permet d'évaluer et d'intégrer, au score, la différence du nombre de personnes par frame dans le graphe optimal et le graphe à évaluer:

$$f_1(\bar{G}^{i,g},\bar{G}^{opt}) = 1 - \frac{\sum_{d=0}^{L} ||P_d^{opt}| - |P_d^{i,t}||}{\sum_d |P_d^{opt}|)}$$

où L est le nombre de frames de la séquence vidéo, $|P_d^{opt}|$ est le nombre de personnes à l'instant d dans la graphe optimal et $|P_d^{i,g}|$ est le nombre de personnes à l'instant d dans le graphe calculé.

6.3.2 Similarité des individus

f_2 fait de même pour la différence du nombre global de personnes dans le graphe optimal et le graphe à évaluer:

$$f_2(\bar{G}^{i,g},\bar{G}^{opt}) = 1 - \frac{||I^{opt}| - |I^{i,t}||}{|I^{opt}|}$$

$|I^{opt}|$ est le nombre d'individu dans le graphe optimal et $|I^{i,g}|$ est le nombre d'individus dans le graphe calculé.

6.3.3 Similarité temporelle

f_3 évalue la ressemblance des deux graphes par rapport à l'instant d'entrée et à l'instant de sortie de chaque individu après avoir effectué leur appariement temporel d'un graphe à l'autre:

$$f_3(\bar{G}^{i,g},\bar{G}^{opt}) = \begin{cases} 0 & \text{si} \quad |I^{opt}| = |I^{i,t}| \\ f_3'(\bar{G}^{i,g},\bar{G}^{opt}) & \text{si} \quad |I^{opt}| \neq |I^{i,t}| \end{cases}$$

avec

$$f_3'(\bar{G}^{i,g},\bar{G}^{opt}) = \min_{b\in\mathcal{B}}(\sum_{j\in I(\bar{G}^{i,g})} \text{t-dist}(j,b(j)) \quad)$$

où $\mathcal{B}$ est l'ensemble des bijections b entre les individus de $\bar{G}^{i,g}$ et ceux de $\bar{G}^{opt}$, I^G l'ensemble des individus de G et t-dist la distance temporelle entre deux individus. En d'autres termes, f_3' représente la plus petite somme de distances temporelles entre les individus de $\bar{G}^{i,g}$ et ceux de $\bar{G}^{opt}$ appariés deux à deux. Ceci est rendu possible dans la mesure où $\bar{G}^{i,g}$ et $\bar{G}^{opt}$ ont le même nombre d'individus.

On parlera dans la suite, de graphes isomorphes pour parler de graphes ayant le même nombre d'individus avec des temps d'entrée et de sortie identiques.

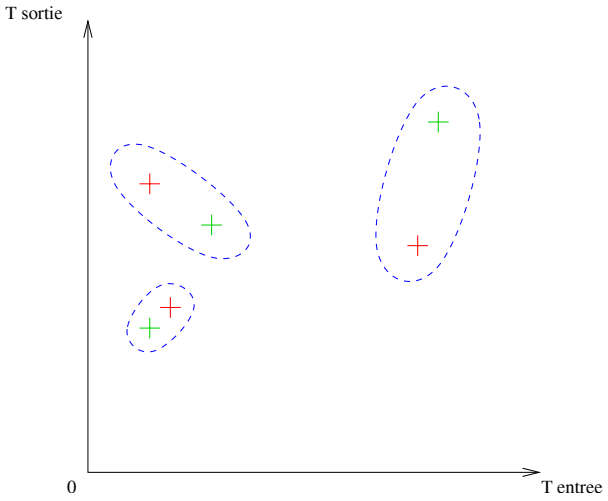


FIG. 6.1 – *Appariement temporel: il s'agit d'apparier chaque individu du graphe à évaluer avec un individu du graphe optimal en utilisant, comme critère de ressemblance, le couple (*start, end*) composé de l'instant d'entrée et de l'instant de sortie.*

6.3.4 Similarité spatiale

f_4 évalue la ressemblance des deux graphes en calculant le cumul de l'erreur de positionnement de chaque individu du graphe à évaluer par rapport à son homologue dans le graphe optimal, sur chaque frame:

$$f_4(\bar{G}^{i,g},\bar{G}^{opt}) = \begin{cases} 0 & \text{si} \quad f_3(\bar{G}^{i,g}) = 1 \\ f'_4(\bar{G}^{i,g},\bar{G}^{opt}) & \text{si} \quad f_3(\bar{G}^{i,g}) \neq 1 \end{cases}$$

avec

$$f'_4(\bar{G}^{i,g},\bar{G}^{opt}) = \min_{b\in\mathcal{B}}\Big(\sum_{j\in I(\bar{G}^{i,g})} \text{s-dist}(j,b(j)) \quad \Big)$$

6.4 Construction d'une population

L'élaboration d'une nouvelle population comporte quatre étapes :

1. sélection
2. croisement

3. mutation
4. normalisation

6.4.1 Sélection

La solution retenue consiste à conserver une certaine proportion d'individus et à compléter la population avec les meilleurs.

$$\forall j \in [0,k[,\ sel(Cr_{i,t+1}) = Cr_{(j \bmod n),t}$$

6.4.2 Croisement

Le premier des deux opérateurs génétiques est le croisement. Il recombine des schémas, plus ou moins avantageusement, en mélangeant le matériel génétique - les paramètres - autour d'un point de croisement. Pour cela, nous avons besoin de deux chromosomes Cr_1 et Cr_2.

Soient H et R, deux variables aléatoires comprises entre 0 et 1.

$$crois(Cr_1,Cr_2) = \begin{cases} (Cr_1,Cr_2) \text{ avec une probabilité } 1 - P_c(\text{ lorsque } H < P_c) \\ (Cr'_1,Cr'_2) \text{ avec une probabilité } P_c(\text{ lorsque } H > P_c) \end{cases}$$

tel que $\forall i \in [0,n-1]$:

$$Cr'_1 = \begin{cases} P_i(C'_1) = P_i(C1) & \text{si } i < n.R \\ P_i(C'_1) = P_i(C2) & \text{si } i > n.R \\ P_i(C'_1) = E_1 & \text{si } i = n.R \end{cases}$$

$$Cr'_2 = \begin{cases} P_i(C'_2) = P_i(C2) & \text{si } i < n.R \\ P_i(C'_2) = P_i(C1) & \text{si } i > n.R \\ P_i(C'_2) = E_2 & \text{si } i = n.R \end{cases}$$

E_1 et E_2 sont obtenus par croisement arithmétique réel :

Dans un premier temps, il faut effectuer un test aléatoire sur la probabilité de croisement P_c qui conditionne la réalisation ou non d'un croisement, puis déterminer une position pour le point de croisement, au hasard sur le chromosome.

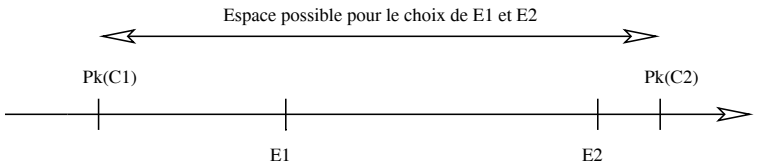


FIG. 6.2 – *Croisement arithmétique: un gène résultant d'un croisement arithmétique se situe sur la droite définie par les 2 gènes à l'origine du croisement.*

Ensuite, tout ce qui est avant le point de croisement ne change pas; si le point de croisement tombe sur un gène, celui-ci subit un croisement arithmétique et tout ce qui est après est échangé avec le matériel de l'autre chromosome.

Le croisement arithmétique, illustré dans la figure 6.2, consiste à choisir, aléatoirement, deux nouvelles valeurs, dans l'espace défini par les anciennes valeurs.

6.4.3 Mutation

Le second opérateur est la mutation, opérateur chargé de l'exploration de l'espace de recherche.

Soit H, une variable aléatoire comprise entre 0 et 1.

$$mut(Cr_{i,t}) = \begin{cases} Cr_{i,t} \text{ avec une probabilité } 1 - P_m(\text{ lorsque } H < P_m) \\ Cr'_{i,t} \text{ avec une probabilité } P_m(\text{ lorsque } H > P_m) \end{cases}$$

avec

$$Cr'_{i,t} \text{ tel que } \forall k \in [0,n[, P_k(Cr'_{i,t}) = (\sigma[-\delta_{mut},\delta_{mut}] + 1).P_k(Cr_{i,t})$$

$\sigma[a,b]$ étant la génération aléatoire d'un nombre réel appartenant à $[a,b]$.

Dans un premier temps, on effectue un test aléatoire sur la probabilité de mutation P_m qui en conditionne la réalisation, puis cet opérateur ajoute ou retranche, au gène à muter, une valeur tirée aléatoirement dans $[-\delta_{mut},\delta_{mut}]$.

6.4.4 Normalisation

La normalisation de chaque chromosome consiste à vérifier l'appartenance de chaque paramètre à son domaine de validité défini par un couple (Cr_{min},Cr_{max}) et à ramener ce paramètre dans son domaine, le cas échéant.

$$\forall i, norm(\{Cr_{i,t}\}) = \begin{cases} \forall j \in [0,5], & \text{si } P_{i,t,j} \notin [P_{min,j}, P_{max,j}], & \text{alors } P_{i,t,j} = mod(P_{i,t,j}) \\ \forall j \in [6,8], & \text{si } \sum_j P_{i,t,j} \neq 100, & \text{alors } P_{i,t,j} = P_{i,t,j} \cdot \frac{100}{\sum_j P_{i,t,j}} \\ \forall j \in [9,14], & \text{si } \sum_j P_{i,t,j} \neq 100, & \text{alors } P_{i,t,j} = P_{i,t,j} \cdot \frac{100}{\sum_j P_{i,t,j}} \end{cases}$$

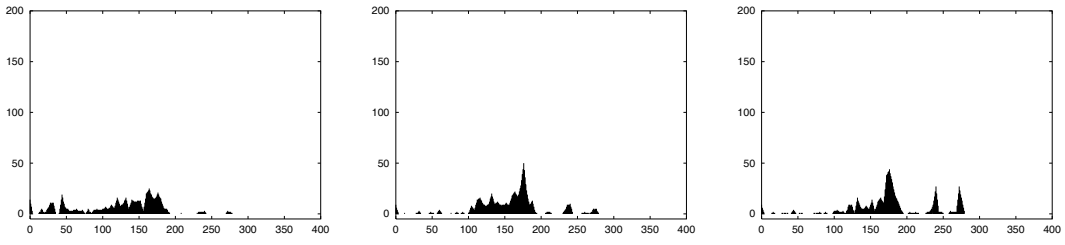
Les paramètres 0 à 5 doivent appartenir à un certain domaine, il suffit donc de vérifier cette appartenance et d'appliquer un modulo si besoin est.

Les paramètres 6 à 8 et 9 à 14 forment deux groupes de paramètres dont la particularité est de représenter la répartition d'un pourcentage. Ainsi, il est nécessaire de normaliser à 100 la somme de chacun de ces deux groupes de paramètres avant l'évaluation.

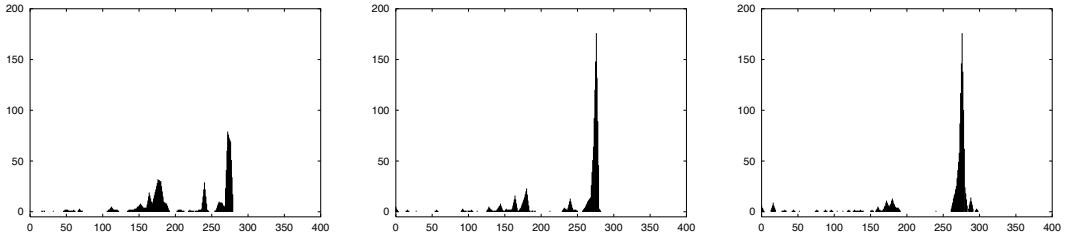
6.5 Exemple d'évolution

Afin de comprendre la nature de l'évolution d'une population quelconque jusqu'à une population idéale, nous présentons, en premier lieu, un exemple de cette évolution. Les figures 6.3 illustrent le comportement de l'algorithme d'apprentissage.

La population dont les scores initiaux sont répartis entre 0 est 200 ($\mathbf{P}_\alpha + \mathbf{P}_\beta$) évolue jusqu'au point où la quasi totalité des scores sont égaux à 400 ($\mathbf{P}_\alpha + \mathbf{P}_\beta + \mathbf{P}_\gamma + \mathbf{P}_\lambda$). Cette évolution peut être suivie sur le déplacement des piques de scores.



Génération 02 Génération 04 Génération 06



Génération 09 Génération 12 Génération 18

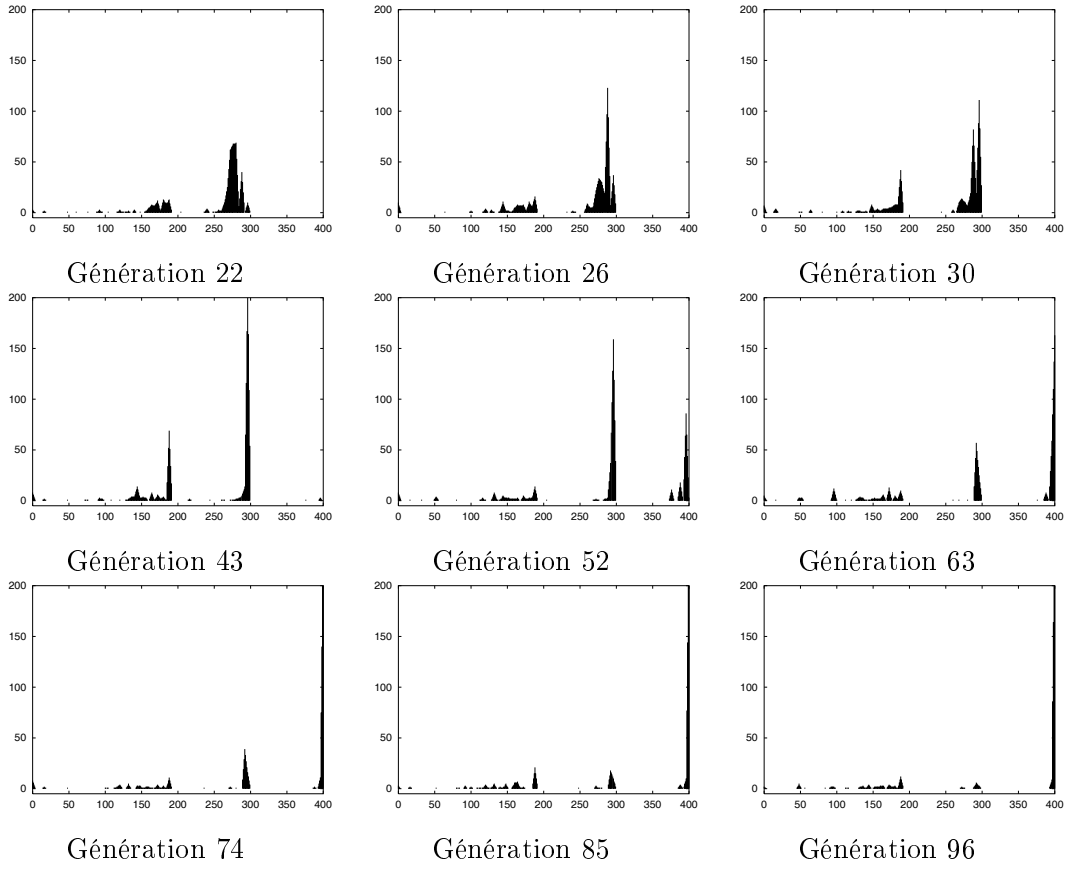


FIG. 6.3 – *Illustration de l'évolution d'un ensemble de parametres par algorithmes génétiques*

6.6 Résultats

Nous présentons dans cette section des résultats obtenus par cette méthode. A ce titre, l'enjeu de la méthode étant de savoir si la fonction de fitness va permettre de faire converger l'ensemble des scores vers l'évaluation souhaitée. Nous définissons un protocole expérimental nous permettant de contrôler l'influence des contraintes de l'approche.

Ce protocole consiste à définir trois classes de tests:

1. La première classe de tests porte sur la recherche d'un chromosome dont on garantit l'existence et dont on facilite la recherche en imposant une population initiale proche de la solution recherchée. En d'autres termes, la première classe de tests a pour objectif la recherche d'un chromosome à partir d'un graphe obtenu par l'algorithme de suivi et initialisée par une population dont les chromosomes sont proches de celui ayant

servi à calculer le graphe en question. Ce chromosome est appelé la «souche».

2. La seconde classe de tests est similaire à la première en ceci que le graphe optimal est issu de l'algorithme de suivi de personnes (donc le chromosome recherché existe). En revanche, l'initialisation est pour la seconde classe de tests totalement aléatoire.
3. La troisième classe de tests est caractérisée par d'une part, une initialisation aléatoire et d'autre part, un graphe optimal fait à la main. C'est à dire que rien ne garanti l'existence d'un jeu de paramètres capables d'obtenir ce graphe. Ce graphe est appelé le graphe "Ground Truth".

N.B: Notons que dans ce protocole le valeur de β est fixée à 0.

6.6.1 Tests de la première classe: avec chromosome souche et existence de la solution

Le chromosome donnant le graphe idéal étant connu, on peut construire la population initiale autour de cette souche en lui appliquant une perturbation aléatoire plus ou moins importante. L'ordre de grandeur du maximum des variations sera successivement de 5%, 10%, 50% de l'espace de validité de chaque paramètre. L'approche consiste à vérifier la convergence d'une population fortement regroupée autour de la souche (5% de variation), puis à augmenter jusqu'à 50%.

Les figures 6.4, 6.5, 6.6 présentent les résultats obtenus sur les tests de la première classe. Ces figures représentent l'évolution des scores (en ordonnée) au cours des génération (en absisse). La moyenne des scores est présentée en vert. L'évolution de l'écart type des scores est présentée en bleu. L'évolution du score maximal par population est présentée en rouge.

Comme nous pouvions nous y attendre, le nombre de générations nécessaires décroit avec l'amplitude des perturbations appliquées à la souche pour générer la population initiale.

Les tests de la première classe ont été faits sur différentes séquences vidéos de nature et de complexité diverse. Les tables 6.1 et 6.2 récapitulent ces résultats. Le tableau 6.1 présente le nombre de générations nécessaires pour obtenir un graphe isomorphe au graphe idéal. Le tableau 6.2 présente le nombre de générations nécessaires pour obtenir un graphe exactement identique au graphe idéal.

On note que pour une même fourchette de variation autour du chromosome souche, deux graphes de deux séquences différentes ne seront pas appris en un même nombre de générations. Par exemple, le graphe associé à ST1-23, avec une fourchette de variation de 50%, est appris en 11 générations alors que le graphe associé à MC2-17, avec la même fourchette de variation est appris en 46 générations. Nous pouvons fournir une première

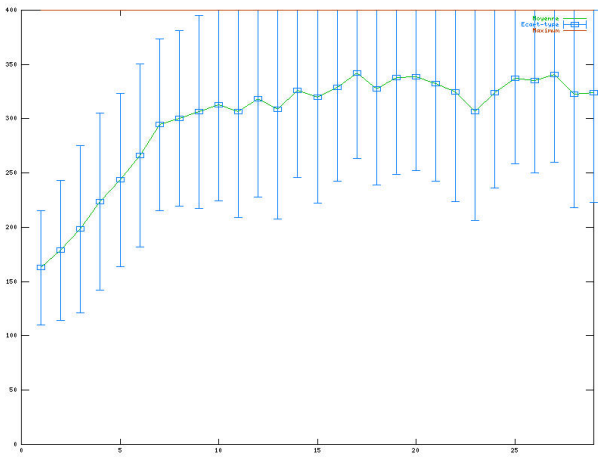


FIG. 6.4 – *Apprentissage sur VA2-7 avec une souche autour de 5%*

ST1-23	0	0	0
C07-2	0	0	0
VA2-7	0	4	22
MC2-17	0	0	16

TAB. 6.1 – *Ce tableau récapitule, pour différentes fourchettes de variation autour du chromosome souche, le nombre de générations nécessaires à l'obtention d'un graphe isomorphe au graphe idéal.*

explication: tout d'abord, le nombre de personnages présents dans la scène nous donne un bon indicateur de sa complexité. En effet, ST1-23 ne comporte qu'une personne alors que MC2-17 en comporte quatre, ce qui peut expliquer la convergence plus lente pour MC2-17. Mais cela ne suffit pas si l'on compare MC2-17 et VA2-7 car, dans ce cas, le nombre de personnes - quatre - est le même. Il est donc évident que certaines séquences nécessitent, de par leur nature, une recherche plus poussée, un réglage plus fin des paramètres pour obtenir le graphe optimal.

6.6.2 Tests de la seconde classe: avec chromosome aléatoire et existence de la solution

Par la suite, on utilise une population générée aléatoirement dans l'espace des chromosomes valides. Cela correspond, dans le cas précédent, à une variation maximale de 100% car, ainsi, la souche n'a plus d'influence sur la composition de la population initiale. De

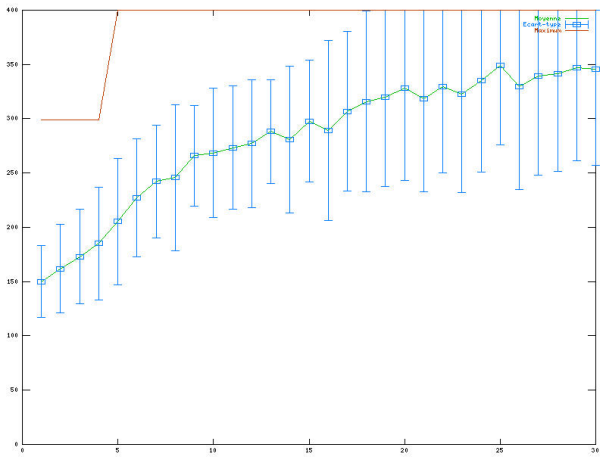


FIG. 6.5 – *Apprentissage sur VA2-7 avec une souche autour de 10%*

référence de la vidéo	5%	10%	50%
ST1-23	0	0	11
C07-2	0	10	26
VA2-7	0	20	32
MC2-17	4	20	46

TAB. 6.2 – *Ce tableau récapitule, pour différentes fourchettes de variation autour du chromosome souche, le nombre de générations nécessaires à l'obtention du graphe idéal, c'est-à-dire sans aucune erreur de placement.*

cette façon, on peut effectuer une recherche des paramètres optimaux, sans connaissance à priori, et vérifier la convergence de l'algorithme.

La figures 6.7 présentent les résultats obtenus sur les tests de la seconde classe.

Les tests de la seconde classe ont eux aussi été faits sur différentes séquences vidéo de nature et de complexité diverse. La table 6.3 récapitule ces résultats.

On note là encore, que le nombre de générations nécessaires à l'obtention du graphe idéal est négligeable par rapport à la taille de l'espace de recherche.

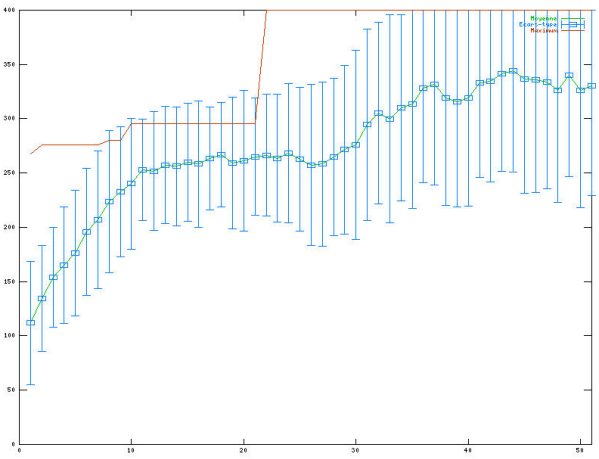


FIG. 6.6 – *Apprentissage sur VA2-7 avec une souche autour de 50%*

réference de la vidéo	nombre de générations pour l'obtention de l'isomorphisme	nombre de générations pour l'obtention du graphe idéal
ST1-23	0	12
C07-2	0	49
VA2-7	43	51
MC2-17	43	120

TAB. 6.3 – *Ce tableau récapitule le nombre de générations nécessaires à l'obtention, à partir d'une population totalement aléatoire, d'un graphe isomorphe au graphe idéal et le nombre de générations nécessaires à l'obtention du graphe idéal, c'est-à-dire sans aucune erreur de placement.*

6.6.3 Tests de la troisième classe: avec chromosome aléatoire sans garanti d'existence de solution

Les tests de la troisième classe ont pour but de recherchée une solution à partir d'un graphe obtenu à la main. Cette classe de tests constitue l'objectif principal de l'approche poursuivi. C'est à dire fournir une méthode permettant de trouver un jeu de paramètres efficace, sans aucune connaissance d'algorithme à paramétrer.

A titre de comparaison avec les tests de la première et la seconde classe, la figures 6.8 présentent les résultats obtenus sur les tests de la troisième classe sur la même séquence vidéo que les résultats présentés sur les figures 6.4, 6.5, 6.6 et 6.7.

La première conclusion à tirer des résultats présentés sur la figure 6.8 est que la conver-

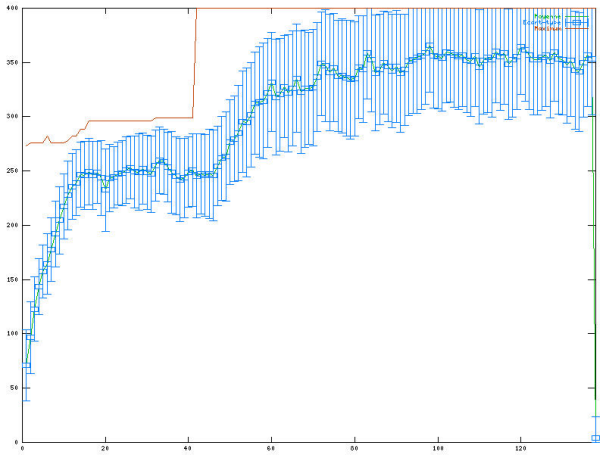


FIG. 6.7 – *Apprentissage sur VA2-7 sans souche*

gence est aussi rapide (moins de 50 génération). En revanche, on remarque aussi que la convergence ne se fait pas sur la valeur de l'évaluation maximum e_{opt}, mais sur $e_{opt} - \epsilon$ ($\epsilon = 126$). Ceci signifie que le graphe optimal n'a pas été trouvé.

Ce phénomène étant attendu, on s'intéresse plutôt à la sémantique de la valeur du score de convergence. En effet, si la convergence se fait sur une valeur légèrement inférieur à $\mathbf{P}_\alpha + \mathbf{P}_\beta + \mathbf{P}_\gamma + \mathbf{P}_\lambda$, alors l'algorithme a convergé vers un graphe isomorphe ayant une faible distance spatiale sur l'ensemble des individus de la vidéo. Si la convergence se fait sur une valeur légèrement supérieure à $\mathbf{P}_\alpha + \mathbf{P}_\beta + \mathbf{P}_\gamma$, alors l'algorithme a convergé vers un graphe isomorphe ayant une forte distance spatiale sur l'ensemble des individus de la vidéo. Si la convergence se fait sur une valeur inférieure à $\mathbf{P}_\alpha + \mathbf{P}_\beta + \mathbf{P}_\gamma$, alors l'algorithme a convergé vers un graphe non isomorphe (c'est à dire ayant le même nombre d'individus, mais entrant ou sortant à des temps différents) Si la convergence se fait sur une valeur inférieure à $\mathbf{P}_\alpha + \mathbf{P}_\beta$, alors l'algorithme a convergé vers un graphe non isomorphe et n'ayant pas le même nombre d'individus.

La table 6.4 récapitule ces résultats. On présente ici les résultats en fonction des coordonnées du point de convergence. C'est à dire la génération à partir de laquelle la population n'évolue plus; couplée avec la valeur du score maximal de cette population. Les tests en question ayant été poussés 400 générations après le point de convergence, on admettra que la population n'évolue plus. On donne en outre, dans cette table, la sémantique de la valeur du score de convergence. C'est à dire la distance spatiale obtenue, si le score est supérieur à 300 ($\mathbf{P}_\alpha + \mathbf{P}_\beta + \mathbf{P}_\gamma$) et la distance temporelle obtenue, si le score est supérieur à 200 ($\mathbf{P}_\alpha + \mathbf{P}_\beta$).

On observe que les résultats de l'approche sont assez hétérogènes. Sur les vidéos C07-2

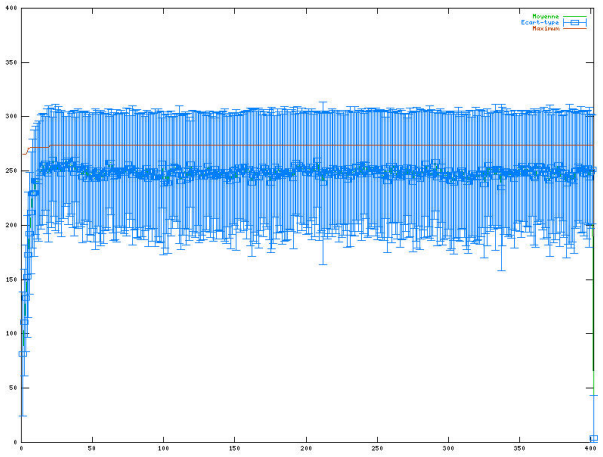


FIG. 6.8 – *Apprentissage sur VA2-7 sans souche et sans garantie d'existence de solution*

référence de la vidéo	nombre de générations jusqu'à la convergence	score de convergence	sémantique de la convergence
C07-2	11	359	41 mètres ($\simeq$ 0.27 mètre par personne)
ST1-23	23	321	581 mètres ($\simeq$ 3 mètres par personne)
MC2-17	10	293	9 frames ($\simeq$ 2 frames individu)
VA2-7	25	274	46 frames ($\simeq$ 12 frames individu)

TAB. 6.4 – *Ce tableau récapitule les résultats obtenus à partir d'une population totalement aléatoire et d'un graphe "Ground Truth".*

et ST1-23, les graphes obtenus sont isomorphes (en 5 générations) et la distance spatiale qui les séparent des graphes "Ground Truth" respectifs est quasiment négligeable (quelques centimètres pour C07-2 et quelques mètres pour ST1-23, dont la taille de la scène est beaucoup plus grande). Sur les vidéo MC2-17 et VA2-7, le graphe isomorphe n'est pas atteint. En effet, trois des quatre humains impliqués dans MC2-17 et VA2-7 sont sujets à un retard de 1 frame sur leur temps d'entrée ainsi que leur temps de sortie. En revanche, on vérifie, empiriquement que le graphe obtenu est satisfaisant dans le cas de MC2-17. C'est à dire qu'à part les erreurs sur les temps d'entrée/sortie, les erreurs sur les distances spatiales sont réduites.

Ce point est important dans la mesure où il révèle l'inconvénient majeur du mode d'évaluation choisi. En effet, l'utilisation comme fonction de fitness, d'une somme interdépendante de termes permet l'interpretation directe du score. En contrepartie, cette fonction impose un séquencement de l'évolution. L'algorithme doit d'abord ajuster le nombre de per-

sonnes, puis le nombre d'individus, puis les temps d'entrée/sortie et enfin les localisations des personnes.

6.7 Analyse de la solution

6.7.1 Analyse de la Robustesse de paramteres idéaux

L'analyse de la robustesse des paramètres idéaux a pour but de testet les paramètres calculés par l'algorithme génétique sur des vidéos différentes afin d'analyser le caractère réutilisable de ceux-ci.

Pour cela nous avons testé chacun des jeux de paramètres obtenus sur des vidéos ayant le même environnements (c.a.d O_t identiques, conditions similaires mais nombre d'individus et problèmes différents).

L'expérience a été menée sur trois séquences. Voici les erreurs rencontrées à l'étape de diagnostic, dont le nombre (6 erreurs sur 535+322+123=980 frames) reste très raisonnable:

frame	diagnostique produit	diagnostique attendu
apprentissage: C07-2, test: C02-2		
0	ENTRY(ombre)	NOISE(ombre)
66	MATCH(piste(N),R)	APPEAR(R),LOST(N)
98	MATCH(piste(R),N)	APPEAR(N),LOST(R)
apprentissage: VA2-7, test: VA2-6		
49	MATCH(a,b),MATCH(c,d)	MATCH(a,d),MATCH(c,b)
apprentissage: MC2-17, test: MC2-18		
89	MATCH(piste(R),R), MATCH(piste(B),chaise)	MATCH(piste(R),R), HIDDEN(piste(B),R), NOISE(chaise)
98	APPEAR(bureau)	NOISE(bureau)

TAB. 6.5 – *Ce tableau récapitule les erreurs rencontrées sur les séquences de test avec des paramètres appris avec les séquences d'apprentissage respectives. Pour l'explication des fonctions, se reporter à la section 5.3.*

Explications concernant les erreurs rencontrées:

- vidéo C02-2:
  - frame 0: une ombre est détecté comme une personne

paramètre	ST1-23	C07-2	VA2-7	MC2-17
α	30	70	30, 41 ou 45	28
γ	97, 98	80	[60,70]	80
Δ_{max}	88	[79,97]	[60,64]	76, 79
λ	7, 8	[1,4]	[5,14]	1, 2
r,k,a	$a >= 50\%$ $r + k <= 50\%$	$r = 0\%$ $k = 0\%$ $a = 100\%$	$r \in [0,6]\%$ $k \in [50,85]\%$ $a \in [10,50]\%$	$r \in [30,80]\%$ $k = 0\%$ $a \in [30,60]\%$
p,w,h,P,W,H	$p = 5\%$ $w \in [15,30]\%$ $h \in [0,15]\%$ $P = 15\%$ $W \in [0,20]\%$ $H \in [20,30]\%$	$p \in [85,100]\%$ $w = 0\%$ $h = 0\%$ $P = 0\%$ $W = 0\%$ $H \in [0,15]\%$	$p \in [75,100]\%$ $w = 0\%$ $h = 0\%$ $P = 0\%$ $W = 0\%$ $H \in [10,30]\%$	$p = 0\%$ $w \in [29,40]\%$ $h = 0\%$ $P \in [25,33]\%$ $W = 0\%$ $H \in [40,46]\%$

TAB. 6.6 – *Ce tableau est un résumé des différents chromosomes amenant au graphe optimal, pour les séquences ST1-23, C07-2, VA2-7 et MC2-17.*

  - frame 66: N, présent dans la scène, est perdu et sa piste est associée à R qui rentre à cet instant.
  - frame 98: N n'a plus de piste et lorsqu'il réapparait, est considéré comme du bruit. R est perdu à son tour, et la piste de R est associée à N.
- vidéo VA2-6:
  - frame 49: deux individus trop proches empêchent le tracker de les différencier, les pistes sont échangées.
- vidéo MC2-18:
  - frame 89: la piste de B (occulté par R) est associée à une chaise.
  - frame 98: un bruit parasite est considéré comme une personne sortant d'une occultation.

6.7.2 Analyse de la nature de paramètres idéaux

La table 6.6 détaille la nature des chromosomes obtenus par la méthode. C'est à dire les valeurs des paramètres obtenus pour chacune des vidéos.

La conclusion à tirer de table 6.6 est que, bien que les valeurs prise par les gênes des chromosomes optimaux soient explicables, elles étaient difficilement prévisibles.

En effet, le fait que la vidéo ST1-23 ne soit composée que d'un seul individu, relâche la contrainte sur le seuil λ (on peut se permettre un λ fort); permettant ainsi de gérer des MATCH délicats. De la même façon, la forte valeur de α ($\alpha = 70$), pour C07-2, peut être expliquée par le haut niveau de bruit de cette vidéo. Quant à la répartition des poids des filtres sur cette même vidéo, le caractère exclusif du poids du filtre moyen (100%) peut être expliqué par le fait d'une part, que la localisation des personnes est souvent bruitée (c.a.d les localisations doivent être filtrées), mais que le mouvement des individus n'étant pas du tout linéaire le filtrage de Kalman ne convient pas. Sur cette même vidéo ST1-23, on peut observer le caractère exclusif de poids de la localisation 2D ($p \in [85,100]\%$). Ceci est encore tout à fait explicable par la nature du mouvement des individus dans la scène. Les individus étant rarement en contact avec le sol, le positionnement dans l'image est préféré au positionnement dans la scène. A l'opposé, sur la vidéo VA2-7, on peut observer la caractère quasi exclusif du poids du filtrage de Kalman ($k \in [50,85]\%$). Ce point est explicable, d'une part, par le caractère relativement linéaire du mouvement des individus et d'autre part, par le grand nombre d'occultations dynamiques qui s'y produisent (le filtrage de Kalman luttant efficacement contre les erreurs de positionnement dues aux occultations).

Ceci tend à laisser penser que les valeurs des gènes des chromosomes peuvent être déterminées à l'avance, de façon qualitative, par l'analyse des conditions du suivi de personnes. Malheureusement, cette analyse ne suffit pas dans la mesure où l'ensemble des règles que l'on imagine amèneraient à des conclusions opposées les unes aux autres.

Par exemple, la qualité très médiocre de la calibration de la caméra de la vidéo ST1-23 nous amène à penser que les mesures 2D seraient préférées aux mesures 3D. On voit sur la table 6.6 que ce n'est pas le cas. De la même façon, la grande dispersion dans l'espace 3D, comparée à la faible dispersion dans l'image des individus de la vidéo VA2-7, pourrait laisser penser que les mesures 3D sont préférables aux mesures 2D. La table 6.6 montre le contraire. Pour finir, dans la vidéo MC2-17, le suivi de personnes a à faire face à de nombreux problèmes de fausses reconnaissances. On imagine alors que la valeur de Δ_{max} devrait être petite. Là encore, l'algorithme génétique montre que le jeu de paramètres idéals ne suit pas cette règle.

La conclusion à tirer de ces résultats est que, bien que les valeurs prise par les gènes des chromosomes optimaux sont explicables *a posteriori*, ces valeurs sont difficilement prévisibles.

6.8 Conclusion

Nous avons présenté, dans ce chapitre, la méthode utilisée pour la paramétrisation de la méthode de suivi de personnes proposée. Cette méthode est l'apprentissage des différents paramètres par algorithme génétique.

La difficulté réelle de l'apprentissage de paramètres pour le suivi de personne est donc de définir un mode d'amélioration efficace capable de transformer un jeu de paramètres quelconques en jeu de paramètres utilisables.

Nous avons proposé comme mode d'amélioration, outre un mode d'évolution basé sur les opérateurs génétiques classiques, une méthode d'évaluation de la qualité d'un jeu quelconque de paramètres. Ce mode d'évaluation consiste à comparer le résultat du suivi de personnes sur une vidéo connue, à un résultat idéal que l'on souhaiterait obtenir.

Nous avons constaté l'efficacité de ce mode d'évaluation sur un ensemble de vidéos aux caractéristiques différentes. De façon plus générale, nous avons pu constater que les algorithmes génériques semblaient apporter une solution fiable au problème récurant de la paramétrisation des algorithmes de suivi de personnes.

Chapitre 7 Reconnaissance de comportements

"**COMPORTEMENT** n.m. PSYCHOL. *Ensemble des réactions, observables objectivement, d'un organisme qui agit en réponse à une stimulation venue de son milieu intérieur ou du milieu extérieur"*

LAROUSSE 1995

L'objectif de ce chapitre est de détailler la méthode utilisée pour résoudre le problème de la reconnaissance de comportements. Cette méthode est en outre, détaillée dans [87] et [88].

Ce problème consiste à reconnaître un ensemble de comportements prédéfinis dans une séquence de représentation du monde. En d'autres termes, la reconnaissance de comportements correspond au processus d'instanciation de modèles de comportements par différents éléments de la séquence de représentation du monde calculés à partir du flux vidéo. L'hypothèse d'une solution de ce problème est donc l'existence d'une part, d'un modèle de représentation du monde calculable à partir du flux vidéo, et d'autre part l'existence d'un formalisme capable de décrire des modèles de comportements particuliers.

A ce titre, l'enjeu d'une solution à ce problème est de permettre de représenter des conditions spatiales et dynamiques, des conditions temporelles, mais aussi des conditions symboliques et logiques pour décrire comportements.

Ces deux points étant acquis, le problème consiste alors à instancier de façon efficace les modèles de comportements par différents éléments de la séquence de représentation. La notion d'efficacité prend ici deux sens différents. La première forme d'efficacité est liée à la quantité de mémoire nécessaire pour conserver l'ensemble des déductions temporaires. La seconde forme d'efficacité est liée à la façon dont sont organisés les calculs afin qu'un calcul donné n'ait pas besoin d'être fait plusieurs fois.

Nous verrons, dans la section 7.1, les principes du formalisme nous permettant de décrire des modèles de comportements et, dans la section 7.2.1, l'algorithme retenu pour la reconnaissance des modèles.

7.1 Modélisation d'un comportement

7.1.1 Définition d'un modèle de comportement

Définissons un modèle de comportement M comme un ensemble de prédicats booléens dont les variables, typées binairement par un (+ ou par un -) seraient les sommets de $\tilde{G}_t$. Nous noterons, dans la suite, un modèle de comportement M de la façon suivante:

$$\begin{array}{rcl} M & = & ((v_1,...,v_n),(c_1,...,c_p)) \\ \text{où } M & = & ((v_1:+,\ldots,v_k:+,v_{k+1}:-,\ldots,v_n:-),(c_1,...,c_p)) \\ \text{où} & & \\ (c_1,...,c_p) & \text{sont} & \text{des prédicats booléens} \\ (v_1,...,v_n) & \text{sont} & \text{des variables typées de } (c_1,...,c_p) \end{array}$$

Définissons maintenant le cardinal positif (resp. négatif) d'un modèle de comportement par le nombre de variables typées d'un + (resp. d'un -). On notera $Card^+(M)$ (resp. $Card^-(M)$) le cardinal positif (resp. négative) d'un modèle M.

On définit, en outre, la durée d'un modèle par la longueur de l' intervalle de temps représenté par l'ensemble des conditions temporelles. Nous considérons que cette durée est infinie si cet intervalle n'est pas borné. On notera $\Delta(M)$ la durée d'un modèle M.

7.1.2 Caractérisation d'une instance d'un modèle

Soit $M = ((v_1,\ldots,v_n),(c_1,\ldots,c_m))$ un modèle de comportement avec un cardinal positif égal à k ($k \leq n$) et $\tilde{G}_t = (\tilde{F}_t,\tilde{A}_t)$ un graphe d'interprétation. On dit que $(f_1,...,f_k)$ un k-uple d'éléments de $\tilde{F}_t$ est une instance du comportement M **si et seulement si**

$$C_0 \quad \wedge \quad \neg C_{k+1} \wedge \cdots \wedge \neg C_m = TRUE \quad \forall (f_{k+1},...,f_n) \in \tilde{F}_t \times \cdots \times \tilde{F}_t$$

où

$$\begin{array}{rcl} C_0 & = & (c_\alpha(f_1,...,f_n) \wedge \cdots \wedge c_\gamma(f_1,...,f_n)) \\ & & \text{n'impliquant que des variables typées d'un } + \\ C_x & = & (c_{\beta,x}(f_1,...,f_n) \wedge \cdots \wedge c_{\eta,x}(f_1,...,f_n)) \\ & & \text{impliquant les variables typées d'un } + \text{ et la } x^{th} \text{ variable typée d'un } - \end{array}$$

En d'autres termes, pour un modèle à cardinal positif égal à k,

$$M = ((v_1:+,\ldots,v_k:+,v_{k+1}:-,\ldots,v_n:-),(c_1,\ldots,c_p))$$

$v_1,\ldots,v_k$ représentent des sommets de $\tilde{G}_t$ qui doivent exister. $v_{k+1},\ldots,v_n$ représentent des sommets qui ne doivent pas exister. $c_1,\ldots,c_p$ représentent un ensemble de conditions nécessaires et suffisantes qui caractérisent le comportement à partir de la représentation du monde donné par le graphe d'interprétation $\tilde{G}_t$.

Par exemple, considérons le modèle de comportement suivant:

$$(x_0 : +, x_1 : -)$$

$$\begin{cases} c_1) & category(x_0) = pedestrian \\ c_2) & category(x_1) = pedestrian \\ c_3) & name(x_1) = name(x_0) \\ c_4) & time(x_1) - time(x_0) = 1 \end{cases}$$

$$\begin{aligned} Card^+(M) &= 1 \\ Card^-(M) &= 1 \\ \Delta(M) &= 2 \end{aligned}$$

Ce modèle est composé de 4 prédicats: $c_1,\ldots,c_4$ impliquant 2 variables, x_0 typée d'un $+$ et x_1 typée d'un -. c_1 spécifie que la première variable x_0 est un sommet de *category pedestrian*. De plus, x_0 est typée $+$ indiquant que ce sommet doit exister. c_2 spécifie que la seconde variable x_1 est aussi un sommet de *category pedestrian*. De plus, x_1 est typée - indiquant que ce sommet ne doit pas exister. c_3 spécifie que x_0 et x_1 ont la même valeur de *name*. c_4 spécifie que x_0 et x_1 ont une distance dans le temps d'une frame. En d'autres termes, une instance de M peut être trouvée si, à l'instant t, il existe un sommet représentant un humain tel que celui ci n'existe pas à l'instant précédent. Ce modèle est *pedestrian appears*; c'est le modèle représentant l'apparition d'une personne.

7.2 Processus d'interprétation

7.2.1 Calul de V_t et R_t

Pour cela, définissons trois problèmes $P_1(M,\tilde{G}_t)$, $P_2(M,\tilde{G}_t)$ et $P_3(K,\tilde{G}_t)$ où P_1 correspond au calcul des instances de M sur $\tilde{G}_t$, $P2$ correspond au calcul des nouveaux sommets et arcs correspondant à chaque solution de $P_1(M,\tilde{G}_t)$ et P_3 est l'extension du problème P_2 à un ensemble de modèle K.

Définition: $P_1(M,\tilde{G}_t)$

Soit $M = ((v_1,...,v_n),(c_1,...,c_m))$ un modèle de comportement et $\tilde{G}_t$ un graphe partiel d'interprétation. On dit que $S = \{s_1,...,s_q\} = \{(f_{1,1},...,f_{1,k}),...,(f_{q,1},...,f_{q,k})\}$ un ensemble

de k-uples d'éléments de $\tilde{F}_t$ est la solution de $P_1(M,\tilde{G}_t)$ **si et seulement si**

$$\forall s_j \in S \ s_j \text{ est une instance de } M$$
$$\textbf{ET } \forall s_j \notin S \ s_j \text{ n'est pas une instance de } M$$

En d'autres termes, la solution de $P_1(M,\tilde{G}_t)$ est l'ensemble de **toutes** les instances de M sur $\tilde{G}_t$.

Définition: $P_2(M,\tilde{G}_t)$

Soit $M = ((v_1,...,v_n),(c_1,...,c_m))$ un modèle de comportements et $\tilde{G}_t$ un graphe partiel d'interprétation.

On dit que le sous-graphe $(V_{i,t},R_{i,t})$ où $V_{i,t} = \{v_j\}$ un ensemble de sommets et $R_{i,t} = \{\{r_{j,g}\}\}$ un ensemble d'arcs entre les sommets v_j de $V_{i,t}$ et $f_{j,g}$) de $\tilde{G}_t$ est la solution de $P_2(M,\tilde{G}_t)$ **si et seulement si**

$$\{s_j = (f_{j,1},...,f_{j,k})\forall j\} \text{ est la solution de } P_1(M,\tilde{G}_t).$$

En d'autres termes, la solution de $P_2(M,\tilde{G}_t)$ est un ensemble de sommets et un ensemble d'arcs, où chaque sommet est associé à une instance de M sur $\tilde{G}_t$ et les arcs relient ces sommets avec les éléments de $\tilde{F}_t$ qui ont donné cette instance.

Définition: $P_3(K,\tilde{G}_t)$

Soit $K = \{M_1,...,M_m\}$ un ensemble de modèles de comportements et $\tilde{G}_t$ un graphe partiel d'interprétation.

On dit que le sous-graphe (V_t,R_t) est la solution de $P_3(K,\tilde{G}_t)$ appelé problème de reconnaissance de comportements **si et seulement si**

$$(V_t,R_t) = \bigcup_{M_i \in K} (V_{i,t},R_{i,t}) \text{ tel que } (V_{i,t},R_{i,t}) \text{ est la solution de } P_2(M_i,\tilde{G}_t)$$

La complexité du calcul de (V_t,R_t), est polynomial par rapport à la complexité du calcul de $P_2(M,\tilde{G}_t)$, qui a la même complexité que $P_1(M,\tilde{G}_t)$ et $P_1(M,\tilde{G}_t)$ est *NP* difficile. On peut facilement montrer ce dernier point en comparant le problème $P_1(M,\tilde{G}_t)$ avec problème de l'énumération des k-uples d'un ensemble de n éléments, qui est moins difficile et déjà *NP* difficile.

7.2.2 *Exemple de Reconnaissance*

Détaillons un exemple de reconnaissance du modèle *pedestrian appears* à l'instant $t = 14$ sur le graphe d'interprétation $\tilde{G}_{15}$ montré par la figure 7.1. Dans ce cas, $P_3(K,\tilde{G}_{15}) = P_2(M_{\textit{pedestrian appears}},\tilde{G}_{15})$, car K est un singleton réduit au modèle $M_{\textit{pedestrian appears}}$. La solution de $P_1(M_{\textit{pedestrian appears}},\tilde{G}_{15})$ est l'ensemble $S = \{s_1,...,s_q\}$ des instances de $M_{\textit{pedestrian appears}}$ sur $\tilde{G}_{15}$.

Le cardinal positif de $M_{pedestrian\ appears}$ est égal à 1, donc une instance de $M_{pedestrian\ appears}$ est caractérisée par un 1-uple (x_0) de sommets de $\tilde{G}_{15}$ tel que

$$\begin{array}{rl} & [(category(x_0) = pedestrian)] \wedge \\ \neg & [(category(x_1) = pedestrian) \wedge (name(x_1) = name(x_0)) \wedge (time(x_1) - time(x_0) = 1)] \\ = & TRUE \quad \forall\, x_1 \in \tilde{G}_{15} \end{array}$$

$$\begin{array}{rrcl} \text{alors} & P_1(M_{pedestrian\ appears}, \tilde{G}_{15}) & = & \{(p_7),(p_{11})\} \\ \text{et} & P_2(M_{pedestrian\ appears}, \tilde{G}_{15}) & = & (V_{15}, R_{15}) \end{array}$$

$$\text{avec} \begin{cases} V_{15} = \{i_1, i_2\} \\ R_{15} = \{r(i_1, p_7), r(i_2, p_{11})\} \end{cases}$$

où i_1 (resp. i_2) est un nouveau sommet de $\bar{G}_{15}$ associé à la première (resp. à la seconde) instance de $M_{pedestrian\ appears}$ et $r(i_1,p_7)$ (resp. $r(i_2,p_{11})$) est un nouvel arc de $\bar{G}_{15}$ représentant la référence de i_1 (resp. i_2) à p_7 (resp. p_{11}).

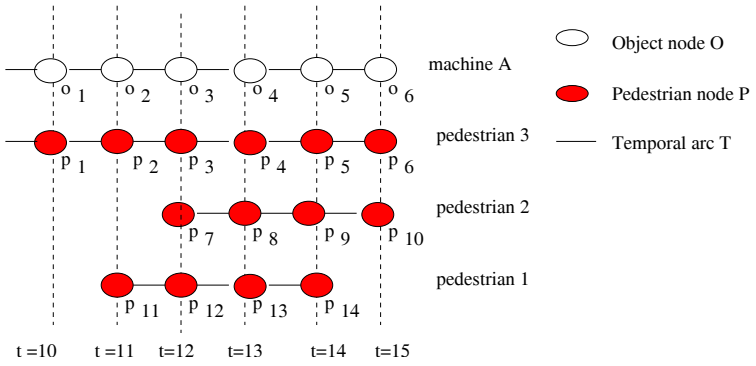


FIG. 7.1 – $\tilde{G}_{15}$*: un graphe d'interprétation à l'instant* $t = 15$ *représentant une portion de temps de 6 frames de long avec un* object *noté* machine A *en blanc et trois* pedestrians *respectivement notés* pedestrian 1, pedestrian 2 *et* pedestrian 3 *en rouge*

7.3 Gestion des Conditions

Comme il a déjà été mentionné au début de ce chapitre , l'enjeu d'une solution à ce problème est de permettre de représenter des conditions spatiales et dynamiques, des conditions temporelles, mais aussi des conditions symboliques et logiques pour décrire les comportements. Les tableaux 7.1, 7.2, 7.3 récapitulent les primitives proposées pour représenter ces conditions.

Conditions	Représentation
f_1 est loin de f_2	$distance(hull(f_1),hull(f_2)) > \Delta_1$
f_1 est proche de f_2	$distance(hull(f_1),hull(f_2)) < \Delta_2$
f_1 en contact avec f_2	$distance(hull(f_1),hull(f_2)) = 0$
f_1 est arrêté	$norm(velocity(f_1)) < \Delta_3$
f_1 est en mouvement	$norm(velocity(f_1)) > \Delta_4$
f_1 est plus proche de f_2 que f_3	$distance(hull(f_1),hull(f_2)) < distance(hull(f_1),hull(f_3))$

TAB. 7.1 – *Gestion des conditions spatiales et dynamiques*

Conditions	Représentation
f_1 est avant f_2	$time(f_1) < time(f_2)$
f_1 est après f_2	$time(f_1) > time(f_2)$
f_1 est n secondes avant f_2	$time(f_2) - time(f_1) = n$
f_1 est en même temps que f_2	$time(f_1) = time(f_2)$

TAB. 7.2 – *Gestion des conditions temporelles*

Conditions	Représentation
f_1 est le même concept réel que f_2	$name(f_1) = name(f_2)$
f_1 est une personne	$category(f_1) = pedestrian$
f_1 est un objet	$category(f_1) = object$
f_1 est vert	$property(f_1,1) = vert$
f_2 est le concept impliqué dans f_1	$ref(f_1,0) = f_2$
f_1 est de la même taille que f_2	$height(hull(f_1)) = height(hull(f_2))$

TAB. 7.3 – *Gestion des conditions symboliques et logiques*

7.4 Reconnaissance de Comportements

Nous proposons dans la section, une solution algorithmique au calcul de V_t et R_t précédemment présentée en tant que problème $P_3(K,\tilde{G}_t)$. Comme nous l'avons vu dans la section précédente (7.1), une solution de ce problème peut être déduite en temps polynomial à partir d'une solution de $P_2(M,\tilde{G}_t)$, qui peut être directement obtenue avec une solution $P_1(M,\tilde{G}_t)$. Ainsi nous n'évoquerons dans cette section que le problème $P_1(M,\tilde{G}_t)$, qui correspond à l'énumération de toutes les instances d'un modèle donné sur un graphe partiel d'interprétation donné $\tilde{G}_t$. Ce problème étant **NP** une solution polynomiale complète ne peut pas être trouvée. Malgré tout, nous présentons dans cette section une solution algorithmique permettant de réduire efficacement, sous certaines hypothèses sur M, l'énumération des instances.

Pour cela, nous présentons 4 propriétés de ce problème. La **Propriété 1** correspond au simple fait que le problème $P_1(M,\tilde{G}_t)$ est équivalent à un problème de satisfaction de contraintes (CSP) si M n'a pas de variable typée **-**. Le **Théorème 1** induit le fait que les instances de M sont nécessairement trouvées sur un sous-graphe de $\tilde{G}_t$ si $\Delta(m)$ n'est pas infini. Le **Théorème 2** et le **Théorème 3** introduisent deux importantes propriétés pour la résolution de $P_1(M,\tilde{G}_t)$. Ces deux théorèmes utilisent une transformation de M en un ensemble de modèle: $\{M_0,M_{k+1},\dots M_n\}$. M_0, appelé le "modèle seulement positif", est défini par l'ensemble des variables typé d'un $+$ ainsi que les conditions associées à ces variables. $M_x \in \{M_{k+1},\dots M_n\}$, appelé le "x-ième modèle négatif", est défini par l'ensemble des variables typé d'un $+$ et la x^{ieme} variable de M typé **-**, que l'on transforme en $+$, ainsi que les conditions associées.

Par exemple, dans le cas du modèle $M_{pedestrian\ appears}$ présenté dans la section 7.2.1 M_0 est :

$$(x_0 : +)$$

$$\{\ c_1)\quad category(x_0) \quad = \quad pedestrian$$

et M_1 est :

$$(x_0 : +, x_1 : +)$$

$$\begin{cases} c_1) & category(x_0) & = & pedestrian \\ c_2) & category(x_1) & = & pedestrian \\ c_3) & name(x_1) & = & name(x_0) \\ c_4) & time(x_1) & - & time(x_0) = 1 \end{cases}$$

7.4.1 Propriété 1

Soit $M = ((v_1,...,v_n),(c_1,...,c_m))$ un modèle de comportement tel que $Card^-(M) = 0$ et $\tilde{G}_t$ un graphe partiel d'interprétation. $P_1(M,\tilde{G}_t)$ est le problème de satisfaction de contraintes (V,D,K) avec

$$\begin{aligned} V &= (v_1,...,v_n) \\ D &= \tilde{F}_t \\ K &= (c_1,...,c_m) \end{aligned}$$

En effet:

$$\begin{aligned} P_1(M,\tilde{F}_t) &= \{(f_{1,1} \dots f_{1,n}) \dots (f_{q,1} \dots f_{q,n})\} \text{ tel que} \\ \exists i \in [1,q] \quad & c_1(f_{i,1},...,f_{i,n}) \wedge ... \wedge c_m(f_{i,1},...,f_{i,n}) = \text{TRUE} \end{aligned}$$

qui est la définition d'un problème de satisfaction de contraintes

7.4.2 Théorème 1

Soit $M = ((v_1,...,v_n),(c_1,...,c_m))$ un modèle de comportement d'un graphe partiel d'interprétation.

$$P_0(M,\tilde{G}_t) = P_0(M,f(\tilde{G}_t))$$

où $f(\tilde{G}_t)$ est le sous-graphe issu de $\tilde{G}_t$ défini par l'ensemble des sommets de $\tilde{G}_t$ dont l'attribut *time* a une valeur supérieure à $t - \Delta(M)$ et les arcs correspondants. On notera, par abus de langage $f(\tilde{F}_t)$, l'ensemble des sommets de $f(\tilde{G}_t)$.

7.4.3 Preuve du théorème 1

$\Delta(M) = d$ pour un modèle donné M signifie par définition que le plus grand intervalle de temps induit par conditions (temporelles) de M est d. En d'autres termes, tout k-uple $(f_1, \dots, f_k)$ tel que $\exists (f_\alpha, f_\gamma) \in \{f_1, \dots, f_k\} \times \{f_1, \dots, f_k\}$ avec $time(f_\alpha) - time(f_\gamma) > d$ ne peut pas être une instance de M. C'est à dire que toute instance de M doit être trouvée sur un sous-graphe de $\tilde{G}_t$ dont les sommets ont une valeur pour l'attribut *time* $\in [t - \Delta(M), t]$.

7.4.4 Théorème 2

- Soit $M = ((v_1 : +,...,v_k : +,v_{k+1} : -,...,v_n : -),(c_1,...,c_m))$ un modèle de comportement à cardinal positif égal à $k, 0 < k < n$ tel que aucune des conditions $(c_1,...,c_m)$ implique deux variables typées d'un **-**.
- Soit $M_0 = ((v_1 : +,...,v_k : +),(c_1,...,c_m))$ un modèle de comportement à cardinal positif égal à k,
- Soit $M_x = ((v_1 : +,...,v_k : +,v_x : +),(c_1,...,c_m))$ $\forall x \in [k+1,n]$ un modèle de comportement à cardinal positif égal à $k+1$.

SI $\forall x \in [k+1,n]\ P_1(M_x,\tilde{G}_t) = \emptyset$

ALORS $P_1(M,\tilde{G}_t) = P_1(M_0,\tilde{G}_t)$

7.4.5 Preuve du théorème 2

$$
\begin{aligned}
P_1(M_0,\tilde{G}_t) &= \{(f_1,...,f_k)\} \text{ tel que } C_0 = \text{TRUE} \\
P_1(M_0,\tilde{G}_t) &= \{(f_1,...,f_k)\} \text{ tel que } (c_\alpha(f_1,...,f_n) \wedge \cdots \wedge c_\gamma(f_1,...,f_n)) = \text{TRUE} \\
P_1(M,\tilde{G}_t) &= \{(f_1,...,f_k)\} \text{ tel que } C_0 \wedge \neg C_{k+1} \wedge \cdots \wedge \neg C_n = \text{TRUE} \\
&\quad \forall (f_{k+1},...,f_n) \in \tilde{F}_t \times \cdots \times \tilde{F}_t \\
P_1(M_0,\tilde{G}_t) &= P_1(M,\tilde{G}_t) \\
&\Leftrightarrow \quad \forall (f_{k+1},...,f_n) \in \tilde{F}_t \times \cdots \times \tilde{F}_t \quad \neg C_{k+1} \wedge \cdots \wedge \neg C_n = \text{TRUE} \\
&\Leftrightarrow \quad \forall x \in [k+1,n] \forall f_x \in \tilde{F}_t \quad \neg C_x = \text{TRUE} \\
&\Leftrightarrow \quad \forall x \in [k+1,n] \forall f_x \in \tilde{F}_t \quad C_x = \text{FALSE} \\
&\Leftrightarrow \quad \forall x \in [k+1,n] \forall f_x \in \tilde{F}_t \quad C_0 \wedge C_x = \text{FALSE} \\
&\Leftrightarrow \quad P_1(M_x,\tilde{G}_t) = \emptyset
\end{aligned}
$$

7.4.6 Théorème 3

- Soit $M = ((v_1 : +,...,v_k : +,v_{k+1} : -,...,v_n : -),(c_1,...,c_m))$ un modèle de comportement à cardinal positif égal à $k, 0 < k < n$ tel que aucune des conditions $(c_1,...,c_m)$ implique deux variables typées d'un **-**.

– Soit $M_x = ((v_1 : +,...,v_k : +,v_x : +),(c_1,...,c_m))$ $\forall x \in [k+1,n]$ un modèle de comportement à cardinal positif égal à $k+1$.

SI $\forall x \in [k+1,n]$ $(f_1,\ldots,f_k,f_x) \in P_1(M_x,\tilde{G}_t)$

ALORS $(f_1,\ldots,f_k) \notin P_1(M,\tilde{G}_t)$

7.4.7 Preuve du théorème 3

$$
\begin{aligned}
& (f_1,...,f_k,f_x) \in P_1(M_x,\tilde{G}_t) \\
\Rightarrow\ & C_0 \wedge C_x = \text{TRUE} \\
\Rightarrow\ & C_0 \wedge \neg C_x = \text{FALSE} \\
\Rightarrow\ & C_0 \wedge \neg C_{k+1} \wedge \cdots \wedge \neg C_n = \text{FALSE} \\
\Rightarrow\ & (f_1,...,f_k) \notin P_1(M,\tilde{G}_t)
\end{aligned}
$$

En d'autres termes, si M n'a aucune condition portant sur deux variables typées - en même temps, le problème $P_1(M,\tilde{G}_t)$ peut être résolu comme une séquence de $P_1(M_x,\tilde{G}_t)$ tel que $Card^-(M_x) = 0$, c'est à dire une séquence de problème de satisfaction de contraintes.

7.5 Algorithme de Reconnaissance de Comportements

Nous détaillons dans cette section une solution algorithmique au problème $P_1(M,\tilde{G}_t)$. Comme nous l'avons vu dans la section précédente, ce problème correspond au problème $P_1(M,f(\tilde{G}_t))$ où $f(\tilde{G}_t)$ est un sous-graphe de $\tilde{G}_t$ dont les sommets ont un attribut *time* supérieur à $t - \Delta(M)$.

Nous avons vu, de plus, que la solution de $P_1(M,f(\tilde{G}_t))$ est la solution de $P_1(M_0,f(\tilde{G}_t))$, qui est un CSP, si tous les $P_1(M_x,f(\tilde{G}_t))$, qui sont aussi des CSPs n'ont pas de solution. Dans le cas contraire, toute solution de $P_1(M_x,f(\tilde{G}_t))$ peut être enlevée des solutions de $P_1(M_0,f(\tilde{G}_t))$.

Le principe de cet algorithme est de gérer un ensemble de réseaux de contraintes $\mathcal{R}_0,\mathcal{R}_{k+1},...,\mathcal{R}_n$ correspondant respectivement à $P_1(M_0,f(\tilde{G}_t))$, $P_1(M_{k+1},f(\tilde{G}_t))$,.., $P_1(M_n,f(\tilde{G}_t))$, afin de déterminer si $P_1(M_0,f(\tilde{G}_t))$ à des solutions. Si oui la solution de $P_1(M,f(\tilde{G}_t))$ est donnée par $P_1(M_0,f(\tilde{G}_t))$ moins les solutions de $P_1(M_x,f(\tilde{G}_t))$.

1. $D_i = f(\tilde{F}_t) \quad \forall i = 1,...,k$
2. **PROPAGER** $(c_1,...,c_m)$ sur $\mathcal{R}_0 = (D_0,...,D_k)$
3. **SI** $\exists D_i \in (D_0,...,D_k)$ tel que $|D_i| = 0$
4. **ALORS** $P_1(M_0,f(\tilde{G}_t)) = \emptyset \Rightarrow P_1(M,f(\tilde{G}_t)) = \emptyset$ **FIN**
5. **SINON** $\mathcal{L}$:= les solutions de $\mathcal{R}_0$
6. **POUR TOUT** x = k+1,..., n
7. $D_i^x = D_i \forall i = 1,...k$ et $D_x^x = f(\tilde{F}_t)$
8. **PROPAGER** $(c_1,...,c_m)$ sur $\mathcal{R}_x = (D_0^x,...,D_k^x,D_x^x)$
9. **SI** $\forall D_i \in (D_0^x,...,D_k^x,D_x^x)$ tel que $|D_i| \neq 0$
10. **ALORS**
11. **POUR TOUT** $(f_1,...f_k,f_x)$ solution de $\mathcal{R}_x$
12. $\mathcal{L} = \mathcal{L} \setminus (f_1,...f_k)$
13. **FIN DU SI**
14. **FIN DU FOR**
15. **FIN DU SI**
16. $P_1(M,f(\tilde{G}_t)) = \mathcal{L}$

La seule opération particulière de cet algorithme est l'opération **PROPAGER** qui consiste à appliquer à un réseau de contraintes un algorithme de consistance d'arc. Nous utilisons pour cela l'algorithme **AC4** de Mohr et Hendenson détaillé dans [72].

La complexité de cet algorithme dépend de la complexité de la propagation de contraintes, qui se trouve être dans notre cas $O(ea^2)$ où e est le nombre de conditions de M et a le nombre de sommets de $f(\tilde{G}_t)$. Soit e_0 le nombre de conditions de M_0 et e_x le nombre de conditions de M_x conditions. La complexité de notre algorithme est alors

$$O(e_0a^2 + \sum_{i=k+1,...,n} (e_x)a^2) \text{ C'est à dire } O(cea^2)$$

où $c = Card^-(M) + 1$

Appliquons cet algorithme au cas de la reconnaissance du modèle $M_{pedestrian\ appears}$ sur $\tilde{G}_{15}$ présenté dans la section précédente.

$$\begin{array}{rcl} \Delta(M_0) & = & 1 \\ D_0 & = & f(\tilde{F}_{15}) \quad = \quad \{o_6,p_6,p_{10}\} \end{array}$$

Après propagation de c_1 sur D_0,

$$\begin{aligned} D_0 &= \{p_6,p_{10}\} \\ \mathcal{L} &= \{(p_6),(p_{10})\} \\ \Delta(M_1) &= 2 \\ D_0^1 &= \{p_6,p_{10}\} \\ D_1^1 &= \{o_5,o_6,p_5,p_6,p_9,p_{10}\} \end{aligned}$$

Après propagation de (c_1,c_2,c_3,c_4) sur D_0^1,D_1^1,

$$\begin{aligned} D_0^1 &= \{p_6,p_{10}\} \\ D_1^1 &= \{p_5,p_9\} \end{aligned}$$

$\{(p_6,p_5),(p_{11},p_{10})\}$ est la solution de $P(M_1,\tilde{G}_{15})$
(p_6) et (p_{10}) sont enlevés de $\mathcal{L}$

$$\begin{aligned} D_0 &= \emptyset \\ P(M,\tilde{G}_{15}) &= \emptyset \end{aligned}$$

7.6 Conclusion

Nous avons présenté dans ce chapitre, la méthode utilisée pour la reconnaissance de comportements. Cette méthode est basée sur la reconnaissance d'un ensemble de modèles prédéfinis constituant la base de comportements.

La base de comportements est définie grâce à un formalisme permettant de gérer des conditions telles que les conditions spatiales, dynamiques, temporelles, symboliques ou logiques (voir section 7.1).

La reconnaissance de chacun des modèles à chaque frame par l'algorithme présenté dans la section 7.4 est basée sur une conversion de ces modèles en ensemble de problèmes de satisfactions de contraintes.

L'avantage de cette approche est en premier lieu son caractère déclaratif. Les comportements à reconnaître sont décrits de façon externe par rapport à l'algorithme qui les reconnaît. Ainsi, l'expert en charge de définir les comportements n'a pas à connaître la manière avec laquelle ils seront reconnus, mais seulement le formalisme de représentation.

Le second avantage, par comparaison avec l'ensemble des méthodes proposées dans la littérature, est l'expressivité du formalisme. Comme nous l'avons précédemment dit, cette approche est capable de gérer des conditions telles que les conditions spatiales, dynamiques, temporelles, symboliques ou logiques.

En ce qui concerne l'algorithme proposé pour la reconnaissance, son avantage par comparaison avec l'ensemble des méthodes proposées dans la littérature est son efficacité en terme de place mémoire nécessaire (c'est à dire une place mémoire quasiment nulle). En effet, il n'est pas question avec cette approche, de gérer un ensemble de modèles partiellement reconnus. Seul les instances des modèles complètement reconnus sont conservées dans le graphe d'interprétation.

Les inconvénients de l'approche sont les contreparties de ces avantages. En premier lieu, il est à remarquer que dans la mesure où rien n'est gardé en mémoire entre deux frames, une certaine partie de calcul est redondante d'une frame à l'autre.

Le second inconvénient de l'approche réside dans la construction des modèles. Nous pouvons remarquer qu'à la construction du modèle doit préexister une phase d'analyse et de formalisation. En effet, la construction d'un modèle requiert une expertise claire.

En outre, on note que la clarté de l'expertise ne suffit pas à une reconnaissance efficace. Les modèles, pour être reconnus de façon efficace, doivent assurer certaines propriétés. La taille *a priori* des domaines de définition des variables est l'une de ces propriétés.

De plus, il apparaît que la quasi-totalité des modèles nécessite certaines valeurs de seuil; valeurs de seuil qui sont souvent très symboliques De façon générale, la paramétrisation d'un modèle quelconque parait délicate.

Chapitre 8 Résultats de la reconnaissance de comportements

L'objectif de ce chapitre est de détailler différents cas d'application de notre approche pour la reconnaissance de comportements.

Bien que les différents cas d'applications présentés dans ce chapitre n'aient pas été réellement choisis, ceux-ci représentent un spectre assez large en termes d'objectifs applicatifs, de type de modèles, de complexité et qualité de résultats de suivi de personnes.

Nous présenterons cinq applications de notre approche. Le premier cas présente, dans la section 8.1, un ensemble de modèles de comportements de base tels que "entrer", "s'approcher de", "s'arrêter", etc... dont le but est de faciliter d'élaboration d'autres bases de comportements. Le second cas présenté dans la section 8.2, des résultats de notre approche pour la sécurité dans les stations de métro. Dans la même veine, le troisième exemple dans la section 8.3, est focalisé sur la sécurité en agences bancaires. Le quatrième exemple d'utilisation présenté dans la section 8.4, est relatif à une application d'aide au travail médiatisé. Le dernier exemple présenté dans la section 8.5, est une étude sur la reconnaissance de comportements basée sur l'interaction humains/objets dans les parkings.

8.1 Librairie de comportements de base

Le but d'une telle librairie est de rendre disponible un ensemble de modèles de comportements de base, afin de simplifier l'élaboration d'autres bases de modèles plus applicatives. Nous avons défini deux ensembles de 13 modèles. Les 13 premiers comportements de base, appelés comportements furtifs (*furtive*) représentent des changements entre deux instants consécutifs. Les seconds, appelés comportements persistants (*persistent*) représentent des changements entre deux instants consécutifs, mais qui persistent dans le temps pendant une durée prédéterminée. La liste des comportements de base furtifs ou persistants, que nous sommes capables de reconnaître est:*pedestrian moves close to an object* , *pedestrian moves away from an object*, *pedestrian moves close to a pedestrian*, *pedestrian moves away from a pedestrian*,*pedestrian enters an area*, *pedestrian leaves an area*, *pedestrian sits on an object*,*pedestrian falls down*, *pedestrian stands up*, *pedestrian stops*, *pedestrian starts*,*pedestrian appears* et *pedestrian disappears*.

$(x_0 : +, x_1 : +, x_2 : +, x_3 : +)$

$$
\begin{cases}
c_1) & category(x_0) = pedestrian \\
c_2) & category(x_1) = pedestrian \\
c_3) & category(x_2) = object \\
c_4) & category(x_3) = object \\
c_5) & name(x_1) = name(x_0) \\
c_6) & name(x_3) = name(x_2) \\
c_7) & time(x_2) = time(x_0) \\
c_8) & time(x_3) = time(x_1) \\
c_9) & time(x_1) - time(x_0) = 1 \\
c_{10}) & time(x_3) - time(x_2) = 1 \\
c_{11}) & distance(hull(x_0), hull(x_2)) > \Delta_1 \\
c_{12}) & distance(hull(x_1), hull(x_3)) < \Delta_2
\end{cases}
$$

$$
\begin{aligned}
Card^+(furtive\ pedestrian\ moves\ close\ to\ an\ object) &= 4 \\
Card^-(furtive\ pedestrian\ moves\ close\ to\ an\ object) &= 0 \\
\Delta(furtive\ pedestrian\ moves\ close\ to\ an\ object) &= 2
\end{aligned}
$$

FIG. 8.1 – *Modèle de* furtive pedestrian moves close to an object

$$
\begin{array}{l}
\qquad (x_0 : +, x_1 : -) \\
\left\{ \begin{array}{lrcl}
c_1) & category(x_0) & = & \textit{furtive pedestrian moves close to an object} \\
c_2) & category(x_1) & = & \textit{furtive pedestrian moves away from an object} \\
c_3) & time(x_1) & - & time(x_0) < \Delta_3 \\
c_4) & name(ref(x_0)) & = & name(ref(x_1))
\end{array} \right.
\end{array}
$$

$$
\begin{array}{rcl}
Card^+(\textit{persistent pedestrian moves close to an object}) & = & 1 \\
Card^-(\textit{persistent pedestrian moves close to an object}) & = & 1 \\
\Delta(\textit{persistent pedestrian moves close to an object}) & = & \Delta_{PERSISTENCE}
\end{array}
$$

FIG. 8.2 – *Modèle de* persistent pedestrian moves close to an object

8.1.1 Représentation de Comportements de Base

Nous ne souhaitons pas donner la liste exhaustive de tous ces modèles , mais nous allons à titre explicatif, détailler la construction de *pedestrian moves close to an object* (montré par la figure 8.1 en version *furtive* et montrée par la figure 8.2 en version *persistent*). Le modèle de *furtive pedestrian moves close to an object* est constitué de 12 conditions basées sur 4 variables typées +. Les 2 premières variables représentent 2 *pedestrians* (cf. c_1 et c_2) présents à deux instants consécutifs (cf. c_9). De plus, ces 2 *pedestrians* ont le même *name* (cf. c_5), c'est à dire que ces deux variables représentent le même humain réel. De même, les 2 dernières variables représentent 2 *objects* (cf. c_3 et c_4) présents à deux instants consécutifs (cf. c_{10}) ayant le même *name* (cf. c_6). Finalement, c_{11} et c_{12} spécifient le fait que la distance entre le premier *pedestrian* et le premier *object* doit être inférieure à Δ_1 centimètres et le fait que la distance entre le second *pedestrian* et le second *object* doit être supérieure à Δ_2 centimètres.

Le modèle de *persistent pedestrian moves close to an object* est constitué de 4 conditions utilisant 2 variables: x_0 typé + et x_1 typé -. La première variable représente une instance du comportement *furtive pedestrian moves close to an object* et la seconde une instance de *furtive pedestrian moves away from an object*. C'est à dire qu'une instance de *persistent pedestrian moves close to an object* peut être trouvée si après un délai $\Delta_{PERSISTENCE}$ commençant à la création d'une instance de *furtive pedestrian moves close to an object*, il n'existe toujours pas d'instance de *furtive pedestrian moves away from an object* tel que les attributs *name* de leurs *references* soient les mêmes.

8.1.2 Reconnaissance de comportements de base

L'enjeu d'une telle librairie, à part être réutilisable, est de rester efficace même si les attributs des sommets de $\tilde{G}_t$ sont bruités. Nous présentons ici quelques résultats de tests de résistance au bruit des modèles de cette librairie.

Le protocole est d'altérer l'attribut *hull* de chaque sommet représentant un *pedestrian* d'un graphe idéal, sur lequel un modèle donné M a eu une instance i_M. La corruption des sommets est faite par un bruit aléatoire additif, dont on contrôle l'amplitude. Cette amplitude est augmentée jusqu'à ce que l'algorithme ne reconnaisse plus i_M au bon instant. Nous recalculons la *velocity*, afin d'avoir une cohérence entre *hull* et *velocity*. La valeur de $\Delta_{PERSISTENCE}$ de tous les modèles de comportements persistants est mise à 5 frames.

Nous avons testé chaque modèle 20 fois pour chaque valeur d' amplitude et pour 30 valeurs différentes d'amplitude. Les résultats dépendent de l'amplitude de ce bruit; c'est pourquoi nous organisons ces résultats autour de l'amplitude du bruit. La table 8.1 présente des tests faits sur les comportements *furtive* et 8.2 présente des essais faits sur les comportements *persistent*.

La colonne 2 (resp. 3, 4 et 5) représente le pourcentage du cas où le modèle est reconnu au bon instant avec l'amplitude de bruit correspondant à 0,2 mètre (resp. 0,5, 1, 2 mètres) soit approximativement 2 % (resp. 5 %, 10 % et 20 %) de la taille moyenne de la scène (%*size*). Par exemple, considérons le *pedestrian stands up*. Si l'amplitude du bruit est de 1 mètre, alors l'instance originale est reconnue dans 25% des cas.

La première conclusion à tirer de ces résultats est, comme on pouvait s'y attendre, que les modèles de comportements persistants sont plus résistants au bruit que les modèles furtifs. En effet, le taux de reconnaissance exacte chute de 10 % avec les modèles de comportements furtifs par rapport aux modèles de comportements persistants.

La seconde conclusion à tirer est le caractère hétérogène des résultats. De ce point de vue, deux catégories de modèles se distinguent à l'intérieur de chacune des deux familles. La première catégorie composée des modèles *pedestrian stops*, *pedestrian starts*, *pedestrian moves close to a pedestrian* et *pedestrian moves away from a pedestrian* parait moins résistante que les autres modèles. Ce point peut être expliqué par le fait que *pedestrian stops*, *pedestrian starts* sont basés sur la vitesse et non la position des personnes et par le fait que *pedestrian moves close to a pedestrian* et *pedestrian moves away from a pedestrian* cumulent deux sources de bruit (une par *pedestrian*). Alors que les autres modèles sont basés sur le changement de distance entre un *pedestrian* et un *object*, c'est à dire une seule source de bruit.

Modèles de comportements de base (*furtive*)	0.2 mètre $\simeq$ 2%*scène*	0.5 mètre $\simeq$ 5%*scène*	1 mètre $\simeq$ 10%*scène*	2 mètres $\simeq$ 20%*scène*
pedestrian stops	95 %	60 %	33 %	17 %
pedestrian starts	88 %	60 %	39 %	26 %
pedestrian falls	100 %	100 %	79 %	48 %
pedestrian stands up	65 %	40 %	25 %	15 %
pedestrian enters an area	85 %	68 %	61 %	56 %
pedestrian leaves an area	58 %	32 %	17 %	9 %
pedestrian moves close to a pedestrian	90 %	72 %	49 %	26 %
pedestrian moves away from an object	83 %	66 %	57 %	40 %
pedestrian moves away from a pedestrian	88 %	82 %	48 %	25 %

TAB. 8.1 – *Résultats en termes de résistance au bruit de la reconnaissance de comportements de base furtifs*

8.1.3 Conclusion

Nous avons présenté, dans cette section, une librairie de comportements de base. La diversité de ces comportements, ainsi que le caractère générique de chacun, nous permet de pouvoir construire d'autre bases de comportements sans avoir à repartir de zéro. De plus, la robustesse de la majorité des modèles, nous permet de s'affranchir des erreurs commises par l'algorithme de suivi de personnes.

Modèles de comportement de base (*persitent*)	0.2 mètre $\simeq$ 2%*scène*	0.5 mètrw $\simeq$ 5%*scène*	1 mètre $\simeq$ 10%*scène*	2 mètres $\simeq$ 20%*scène*
pedestrian stops	95 %	66 %	44 %	28 %
pedestrian starts	88 %	61 %	39 %	26 %
pedestrian falls	100 %	99 %	90 %	72 %
pedestrian stands up	65 %	40 %	25 %	15 %
pedestrian enters an area	90 %	70 %	63 %	52 %
pedestrian leaves an area	76 %	50 %	27 %	14 %
pedestrian moves close to an object	98 %	82 %	57 %	37 %
pedestrian moves away from an object	83 %	66 %	48 %	30 %
pedestrian moves away from a pedestrian	86 %	66 %	43 %	26 %

TAB. 8.2 – *Résultats en termes de résistance au bruit de la reconnaissance de comportements de base persistants*

8.2 Résultats de la représentation et de la reconnaissance de comportements pour la sécurité dans les stations de métro

8.2.1 Représentation de comportements pour la sécurité dans les stations de métro

Nous présentons dans cette section des résultats de notre approche dans le cadre de la sécurité dans les stations de métro.

Le problématique de la sécurité dans les stations de métro, comme elle a été exposée par les experts de sécurité des métros de Nuremberg, Bruxelles et Charleroi du projet européen AVS-PV [1] et dans [55] se compose de trois objectifs: la protection des personnes contre les personnes (liée aux comportements de violence), la protection des personnes contre elles-mêmes (liée à l'accès aux zones dangereuses, comme des rails), la protection des équipements contre les personnes: (associée aux comportements de vandalisme et de graffiti). Nous ne traiterons pas la première catégorie de comportements liés à la violence.

En utilisant la librairie de comportements de base, nous avons construit un ensemble de modèles permettant de reconnaître des comportements de la seconde et de la troisième catégorie. La deuxième catégorie de comportements est caractérisée par la présence des personnes dans une zone dangereuse prédéfinie pendant une certaine durée. Le modèle de ce comportement montrée par la figure 8.4, est établi en utilisant le comportement de base *pedestrian entre dans une zone*. La dernière catégorie de comportements, montré par la figure 8.3, se caractérise par la présence de personnes près de certains équipements *fragiles* pendant une période de temps plus longue que la durée normale prédéfinie.

1. Advanced Video Surveillance for Prevention of Vandalism

$(x_0 : +, x_1 : +, x_2 : -)$

$$\begin{cases} c_1) & category(x_0) & = & \textit{furtive pedestrian moves close to an object} \\ c_2) & category(x_1) & = & \textit{object} \\ c_3) & ref(x_0) & = & x_1 \\ c_4) & property(x_1) & = & \textit{fragile} \\ c_5) & category(x_2) & = & \textit{furtive pedestrian moves away from an object} \\ c_6) & time(x_2) & - & time(x_0) < \Delta_{VANDALISM} \\ c_7) & name(ref(x_0)) & = & name(ref(x_2)) \end{cases}$$

$$\begin{aligned} Card^+(VANDALISM) &= 2 \\ Card^-(VANDALISM) &= 1 \\ \Delta(VANDALISM) &= \Delta_{VANDALISM} \end{aligned}$$

FIG. 8.3 – *Le modèle de* VANDALISM *est composé de 7 conditions basées sur 3 variables. La première variable représente une instance de* furtive pedestrian moves close to an object *(cf.* c_1*). La seconde variable représente un* object fragile *(cf.*c_2 *et* c_4*), qui est une des références de la première variable (cf.* c_3*). La dernière variable, typée* **-** *est une instance de* furtive pedestrian moves away from an object *ayant les mêmes* name *de références que la première variable,* $\Delta_{VANDALISM}$ *plus tard (cf.* c_5,c_6 *et* c_7*). En d'autres termes,* VANDALISM *sera reconnu si après* $\Delta_{VANDALISM}$ *frames à partir de la création d'un sommet* furtive pedestrian moves close to an object *dont l'objet porte la propriété* fragile, *aucune instance de* furtive pedestrian moves away from an object *ayant les mêmes références n'a été reconnue.*

$(x_0 : +, x_1 : +, x_2 : -)$

$$\begin{cases} c_1) & category(x_0) & = & \textit{furtive pedestrian enters a aera} \\ c_2) & category(x_1) & = & \textit{object} \\ c_3) & ref(x_0) & = & x_1 \\ c_4) & property(x_1) & = & \textit{dangerous} \\ c_5) & category(x_2) & = & \textit{furtive pedestrian leaves an area} \\ c_6) & time(x_2) & - & time(x_0) < \Delta_{DANGER} \\ c_7) & name(ref(x_0)) & = & name(ref(x_2)) \end{cases}$$

$$\begin{aligned} Card^+(DANGER) &= 2 \\ Card^-(DANGER) &= 1 \\ \Delta(DANGER) &= \Delta_{DANGER} \end{aligned}$$

FIG. 8.4 – *Le modèle de* DANGER *est composé de 7 conditions basés sur 3 variables. La première variable représente une instance de* furtive pedestrian enters an area *(cf.* c_1*). La seconde variable représente un* object dangerous *(cf.*c_2 *et* c_4*), qui est une des références de la première variable (cf.* c_3*). La dernière variable, typée* - *est une instance de* furtive pedestrian moves away from an object *ayant les mêmes* name *de références que la première variable,* Δ_{DANGER} *plus tard (cf.* c_5,c_6 *et* c_7*). En d'autres termes,* DANGER *sera reconnu si après* Δ_{DANGER} *frames à partir de la création d'un sommet* furtive pedestrian enters an area *dont l'objet porte la propriété* dangerous*, aucune instance de* furtive pedestrian leaves an area *ayant les mêmes références n'a été reconnue.*

8.2.2 Reconnaissance de comportements pour la sécurité dans les stations de métro

Les figures 8.5, 8.6, 8.7, 8.8 et 8.9 sont une partie de la vidéo VA2-6 (cf. 8.2.3) filmée dans une station du métro de Nuremberg et illustrent la reconnaissance d'un comportement de vandalisme. Chaque figure se compose de 3 images. La première du côté gauche est l'image réelle issue du flux vidéo. Les deux autres (au centre et à droite) sont des images de la reconstruction au même instant. Au centre, le point de vue est le même que la caméra et, à droite, nous avons une vue du haut de la scène. Nous pouvons voir, sur les vues de la reconstruction en gris, en vert et en orange les éléments de l'environnement: les sommets O_t de $\bar{G}_t$. les sommets P_t de $\bar{G}_t$ (les humains) sont montrés en jaune. Les arcs $\bar{T}_t$ de $\bar{G}_t$ (les relations temporelles) sont montrés avec les lignes rouges. $\bar{V}_t$ de $\bar{G}_t$ (les sommets de comportements) sont représentés de deux façons. Ils sont représentés avec de petites flèches oranges et rouges sur la reconstruction et représentés avec de petits commentaires oranges et rouges en haut des images réelles. La couleur orange est associée à un reconnaissance partielle et la couleur rouge est associée à une reconnaissance complète. Conscient que ces flèches et commentaires ne sont pas vraiment lisibles, nous reportons à chaque fois la reconnaissance de comportements sur la légende de la figure.

Les figures 8.10, 8.11, 8.12, 8.13 et 8.14 sont une partie de la vidéo ST1-23 (cf. 8.2.3) filmée sur une station du métro de Bruxelles et illustrent les reconnaissance des comportements d'accès à une zone dangereuse et de vandalisme. Le sémantique des figures est identique aux précédentes, excepté que tous les éléments de l'environnement sont en gris.

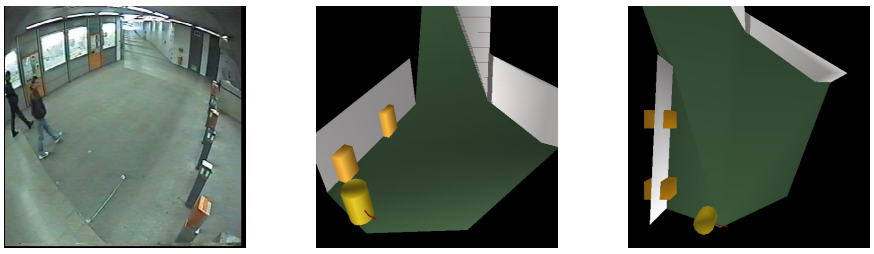


FIG. 8.5 – *Dans une station de métro de Nuremberg, deux humains h_1 et h_2 entrent dans la scène par la droite.*

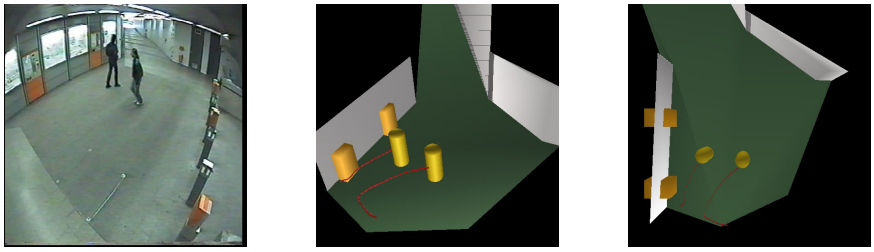


FIG. 8.6 – h_1 *et* h_2 *contrôlent que le couloir est vide.*

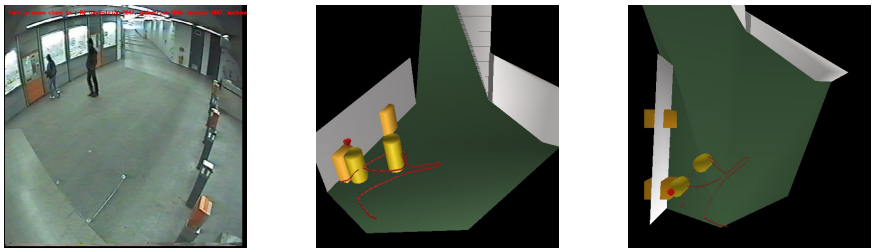


FIG. 8.7 – h_1 *s'approche du distributeur de billets A (à droite dans la scène). Le comportement* pedestrian moves close to an object *est reconnu.*

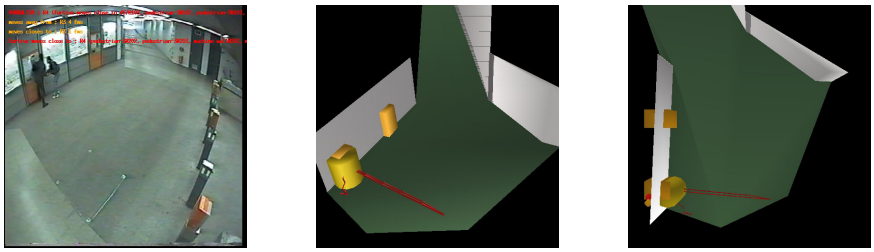


FIG. 8.8 – $\Delta_{VANDALISM}$ *frames plus tard, (*$\Delta_{VANDALISM}$*=150), aucun* pedestrian moves close to an object *n'a été reconnu impliquant* h_1 *et cette machine, le comportement* VANDALISM *est reconnu N.B. On peut observer, en bas de la reconstruction, une erreur typique de suivi de personnes causée par une courte perte de détection cumulée à un bruit reconnu comme un humain.*

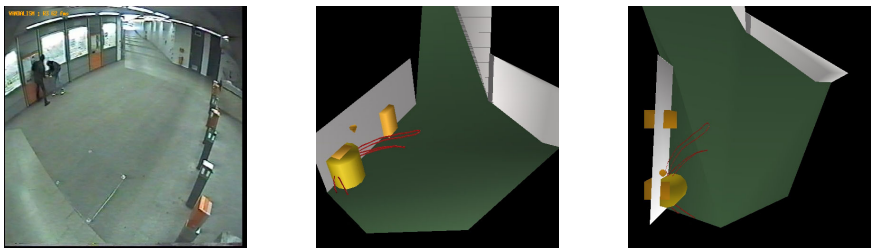


FIG. 8.9 – *Fin*

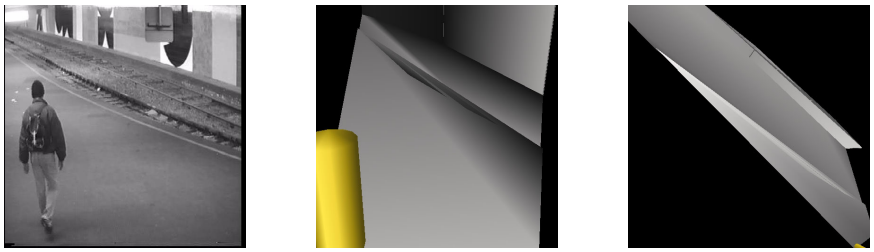


FIG. 8.10 – *Dans une station de métro de Bruxelles, un humain h_1 entre dans la scène.*

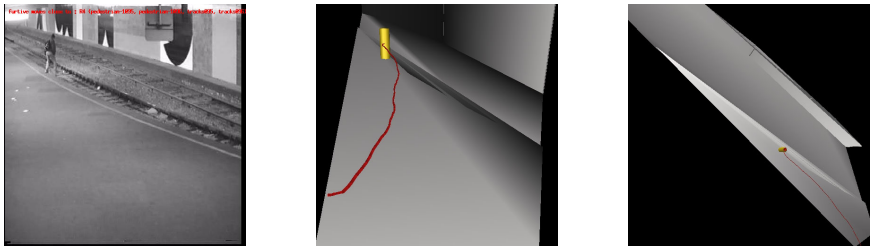


FIG. 8.11 – h_1 *marche jusqu'au bord, alors le comportement* pedestrian enters an area *est reconnu.*

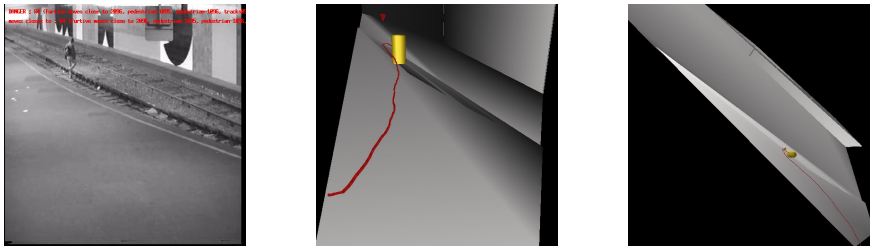


FIG. 8.12 – Δ_{DANGER} *frames plus tard* (Δ_{DANGER}=5)*, aucun* pedestrian leaves an area *n'a été reconnu impliquant* h_1*, le comportement* DANGER *reconnu à son tour.*

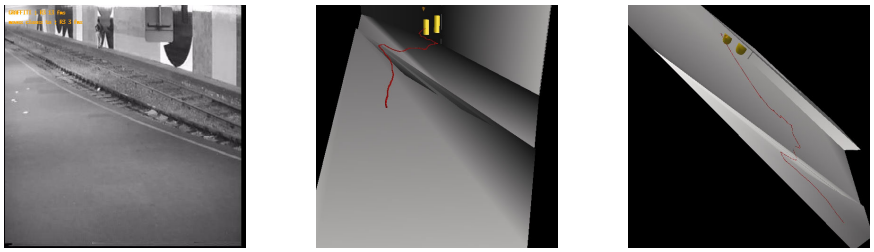


FIG. 8.13 – h_1 *s'approche du mur du fond* pedestrian moves close to an object *est reconnu. N.B. On peut voir sur les deux reconstructions une erreur typique de reconnaissance de personne causée par la réflexion de* h_1 *sur le mur.*

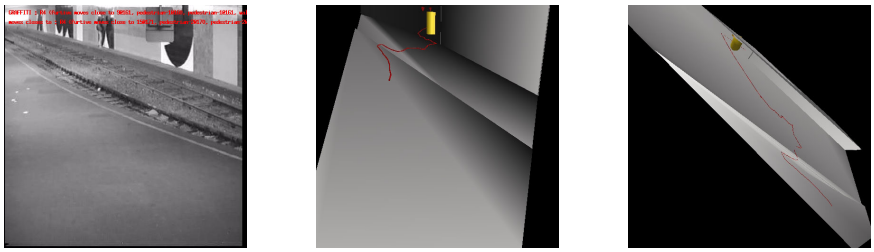


FIG. 8.14 – $\Delta_{GRAFFITI}$ *frames plus tard* ($\Delta_{GRAFFITI}$=*50), aucun* pedestrian away from an object *n'a été reconnu impliquant* h_1 *et le mur, le comportement* GRAFFITI *est reconnu.*

8.2.3 Résultats quantitatifs de la reconnaissance de comportements pour la sécurité dans les stations de métro

On peut voir qu'il n'y a aucune erreur commise dans le cas de données segmentées à la main (sur la table 8.4), exceptée une instance de *GRAFFITI* (qui n'est pas problématique). Les différences entre la vérité terrain (sur la table 8.3) et les résultats obtenus sur des données réelles (sur la table 8.5) sont de deux types: on peut, en premier lieu, observer un délai, allant de 1 frame (*VANDALISM* sur VA2-6) à 15 frames (*VANDALISM* sur VA2-4, qui est très bruité). Le second point est l'erreur commise sur la vidéo ST1-23. deux fausses reconnaissances de *GRAFFITI* apparaissent à la frame 179 et 186. Ceci est dû à la réflexion d'une personne traitée comme étant une autre personne (voir figure 8.13)

	VA2-7	VA2-4	VA2-6	ST1-23
VANDALISM		[260,340]	[140,320], [200,320]	
DANGER				[95,190]
GRAFFITI				[180,190]

TAB. 8.3 – *La vérité terrain de la reconnaissance de comportements pour la sécurité dans les stations de métro. Chaque occurrence d'un comportement particulier est représentée par l'intervalle de temps correspondant à sa durée.*

	VA2-7	VA2-4	VA2-6	ST1-23
VANDALISM		270,	148, 204	
DANGER				100
GRAFFITI				180, 186

TAB. 8.4 – *Résultats de la reconnaissance de comportements pour la sécurité dans les stations de métro obtenus à partir de données segmentées à la main. Chaque comportement particulier est représenté par l'instant où il est reconnu.*

8.2.4 Conclusion

Nous avons présenté dans cette section, les résultats de notre approche dans le cadre de la sécurité dans les stations de métro.

	VA2-7	VA2-4	VA2-6	ST1-23
VANDALISM		285	163, 203	
DANGER				101
GRAFFITI				176, 179, 184, 186

TAB. 8.5 – *Résultats de la reconnaissance de comportements pour la sécurité dans les stations de métro obtenue à partir de données réelles. Chaque comportement particulier est représenté par l'instant où il est reconnu.*

8.3 Résultats de la représentation et reconnaissance de comportements pour la sécurité dans les agences bancaires

8.3.1 Représentation de comportements pour la sécurité dans les agences bancaires

Le principe de la sécurité en agence bancaire est de se préserver des attaques. L'attaque consiste en une intrusion dans la banque d'une ou plusieurs personnes afin d'y récupérer des fonds qui s'y trouvent. Nous introduisons la notion de rôle afin de différencier trois classes de personnes impliquées. Les employés de l'agence, les clients de l'agence et les agresseurs. Le problème de la sécurité dans les agences bancaires consiste alors à déterminer si une personne entrant dans la banque est un client ou un agresseur par l'analyse de son comportement.

On distingue dès lors trois types d'attaques en fonction de l'endroit ou se trouvent les fonds recherchés : l'attaque guichet, l'attaque du local automate et l'attaque de coffre. Nous ne nous intéresserons bien évidement pas à la troisième catégorie d'attaque puisque celle ci ne relève pas du comportement visuel. Nous appellerons dans la suite l'objectif l'endroit logique où se trouvent les fonds recherchés par le ou les agresseurs. L'objectif de l'attaque induit en premier lieu la durée de l'attaque. De ce point de vue, l'attaque guichet est de l'ordre de quelques secondes, alors que l'attaque du local automate est de l'ordre de quelques minutes.

Le second point est les moyens de l'attaque. C'est à dire comment le ou les agresseurs vont assurer la réussite de leur attaque. Les moyens dépendent de la durée de l'attaque, donc de l'objectif. Dans le cas de l'attaque guichet, le contrôle de l'ensemble des presonnes présentes peut être totalement inexistant. Dans le cas de l'attaque automate donc de longue durée , les agresseurs doivent contrôler l'agence et ses occupants. Le contrôle des occupants de l'agence est fait soit en regroupant les clients dans un endroit contrôlable (c'est à dire un bureau, un coin de l'agence), soit en immobilisant les clients (au sol par exemple). Le contrôle de l'agence passe aussi par le contrôle des entrées de nouveaux clients. Generalement, l'attitude de l'agresseur consiste soit à regrouper les clients arrivant avec les autres occupants déjà contrôlé, soit à immobiliser les clients arrivant hors de vue de l'extérieur de l'agence. On appellera le "pool" l'ensemble des occupants contrôlés de l'agence ou "contrainte sur un groupe de personnes".

Dans les deux types d'attaque envisagés, les agresseurs peuvent assurer leur sortie d'agence par un ou des otages (appelée sortie sous la contrainte) qui peuvent être soit les employés soit des clients. Cette caractérisation est résumée dans les tableaux 8.6 et 8.7.

Dès lors, les possibilités de reconnaissance se différencient par les moyens et les objectifs.

- Reconnaissance de comportements liés aux objectifs

l'attaque guichet	
caractérisée par	**aggravée par**
• la mise en relation furtive entre un agresseur et un employé	• une contrainte sur un client ou sur un employé • une sortie sous la contrainte d'un client ou d'un employé

TAB. 8.6 – *Caractérisation de l'attaque guichet*

l'attaque automate	
caractérisée par	**aggravée par**
• la mise en relation furtive entre un agresseur et un employé • le contrôle de l'agence par : → la création d'un pool → le contrôle distant	• une contrainte sur un client ou sur un employé • une sortie sous la contrainte d'un client ou d'un employé

TAB. 8.7 – *Caractérisation de l'attaque automate*

Si l'objectif de l'attaque est le local automate, une condition nécessaire et suffisante est l'accès d'un client (i.e. agresseur) dans une zone réservée aux employés. Si l'objectif de l'attaque est le guichet, les conditions nécessaires ne sont pas suffisantes pour discriminer le comportement du client de celui de l'agresseur.

- Reconnaissance de comportements liés aux moyens
  Les comportements liés aux moyens sont nombreux :
  - accès à une zone réservée (arrière guichet, accès aux coffres) dont le modèle est montré par la figure 8.15,
  - contrainte sur un groupe dont le modèle est montré par la figure 8.17,
  - contrainte sur une personne dont le modèle est montré par la figure 8.16,
  - sortie sous la contrainte dont le modèle est similaire au précédent.

$(x_0 : +, x_1 : +, x_2 : -)$

$$
\begin{array}{rrcl}
c_1) & category(x_0) & = & \textit{furtive pedestrian enters an area} \\
c_2) & category(x_1) & = & \textit{object} \\
c_3) & ref(x_0,3) & = & x_1 \\
8\ c_4) & property(x_1,0) & = & \textit{restricted} \\
c_5) & category(x_2) & = & \textit{furtive pedestrian leaves an area} \\
c_7) & time(x_2) - time(x_0) & < & \Delta_{RESTRICTED} \\
c_8) & name(ref(x_0)) & = & name(ref(x_2))
\end{array}
\quad .
$$

$$
\begin{aligned}
Card^{+}(RESTRICTED) &= 2 \\
Card^{-}(RESTRICTED) &= 1 \\
\Delta(RESTRICTED) &= \Delta_{RESTRICTED}
\end{aligned}
$$

FIG. 8.15 – *Le modèle de* RESTRICTED *est composé de 7 conditions basées sur 3 variables. La première variable représente une instance de* furtive pedestrian enters an area *(cf.* c_1*). La seconde variable représente un* object restricted *(cf.*c_2 *et* c_4*), qui est une des références de la première variable (cf.* c_3*). La dernière variable, typée* - *est une instance de* furtive pedestrian moves away from an object *ayant les mêmes* name *de référence que la première variable,* $\Delta_{RESTRICTED}$ *plus tard (cf.* c_5,c_6 *et* c_7*). En d'autres termes,* RESTRICTED *sera reconnu si après* $\Delta_{RESTRICTED}$ *frames à partir de la création d'un sommet* furtive pedestrian enters an area *dont l'objet porte la propriété* restricted*, aucune instance de* furtive pedestrian leaves an area *ayant les mêmes références n'a été reconnue.*

$$
\begin{array}{l}
(x_0 : +, x_1 : -) \\
\left.\begin{array}{lrcl}
c_1) & category(x_0) & = & \textit{furtive pedestrian moves close to a pedestrian} \\
c_2) & category(x_1) & = & \textit{furtive pedestrian moves away from a pedestrian} \\
c_3) & time(x_1) & - & time(x_0) < \Delta_{HOSTAGE} \\
c_4) & name(ref(x_0)) & = & name(ref(x_1))
\end{array}\right\} 8 \\
\end{array}
.
$$

$$
\begin{aligned}
Card^{+}(HOSTAGE) &= 1 \\
Card^{-}(HOSTAGE) &= 1 \\
\Delta(HOSTAGE) &= \Delta_{HOSTAGE}
\end{aligned}
$$

FIG. 8.16 – *Le modèle de* HOSTAGE *est composé de 4 conditions basées sur 2 variables. La première variable représente une instance de* furtive pedestrian moves close to a pedestrian *(cf.* c_1*). La dernière variable est une instance de* furtive pedestrian moves away from an object *ayant les mêmes* name *de référence que la première variable,* $\Delta_{HOSTAGE}$ *plus tard (cf.* c_5,c_6 *et* c_7*). En d'autres termes,* HOSTAGE *sera reconnu si après* $\Delta_{HOSTAGE}$ *frames à partir de la création d'un sommet* furtive pedestrian moves close to a pedestrian *aucune instance de* furtive pedestrian moves away from a pedestrian *ayant les mêmes références n'a été reconnue.*

$(x_0 : +, x_1 : +, x_2 : -)$

$$
\begin{array}{rrcl}
c_1) & category(x_0) & = & \textit{furtive pedestrian moves close to a static pedestrian} \\
c_2) & category(x_1) & = & \textit{pedestrian stops} \\
c_3) & name(ref(x_0,0)) & = & name(ref(x_1,0)) \\
c_4) & category(x_2) & = & \textit{pedestrian moves away from a pedestrian} \\
c_5) & name(ref(x_2,0)) & = & name(ref(x_0,0)) \\
c_6) & name(ref(x_2,1)) & = & name(ref(x_1,0)) \\
c_7) & time(x_2) & - & time(x_0) < \Delta_{POOLING}
\end{array}
$$

$$
\begin{aligned}
Card^+(POOLING) &= 2 \\
Card^-(POOLING) &= 1 \\
\Delta(POOLING) &= \Delta_{POOLING}
\end{aligned}
$$

FIG. 8.17 – *Le modèle de* POOLING *est composé de 7 conditions basées sur 3 variables. La première variable représente une instance de* furtive pedestrian moves close to a static pedestrian, *qui est très similaire à* pedestrian moves close to a pedestrian *(cf.* c_1*). La seconde variable représente une instance de* pedestrian stops *dont les références sont les même que la première variable. (cf.*c_2,c_3 *et* c_4*). La dernière variable, typée* - *est une instance de* furtive pedestrian moves away from a pedestrian *ayant les mêmes* name *de références que la première variable,* $\Delta_{POOLING}$ *frames plus tard (cf.* c_5,c_6 *et* c_7*). En d'autres termes,* POOLING *sera reconnu si après* $\Delta_{POOLING}$ *frames à partir de la création d'un sommet* furtive pedestrian moves close to a static pedestrian , *une instance de* pedestrian stops *a été reconnue et aucune instance de* furtive pedestrian moves away from a pedestrian *ayant les mêmes références n'a été reconnue.*

8.3.2 Reconnaissance de comportements pour la sécurité dans les agences bancaires

Les figures 8.18, 8.19, 8.20, 8.21 et 8.22 sont prises de la vidéo MC2-17 (cf. 8.3.3) filmée dans une agence bancaire de la caisse régionale de la Brie et illustrent la reconnaissance d'un comportement de contrainte sur une personne (*HOSTAGE*), où la personne est un client Les figures 8.23, 8.24, 8.25, 8.26 et 8.27 sont prises de la vidéo MC1-22 (cf. 8.3.3) filmée dans une agence bancaire de la Caisse régionale de la Brie illustre d'une part la reconnaissance d'uncomportement de contrainte sur une personne (*HOSTAGE*), ou la personne est un banquier et la reconnaissance d'un accès à une zone réservée (*RESTRICTED*). Chaque figure se compose de 3 images. La première du côté gauche est la vraie image prise du flux vidéo. Les deux autres (au centre et à droite) sont des images de la reconstruction au même instant. Au centre, le point de vue est le même que la caméra et, à droite, nous avons une vue du haut de la scène. Nous pouvons voir, sur les vues de la reconstruction en gris, en vert et en orange les éléments de l'environnement: les sommets O_t de $\bar{G}_t$. les sommets P_t de $\bar{G}_t$ (les humains) sont montrés en jaune. Les arcs $\bar{T}_t$ de $\bar{G}_t$ (les relations temporelles) sont montrés avec les lignes rouges. $\bar{V}_t$ de $\bar{G}_t$ (les sommets de comportements) sont représentés de deux façons. Ils sont représentés avec de petites flèches oranges et rouges sur la reconstruction et représentés avec de petits commentaires oranges et rouges en haut des images réelles. La couleur orange est associée à un reconnaissance partielle et la couleur rouge est associée à une reconnaissance complète. Conscient que ces flèches et commentaires ne sont pas vraiment lisible, nous reportons à chaque fois la reconnaissance de comportements sur la légende de la figure.

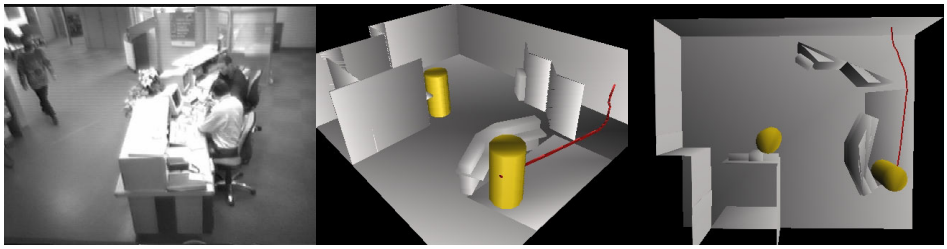


FIG. 8.18 – *Dans une agence bancaire de la Caisse régionale de la Brie , deux banquiers h_2 (au premier plan) et h_1 (au fond) sont en train de travailler derrière le guichet, quand un client h_3 entre dans la scène (par la gauche). N.B. A cause d'une mauvaise initialisation combinée avec le fait que h_2 occulte h_1, h_2 n'est pas reconnu.*

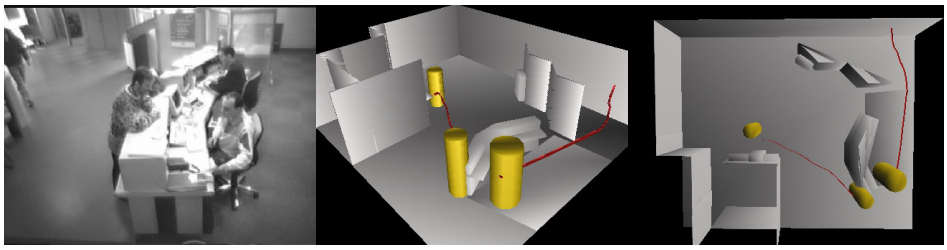


FIG. 8.19 – *h_3 s'approche du guichet. Un nouvel humain h_4 entre dans la scène (à gauche).*

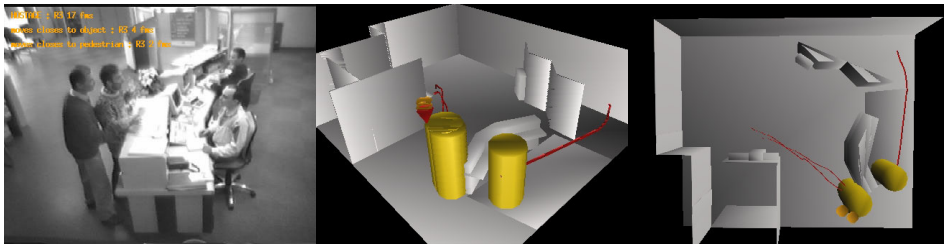


FIG. 8.20 – *h_4 s'approche de h_3, un comportement* pedestrian moves close to a pedestrian *est alors reconnu.*

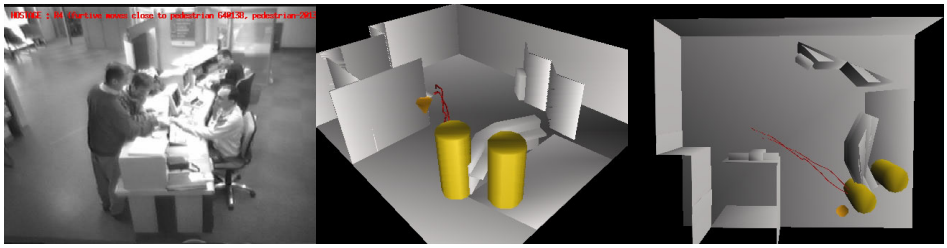


FIG. 8.21 – *$\Delta_{HOSTAGE}$ frames plus tard, ($\Delta_{HOSTAGE}$=20), un comportement* HOSTAGE *est reconnu. h_3 et h_4 sont superposés sur la reconstruction mais toujours présents.*

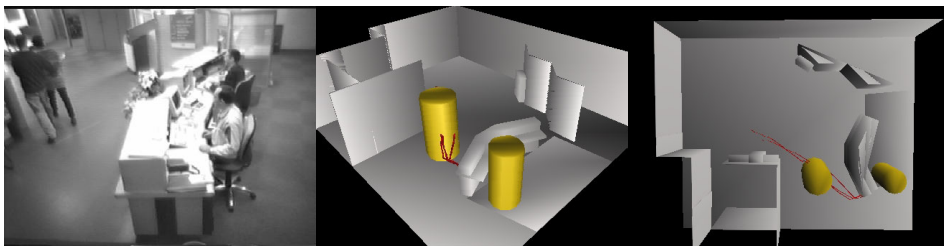


FIG. 8.22 – *h_4 sort de l'agence avec h_3*

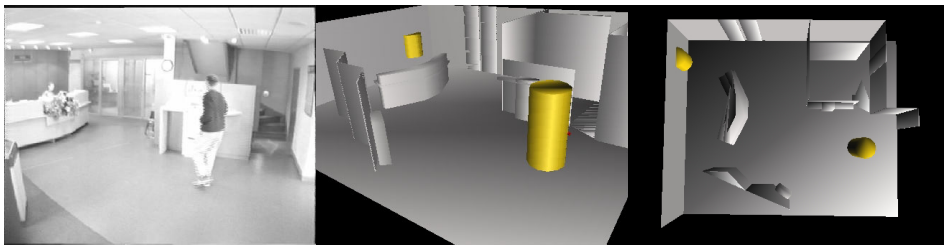


FIG. 8.23 – *Dans une agence bancaire de la Caisse régionale de la Brie , un banquier h_1 est en train de travailler au guichet, lorsqu'un client h_2 entre dans la scène (à droite). N.B. On peut observer que la localisation de h_1 est partiellement fausse, à cause de l'occultation avec le guichet.*

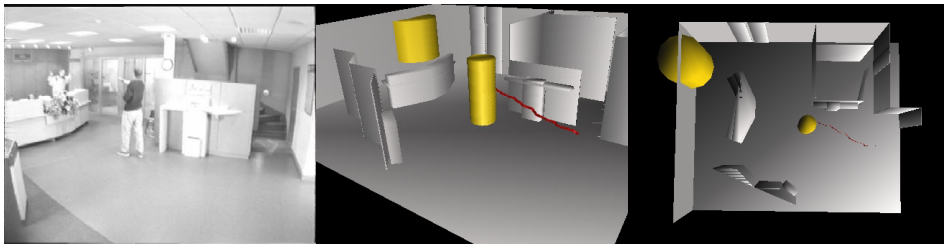


FIG. 8.24 – *h_2 s'approche du guichet et menace h_1 avec une arme. h_1 se lève.*

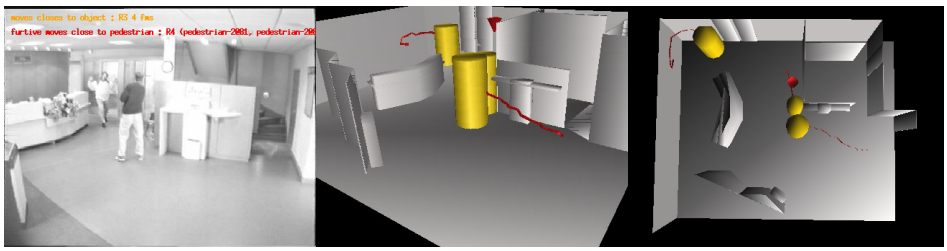


FIG. 8.25 – h_1 pedestrian moves close to a pedestrian h_2. *N.B. On peut observer que l'ombre de h_2 sur le mur à sa droite est reconnue comme un humain. Cette erreur de suivi peut être considérée, du point de vue de la reconnaissance de comportements comme du bruit. Malgré tout, ce bruit ne perturbe pas la reconnaissance du modèle* HOSTAGE.

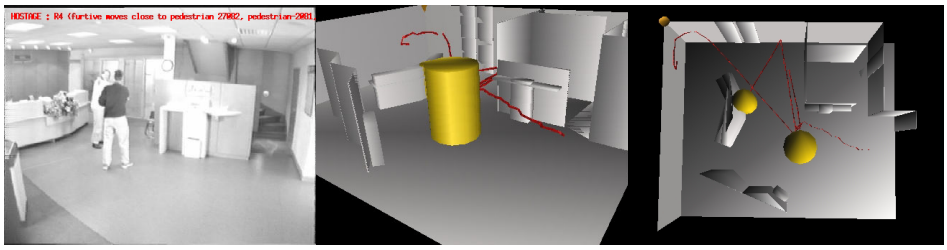


FIG. 8.26 – $\Delta_{HOSTAGE}$ *frames plus tard* ($\Delta_{HOSTAGE}$=*20), le comportement*HOSTAGE *est reconnu. N.B. l'ombre de* h_1 *est maintenant sur le guichet. Ceci créer un suivi très bruité, comme on peut le voir sur la vue de dessus de la reconstruction.*

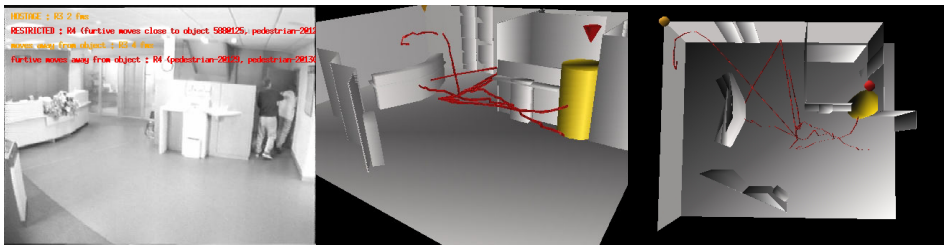


FIG. 8.27 – h_1 *et* h_2 *entrent dans une zone réservée au personnel de l'agence (le couloir à droite),* $\Delta_{RESTRICTED}$ *frames plus tard* ($\Delta_{RESTRICTED}$=*5), le comportement* RESTRICTED *est reconnu.*

8.3.3 Résultats quantitatif de la reconnaissance de comportements pour la sécurité en agence bancaire

Les tableaux 8.8, 8.9 et 8.10 récapitulent les résultats de la reconnaissance de comportements pour la sécurité en agences bancaires. Ces résultats peuvent être interprétés selon deux axes. Le premier point concerne les séquences MC1-22, MC2-17 et MC2-18 qui illustrent des comportements de contrainte sur personne et des accès à des zones réservées. On peut s'apercevoir sur ces vidéos que les comportements attendus sont reconnus aussi bien sur des données segmentées à la main que sur des données réelles. De plus, à l'exception d'une occurrence de *HOSTAGE* dans MC1-22 sur données réelles, aucune fausse reconnaissance n'est observée. En ce qui concerne la vidéo MC2-24 qui illustre un comportement de *POOLING*, les résultats sont moins éloquents. En effet, on peut observer que le comportement en question est reconnu sur des données segmentées à la main, mais qu'il ne l'est pas sur des données réelles (l'occurrence de ce comportement sur MC2-24 en données réelles n'est pas celle attendue. C'est une fausse reconnaissance). Ceci s'explique par le fait que cette vidéo comprend beaucoup de personnes en peu de temps et peu d'espace, ce qui rend le suivi de personnes très compliqué. Malheureusement, le fait qu'il y ait beaucoup de personnes dans la scène est lié au comportement lui-même car par définition, si il y à *POOLING*, c'est qu'il a y un certain nombre de personnes dans l'agence.

	MC1-22	MC2-17	MC2-18	MC2-24
HOSTAGE	[90,150]	[149,197]	[99,127]	[80,140], [289,350], [340,345]
RESTRICTED	[124,150]		[30,43]	
POOLING				[255,355]

TAB. 8.8 – *La vérité terrain de la reconnaissance de comportements pour la sécurité dans les agences bancaires. Chaque occurrence d'un comportement particulier est représentée par l'intervalle de temps correspondant à sa durée.*

8.3.4 Conclusion

Nous avons présenté dans cette section des résultats de notre approche dans le cadre de la sécurité en agence bancaire. Le but poursuivi ici est la reconnaissance de trois types de comportements: les accès aux secteurs réservés au personnel, les comportements de

	MC1-22	MC2-17	MC2-18	MC2-24
HOSTAGE	102	159	109	287, 310, 340
RESTRICTED	129		30	
POOLING				306

TAB. 8.9 – *Résultats de la reconnaissance de comportements pour la sécurité dans les agences bancaire obtenu à partir de données segmentées à la main. Chaque comportement particulier est représenté par l'instant où il est reconnu.*

	MC1-22	MC2-17	MC2-18	MC2-24
HOSTAGE	102, 132	158	99	96, 134, 307
RESTRICTED	130		28	119
POOLING				266

TAB. 8.10 – *Résultats de la reconnaissance de comportements pour la sécurité dans les agences bancaires obtenu à partir de données réelles. Chaque comportement particulier est représenté par l'instant où il est reconnu.*

contrainte sur personnes et les regroupements contraints de personnes.

La particularité de la problématique est ici l'existence de la notion de rôles: les banquiers, les clients, dont les droits sont différents.

La qualité des résultats présentés diffère en fonction du comportement considéré. Les reconnaissances de comportements d'accès aux secteurs réservés et de contrainte sur personnes semblent suffisamment robustes sur des données réelles, alors que les comportements de regroupements restent encore problématiques sur données réelles.

8.4 Résultats de le représentation et reconnaissance de comportements pour le travail médiatisé

8.4.1 Représentation de comportements pour le travail médiatisé

Cette approche a été utilisée dans le cadre d'un projet d'application d'aide au travail collaboratif, appelé le MEDIASPACE [42, 91], dont les principes généraux ont été posés avec la collaboration des membres des équipes PRIMA et IIHM de l'IMAG.

Cette application a pour but de fournir à un ensemble d'utilisateurs un espace de travail virtuel dans lequel chacun de leur poste de travail réel est représenté par un flux vidéo. A chaque poste de travail est associé un état correspondant au niveau d'occupation du poste. Cette état est appelé le "flag". Le flag est vert lorsque le poste est inoccupé, orange si l'utilisateur est présent et rouge si le poste de travail est occupé par plus d'une personne. Dans ce contexte, la reconnaissance de comportements a pour but de fournir à chaque instant, l'état des postes ainsi que les changements d'états éventuels, afin de modifier automatiquement le mode de transmissions d'images vers les autres utilisateurs.

Les figures 8.28, 8.29 et 8.30 montrent les modèles des états du flag: *GREEN FLAG*, *ORANGE FLAG* et *RED FLAG*. Les figures 8.31, 8.32 montrent les modèles des changements du flag *CHANGE GREEN TO ORANGE*, *CHANGE TO GREEN*, *CHANGE TO RED* et *CHANGE RED TO ORANGE*.

$$(x_0 : -)$$

$$\left\{ \; c_1) \quad category(x_0) \quad = \quad pedestrian \right.$$

FIG. 8.28 – *Le modèle de* GREEN FLAG *est composé d'une condition portant sur une variable typée* -. *Ce modèle signifie que* GREEN FLAG *est reconnu à un instant donné, s'il n'existe aucune personne.*

$$(x_0 : +, x_1 : -)$$

$$\begin{cases} c_1) & category(x_0) & = & pedestrian \\ c_2) & category(x_1) & = & pedestrian \\ c_3) & name(x_0) & \neq & name(x_1) \\ c_4) & time(x_1) & = & time(x_0) \end{cases}$$

FIG. 8.29 – *Le modèle de* ORANGE FLAG *est composé de 4 conditions portant sur 2 variables. Ce modèle est reconnu s'il existe un* pedestrian *(cf.* c_1*) et n'existe aucun autre* pedestrian *(cf.* c_2,c_3*) au même instant (cf.* c_4*). En d'autres termes,* ORANGE FLAG *est reconnu s'il existe un unique* pedestrian *à un instant donné.*

$$(x_0 : +, x_1 : +)$$

$$\begin{cases} c_1) & category(x_0) & = & pedestrian \\ c_2) & category(x_1) & = & pedestrian \\ c_3) & name(x_0) & \neq & name(x_1) \\ c_4) & time(x_1) & = & time(x_0) \end{cases}$$

FIG. 8.30 – *Le modèle de* RED FLAG *est composé de 4 conditions portant sur 2 variables. Ce modèle est reconnu s'il existe un* pedestrian *(cf.* c_1*) et si il existe un autre* pedestrian *(cf.* c_2,c_3*) au même instant (cf.* c_4*). En d'autres termes,* RED FLAG *est reconnu si il existe au moins deux* pedestrian *à un instant donné.*

$$(x_0 : +, x_1 : +)$$

$$\begin{cases} c_1) & category(x_0) & = & GREEN\ FLAG \\ c_2) & category(x_1) & = & ORANGE\ FLAG \\ c_3) & time(x_1) & - & time(x_0) = 1 \end{cases}$$

FIG. 8.31 – *Le modèle* CHANGE GREEN TO ORANGE *est reconnu lorsque, entre deux instants consécutifs, un* GREEN FLAG *est suivi d'un* ORANGE FLAG.

$$(x_0 : +, x_1 : +)$$

$$\begin{cases} c_1) & category(x_0) & = & ORANGE\ FLAG \\ c_2) & category(x_1) & = & GREEN\ FLAG \\ c_3) & time(x_1) & - & time(x_0) = 1 \end{cases}$$

FIG. 8.32 – *Le modèle* CHANGE TO GREEN *est reconnu lorsque, entre deux instants consécutifs, un* ORANGE FLAG *est suivi d'un* GREEN FLAG

$$(x_0 : +, x_1 : +)$$

$$\begin{cases} c_1) & category(x_0) & = & ORANGE\ FLAG \\ c_2) & category(x_1) & = & RED\ FLAG \\ c_3) & time(x_1) & - & time(x_0) = 1 \end{cases}$$

FIG. 8.33 – *Le modèle* CHANGE TO RED *est reconnu lorsque, entre deux instants consécutifs, un* ORANGE FLAG *est suivi d'un* RED FLAG.

$$(x_0 : +, x_1 : +)$$

$$\begin{cases} c_1) & category(x_0) & = & RED\ FLAG \\ c_2) & category(x_1) & = & ORANGE\ FLAG \\ c_3) & time(x_1) & - & time(x_0) = 1 \end{cases}$$

FIG. 8.34 – *Le modèle* CHANGE RED TO ORANGE *est reconnu lorsque, entre deux instants consécutifs, un* RED FLAG *est suivi d'un* ORANGE FLAG.

8.4.2 Résultats de la reconnaissance de comportements pour le travail médiatisé

Les figures 8.35, 8.36, 8.37, 8.38 et 8.39 sont prises de la vidéo B008 filmée dans notre propre bureau. Les figures 8.40, 8.41, 8.42, 8.43 et 8.44 sont prises de la vidéo C02-2. Ces deux vidéos illustrent la reconnaissance de comportements pour le travail médiatisé

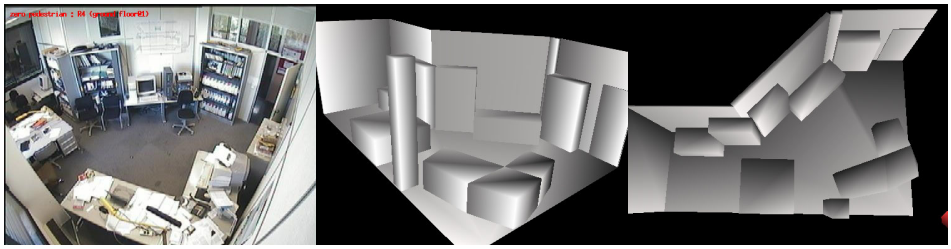


FIG. 8.35 – *Le bureau est vide. Le flag est* GREEN.

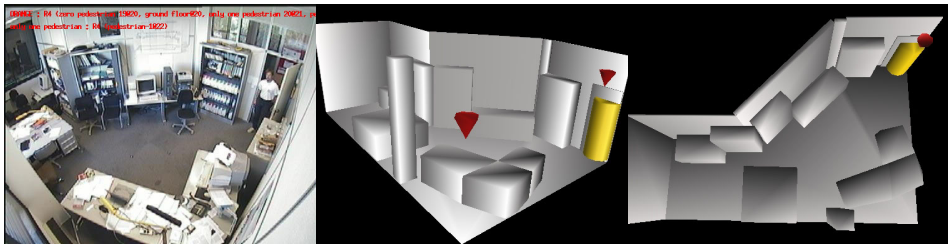


FIG. 8.36 – *Un humain* h_1 *entre. Le flag change à* ORANGE*, le comportement* GREEN TO ORANGE *est reconnu.*

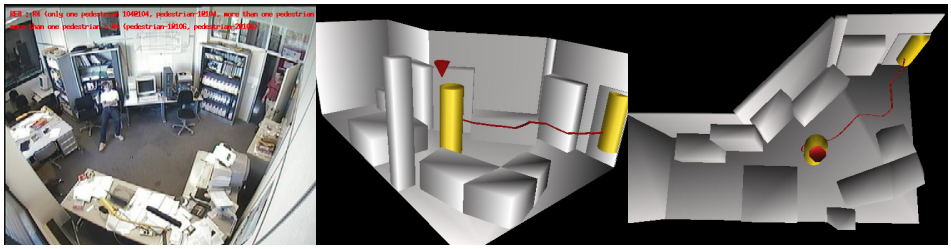


FIG. 8.37 – h_1 *est assis et un nouvel humain* h_2 *entre dans le bureau . Le flag est maintenant* RED *et le changement* TO RED *est reconnu.*

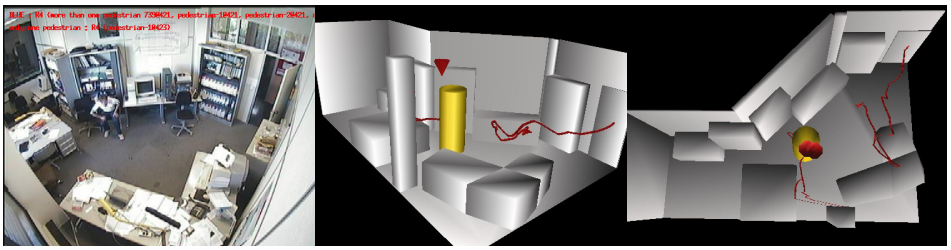


FIG. 8.38 – h_2 *sort du bureau, le flag revient à* ORANGE. *Le comportement* RED TO ORANGE *est reconnu.*

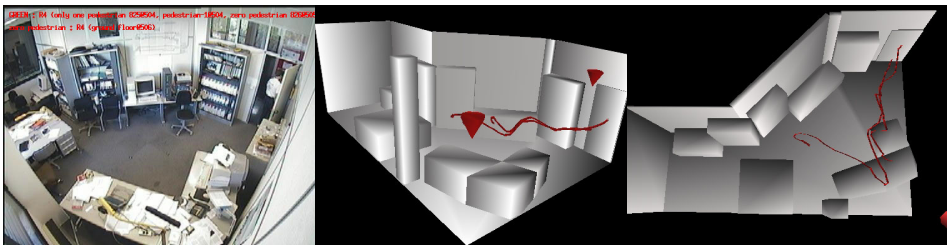


FIG. 8.39 – h_1 *sort du bureau, le flag repasse à* GREEN *et le comportement* TO GREEN *est reconnu*

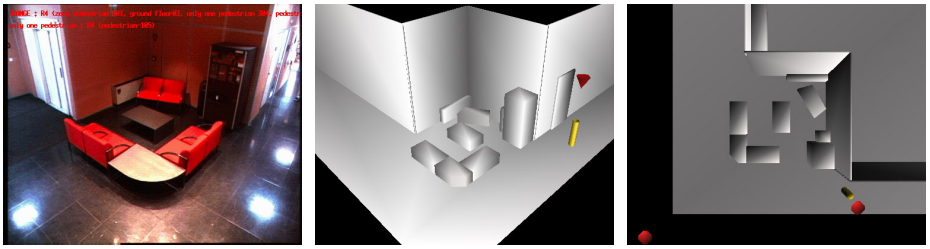


FIG. 8.40 – *La pièce est vide. Le flag est* GREEN.

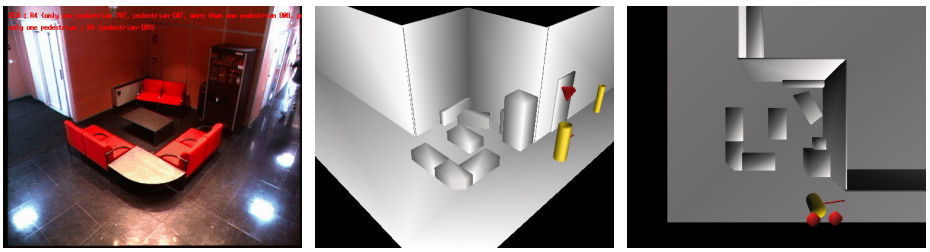


FIG. 8.41 – *Un humain* h_1 *entre, le flag change à* ORANGE, *le comportement* GREEN TO ORANGE *est reconnu.*

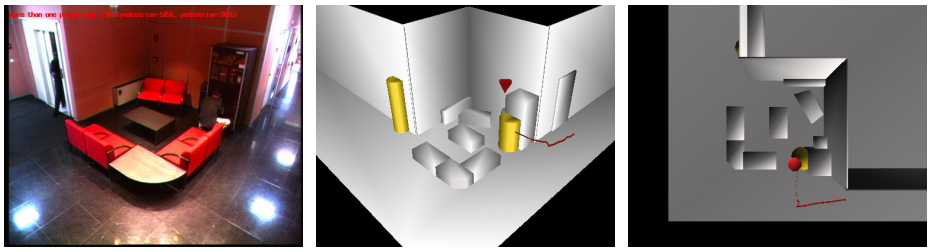


FIG. 8.42 – h_1 *est devant la machine à café et un nouvel humain* h_2 *entre dans la pièce par l'ascenseur (à droite). Le flag passe à* RED *et le changement* TO RED *est reconnu.*

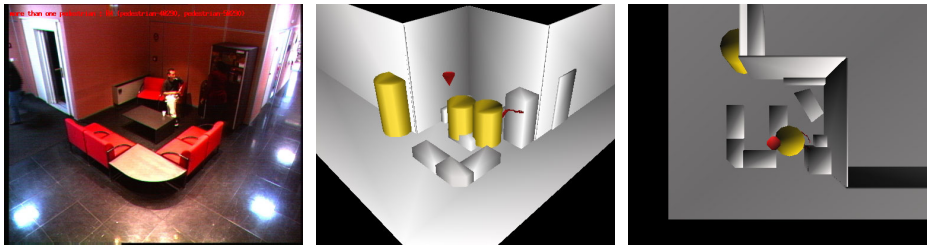


FIG. 8.43 – *Un nouvel humain* h_3 *entre dans la scène par l'ascenseur, mais la scène est déjà dans l'état* RED FLAG*, alors aucun changement n'intervient.*

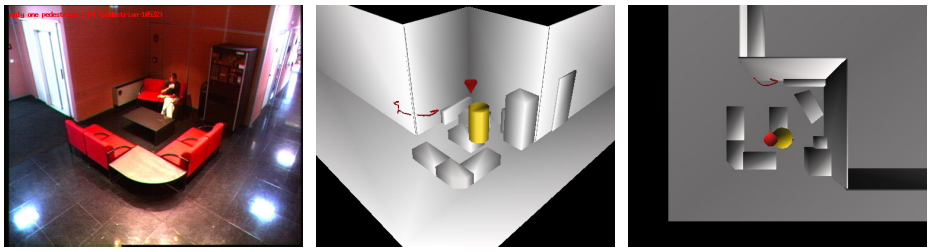


FIG. 8.44 – h_2 *sort du bureau. Le flag repasse à* ORANGE *et le comportement* RED TO ORANGE *est reconnu.*

8.4.3 Résultats quantitatifs de la reconnaissance de comportements pour la travail médiatisé

Les tableaux 8.11, 8.12 et 8.13 récapitulent les résultats obtenus pour la reconnaissance de comportements pour le travail médiatisé. Les résultats obtenus à partir de données segmentées à la main sont identiques à la vérité terrain. Les résultats obtenus à partir de données réelles sont entachées par trois erreurs, correspondant à chaque fois au même problème. Un bruit reconnu comme étant un personne à une frame donnée, provoque la reconnaissance à cette frame d'un flag *ORANGE* au lieu de *GREEN*. Ceci a pour effet de les reconnaissances successives d'un comportement *CHANGE TO RED* et d'un comportement *CHANGE ORANGE TO RED*, comme c'est le cas sur C02-2 (voir figure 8.41) ou sur C07-2 aux frames 16 et 105. Bien que la reconnaissance de l'état du flag soit difficile à changer, ces erreurs auraient pu être facilement gérées en introduisant des comportements de changement du flag *persistent*, comme pour les comportements de base présentés dans la section 8.1.

	B008	CO2-2	CO7-2
CHANGE GREEN TO ORANGE	20	10	13
CHANGE TO RED	105	51	52
CHANGE TO GREEN	509		100
CHANGE RED TO ORANGE	424	532	82

TAB. 8.11 – *La vérité terrain de la reconnaissance de comportements pour le travail médiatisé. Chaque occurrence d'un comportement particulier est représenté par l'instant du changement.*

	B008	CO2-2	CO7-2
CHANGE GREEN TO ORANGE	20	10	13
CHANGE TO RED	105	51	52
CHANGE TO GREEN	509		100
CHANGE RED TO ORANGE	424	532	82

TAB. 8.12 – *Résultats de la reconnaissance de comportements pour le travail médiatisé obtenus à partir de données segmentées à la main. Chaque comportement particulier est représenté par l'instant où il est reconnu.*

	B008	CO2-2	CO7-2
CHANGE GREEN TO ORANGE	22	5	13
CHANGE TO RED	106	9, 51, 534	16, 52, 100
CHANGE TO GREEN	506		106
CHANGE RED TO ORANGE	423	10, 532	17, 82, 105

TAB. 8.13 – *Résultats de la reconnaissance de comportements pour le travail médiatisé obtenus à partir de données réelles. Chaque comportement particulier est représenté par l'instant où il est reconnu.*

8.4.4 Conclusion

Nous avons présenté dans cette section, des résultats de notre approche dans le cadre de l'aide au travail médiatisé. Le but poursuivi ici est la reconnaissance des états et des changements de niveau de disponibilité des postes de travail.

La particularité de cet exemple est double. D'un premier point de vue, on peut remarquer qu'il ne s'agit pas de reconnaître des comportements humains à proprement parler, mais plutôt de comportements de scène globale. D'un second point de vue, il est à noter que le volume d'instance de comportements à reconnaître ici est beaucoup plus grand que dans les applications "métro" ou "banque". En effet, alors que le nombre moyen de comportement à reconnaître dans les applications "métro" ou "banque" est de 1 pour 400 ou 500 frames, le nombre de comportements à reconnaître dans ce cadre, est au moins 1 par frame.

La qualité des résultats présentés semble suffisamment robuste sur des données réelles. En outre, la particularité évoquée précédemment n'a pas compliqué ou ralenti l'algorithme de reconnaissance.

8.5 Résultats de la représentation et de la reconnaissance de comportements d'interactions dans les Parkings

8.5.1 Représentation de comportements d'interactions dans les Parkings

Nous présentons dans cette section, les résultats d'une étude menée conjointement avec l'équipe du Professeur L. Davis de l'université de Maryland, spécialisée en suivi de personnes. Dans le cadre de cette étude, la partie suivi de personnes était gérée par C. Ben Abdel Cader de l'équipe de L. Davis.

Cette étude vise à reconnaître certains comportements liés aux interactions entre plusieurs personnes via certains objets. L'objectif est ici de reconnaître le changement d'appartenance d'un objet d'une personne à une autre, afin de pouvoir reconnaître le comportement d'une personne s'appropriant une valise ou un attaché-case laissée par inadvertance par une autre personne.

L'objectif étant de différencier ce comportement de celui d'une personne qui récupère sa valise ou son attaché-case qu'il avait laissé quelques instants auparavant.

Afin de différencier ces deux cas, nous avons construit un ensemble de modèles dont les deux modèles de plus haut niveau sont *Ownership-Change*, montré sur la figure 8.47, représentant le changement de propriété et *Lost-and-Found*, montré sur la figure 8.48, représentant la conservation de propriété.

Dans les deux cas, on cherche à reconnaître le couple d'actions: "une personne dépose un objet" et "une personne ramasse cet objet". Dans le premier cas, les personnes impliquées dans les deux actions sont différentes et dans le second cas, les personnes impliquées dans les deux actions sont le même humain réel.

L'action "une personne dépose un objet" est représentée par le modèle de *Object-Deposite*, montré sur la figure 8.45, et l'action "une personne ramasse cet objet" est représentée par le modèle de *Object-PickUp*, montrée sur la figure 8.46.

Notons que ces deux modèles impliquent l'existence de deux autres modèles *Object-Appears* et *Object-Disappears*, qui sont très similaire à *Pedestrian-Appears* et *Pedestrian-Disappears* définis dans la librairie de comportements de base, détaillée dans la section 8.1.

$$(x_0 : +, x_1 : +, x_2 : -)$$

$$\begin{cases} c_1) & category(x_0) & = & pedestrian \\ c_2) & name(x_1) & = & Object\text{-}Appears \\ c_3) & time(x_1) & = & time(x_0) \\ c_4) & distance(hull(x_0), hull(x_1)) & < & 300 \\ c_5) & name(x_2) & = & Object\text{-}Disappears \\ c_6) & name(ref(x_1,1)) & = & name(ref(x_2,1)) \\ c_7) & time(x_1) - time(x_2) & > & 0 \end{cases}$$

FIG. 8.45 – *Modèle de* Object-Deposite

$$(x_0 : +, x_1 : +, x_2 : -, x_3 : -)$$

$$\begin{cases} c_1) & category(x_0) & = & pedestrian \\ c_2) & name(x_1) & = & Object\text{-}Disappears \\ c_3) & time(x_1) & = & time(x_0) \\ c_4) & distance(hull(x_0), hull(x_1)) & < & 400 \\ c_5) & name(x_2) & = & Object\text{-}Appears \\ c_6) & name(ref(x_1,1)) & = & name(ref(x_2,1)) \\ c_7) & time(x_1) - time(x_2) & > & 0 \\ c_8) & name(x_3) & = & Object\text{-}PickUp \\ c_9) & name(ref(x_1,1)) & = & name(ref(x_3,1)) \\ c_{10}) & time(x_1) - time(x_3) & > & 0 \end{cases}$$

FIG. 8.46 – *Modèle de* Object-PickUp

$$(x_0 : +, x_1 : +)$$

$$\begin{cases} c_1) & name(x_0) & = & Object\text{-}Deposits \\ c_2) & name(x_1) & = & Object\text{-}PickUp \\ c_3) & name(ref(x_0,1)) \neq & name(ref(x_1,1)) & \\ c_4) & name(ref(x_0,2)) & = & name(ref(x_1,2)) \end{cases}$$

FIG. 8.47 – *Modèle de* Ownership-Change

$$(x_0 : +, x_1 : +)$$

$$\begin{cases} c_1) & name(x_0) & = & \textit{Object-Deposits} \\ c_2) & name(x_1) & = & \textit{Object-PickUp} \\ c_3) & time(x_0) - time(x_1) & > & 0 \\ c_4) & name(ref(x_0,1)) & = & name(ref(x_1,1)) \\ c_5) & name(ref(x_0,2)) & = & name(ref(x_1,2)) \end{cases}$$

FIG. 8.48 – *Modèle de* Lost-and-Found

8.5.2 Reconnaissance de comportements d'interaction dans les parkings

Les figures 8.49, 8.50, 8.51, 8.52 et 8.53 sont une partie de la vidéo SEQ5 filmée sur le parvis de l'université du Maryland et illustrent l'identification du comportement de changement de propriété d'un attaché-case. Le sémantique des figures est identique à la précédente excepté que tous les éléments de l'environnement sont en gris.

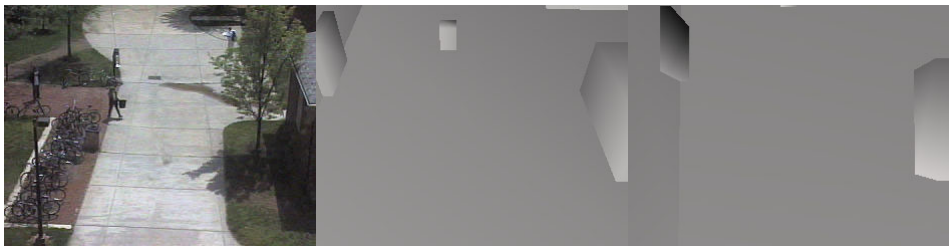


FIG. 8.49 – *Un humain h_1 traverse le parvis un attaché-case à la main. Un second humain h_2 arrive sur la gauche.*

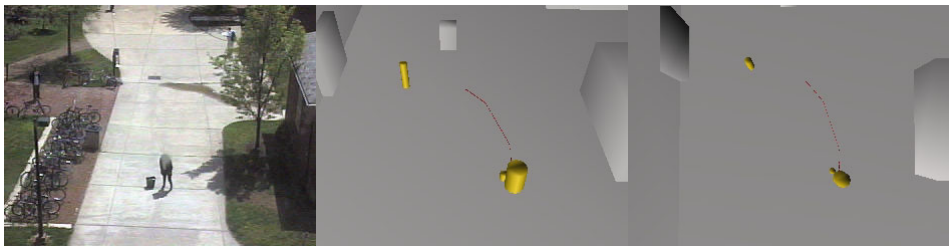


FIG. 8.50 – *h_1 dépose son attaché-case au milieu du parvis. Un premier comportement* Object-Deposits *est reconnu.*

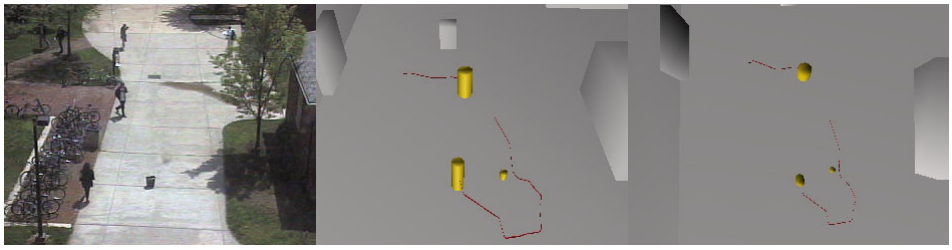


FIG. 8.51 – *h_1 s'éloigne de l'attaché-case et h_2 s'approche.*

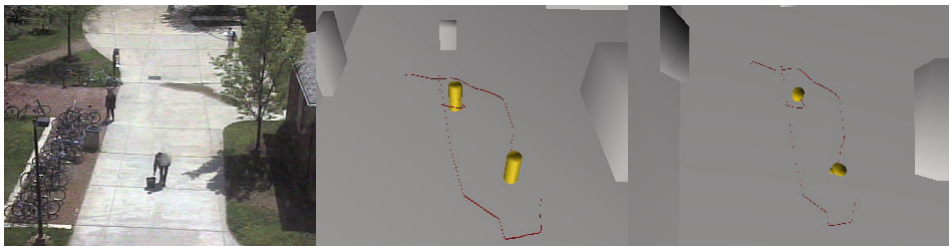


FIG. 8.52 – h_2 *ramasse l'attaché-case Un comportement* Object-PickUp *est reconnu; suivi d'un* Ownership-Change

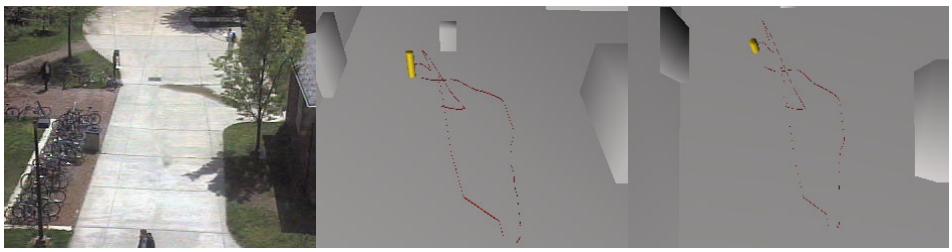


FIG. 8.53 – h_2 *sort de la scène (en bas de l'image) avec l'attaché-case.*

8.5.3 Résultats qualitatifs de la reconnaissance de comportements d'interaction dans les parkings

Les tableaux 8.14 et 8.15 présentent les résultats obtenus dans ce cadre. On peut voire que les résultats obtenus avec des données réelles sont très similaires à ceux de la vérité terrain. Les deux occurrences de *OWNERSHIP CHANGE* et *LOST AND FOUND* sont reconnues et il n'y a pas de fausse reconnaissance. On peut observer un retard de 15 frames sur les instances de *OBJECT DEPOSIT* et *OBJECT PICKUP* causé par un délai imposé par l'algorithme de suivi de personnes utilisé. On peut aussi observer un retard de 16 frames pour les instances de *OWNERSHIP CHANGE* et *LOST AND FOUND* causé par le retard de reconnaissance de *OBJECT PICKUP*; plus une frame causée par l'algorithme de reconnaissance de comportements.

En revanche, ce qui n'est pas visible dans ces résultats est le grand nombre de fausses occurrences de *OBJECT DEPOSIT* obtenues avec des séquences vidéos non-mentionnées ici. On explique ceci par la classification erronée de certains humains en temps qu'objet.

	SEQ5	SEQ6
OBJECT DEPOSIT	168	791
OBJECT PICKUP	311	981
OWNERSHIP CHANGE	311	
LOST AND FOUND		981

TAB. 8.14 – *La vérité terrain de la reconnaissance de comportements d'intéraction dans les parkings. Chaque occurrence d'un comportement particulier est représentée par l'instant correspondant à son occurrence.*

	SEQ5	SEQ6
OBJECT DEPOSIT	183	806
OBJECT PICKUP	326	996
OWNERSHIP CHANGE	327	
LOST AND FOUND		997

TAB. 8.15 – *Le résultat de la reconnaissance de comportements d'intéraction dans les parkings. Chaque occurrence d'un comportement particulier est représentée par l'instant correspondant à son occurrence.*

8.5.4 Conclusion

Nous avons présenté dans cette section, les résultats d'une étude menée en collaboration avec l'équipe de L. Davis de l'université de Maryland, dont l'objectif était la reconnaissance d'interactions spécifiques entre humains et objets.

Un point intéressant de cette étude est en premier lieu, la nature très différente des vidéos analysées. Comparée aux applications "métro", "banque" ou "bureau", on se situe ici, en extérieur avec des contraintes d'illuminations qui rendent le suivi de personnes bien plus complexe.

Un second point intéressant de cette étude est la nature différente des objets non-humains. En effet, nous avons jusqu'alors des applications pour lesquelles tous les objets mobiles issus du suivi étaient des personnes. Ici, la notion d'objet (non-humain) prend un caractère dynamique. Les objets du décor ne sont pas tous connus à l'instant initial.

8.6 Performance de l'approche en terme de temps de calcul

Cette section présente une analyse des performances en terme de temps de calcul de notre approche. Bien que le problème du temps réel ne soit pas l'une de nos contraintes, il n'en reste pas moins que ce point continue à être un critère pour beaucoup.

Ces tests ont été réalisés sur une machine SUN ULTRA 10, 333 MHz. Les résultats de ces tests sont organisés par famille d'applications: "métro", "banque" et "bureau".

Chacune des figures présente les temps de calcul observés et présentés en fonction des différents algorithmes. Les courbes rouges représentent le nombre de secondes nécessaires au calcul de B_t (segmentation des blobs présenté dans le chapitre 4). Les courbes vertes représentent le nombres de secondes nécessaires au calcul de Q_t (groupement des blobs présenté dans le chapitre 4). Les courbes bleues présentent les temps de calcul de (P_t,T_t) (mise en correspondance temporelle présentée dans le chapitre 5). Les courbes violettes représentent les temps de calcul d'un boucle complète, c'est à dire jusqu'à la reconnaissance de comportements (voir 7).

Les figures 8.54 et 8.55 représentent les performances dans le cadre "métro". Les images du flux sont de taille 512 par 512 et sont relativement bruitées se qui induit un grand nombre de blobs. Les scènes sont globalement peu occupées ($|\bar{P}_t|$ est petit) et le modèle de scène relativement simple ($|\bar{O}_t|$ est petit). La base de modèle de comportements est celle présentée dans la section 8.2.

La durée moyenne d'un cycle de l'algorithme est 600 millisecondes.

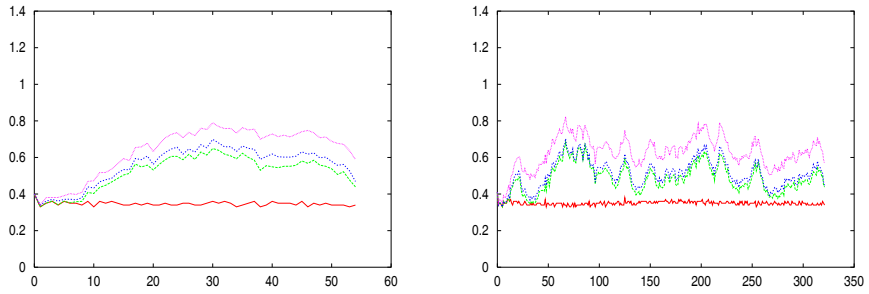


FIG. 8.54 – *Performance en terme de temps de calcul du cadre "métro" sur les vidéos VA2-7 et VA2-6*

Les figures 8.56 et 8.57 représentent les performances dans le cadre "banque". Les images du flux sont de taille 440 par 334 et sont relativement peu bruitées. Les scènes sont relativement (voir très) occupées ($|\bar{P}_t|$ est grand) et le modèle de scène assez complexe ($|\bar{O}_t|$ est grand). La base de modèle de comportements est celle présentée dans la section 8.3.

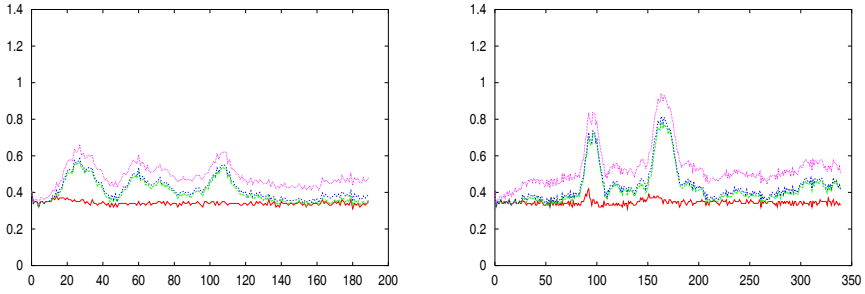


FIG. 8.55 – *Performance en terme de temps de calcul du cadre "métro" sur les vidéos ST1-23 et VA2-4*

La durée moyenne d'un cycle de l'algorithme est 900 millisecondes.

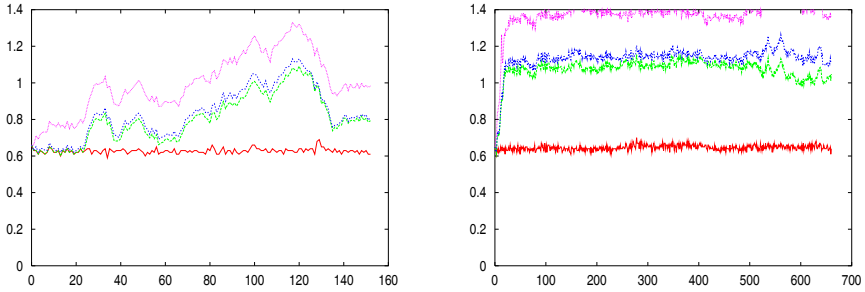


FIG. 8.56 – *Performance en terme de temps de calcul du cadre "banque" sur les vidéos MC1-22 et MC1-30*

La figure 8.58 représente les performances dans le cadre "bureau". Les images du flux sont de taille 504 par 406 et sont relativement peu bruitées, mais de mauvaise qualité. Les scènes sont relativement peu occupées ($|\bar{P}_t|$ est petit) et le modèle de scène assez complexe ($|\bar{O}_t|$ est grand). La base de modèle de comportements est celle présentée dans la section 8.4.

La durée moyenne d'un cycle de l'algorithme est 500 millisecondes.

8.6.1 Conclusion

Nous avons présenté dans cette section une analyse des performances en terme de temps de calcul de notre approche.

Les conclusions à tirer de cette analyse sont de plusieurs ordres. Premièrement, d'un point de vue global, on peut observer que la durée moyenne d'un cycle complet (de l'image

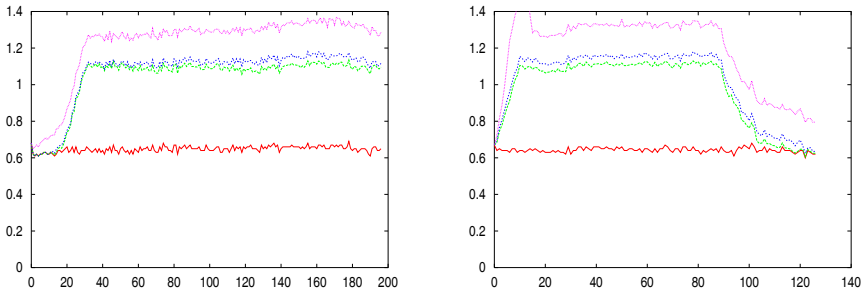


FIG. 8.57 – *Performance en terme de temps de calcul du cadre "banque" sur les vidéos MC2-17 et MC2-18*

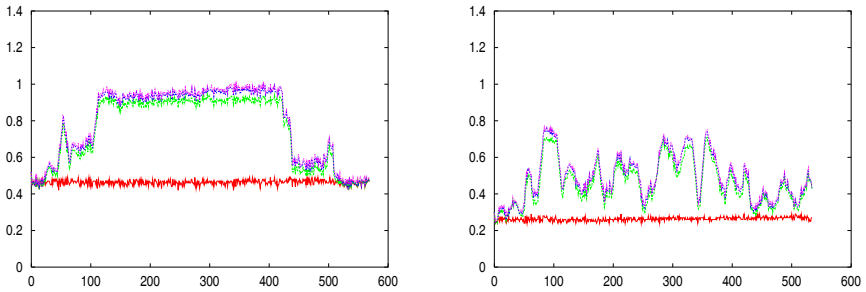


FIG. 8.58 – *Performance en termes de temps de calcul du cadre "bureau" sur les vidéos B008 et C02-2*

aux comportements) varie entre 500 millisecondes et 1400 millisecondes, ce qui est bien au delà des contraintes temps réel (40 millisecondes à la cadence vidéo). Malgrès tout, ces performances ne sont qu'à un facteur deux ou trois des temps de calculs admissibles pour un flux d'images à 5 images par seconde.

D'un point de vue plus spécifique, on peut remarquer que la majeur partie du temps de calcul est octroyé à la segmentation (les courbes rouges); partie sur laquelle beaucoup de possibilités sont envisageables pour réduire ce temps de calcul.

On peut remarquer que la solution proposée pour la reconnaissance de personnes est caractérisée par des temps de calcul totalement instables. En effet, la complexité algorithmique de la méthode dépendant de la longueur de parcours heuristique, ainsi que du nombre de blobs issus de la segmentation, paraît difficilement contrôlable. Une solution envisagée pour stabiliser cette partie est d'opérer une présélection des blobs avant de partitionner, afin d'assurer un nombre maximal de blobs.

La part de temps de calcul demandée par la solution proposée au problème de mise en correspondance temporelle est, d'une part relativement infime, et d'autre part relativement

stable.

Quant aux temps de calculs investis sur la reconnaissance de comportements proprement dite, la nature hétérogène de ceux-ci est explicable et attendue. En effet, la complexité algorithmique de la méthode dépend, d'une part du nombre et de la complexité des modèles de la base, et d'autre part de la taille du graphe partiel d'interprétation. Ainsi, pour des applications telles que les applications bureaux, les modèles étant simples et peu nombreux, la part de temps de calcul est quasiment négligeable.

A l'opposé, pour des applications telles que les applications "banque", les modèles étant complexes et nombreux, la taille du graphe partiel d'interprétation importante, la part de temps de calcul est quasiment élevée.

8.7 Conclusion

Nous avons présenté dans ce chapitre, cinq utilisations de notre approche pour la reconnaissance de comportements.

La première application, présentée dans la section 8.1, consiste en un ensemble de comportements de base. La seconde application, présentée dans la section 8.2, a pour objectif la sécurité dans les stations de métro. La troisième application, présentée dans la section 8.3, est focalisée sur la sécurité en agences bancaires. Le quatrième exemple d'utilisation, dans la section 8.4, est relatif à une application d'aide au travail médiatisé. Le dernier exemple présenté dans la section 8.5, est une étude sur la reconnaissance de comportements basése sur les interactions humains/objets dans les parkings.

La première conclusion à tirer de ces différentes utilisations de notre approche, correspond à l'objectif de généricité recherché avec cette approche. En effet, nous avons vu dans ce chapitre, qu'il nous a été possible d'appliquer cette méthode à des applications aux caractéristiques très diverses. Mais de plus, il nous a été possible d'intégrer simplement les spécificités de chaque cas. Le calcul d'états dans l'application "bureau" a été rendu possible sans aménagement particulier. La caractère particulier des objets de décor non connu à l'instant initial, dans l'application "parking", a été intégré. Enfin la notion de rôles des personnes dans l'expertise "banque", n'a pas demandé de transformation lourde du formalisme.

La seconde conclusion à tirer, de façon transversale à ces différents résultats, concerne la robustesse de la reconnaissance. A ce titre, on a pu s'apercevoir que la robustesse de la reconnaissance dépend de la nature des comportements à reconnaître. Pour simplifier, on pourrait dire que plus un comportement fait intervenir de personnes, de façon directe

on indirecte, plus sa reconnaissance devient fragile. On pourrait prendre comme exemple les comportements de base d'interactions entre deux personnes ou le comportement de regroupement de personnes de l'application "banque", dont la reconnaissance reste fragile. *A contrario*, les comportements d'accès ("banque" ou "métro") ou les comportements de station proche d'un élément du décor profitent, quant à eux, d'une reconnaissance robuste.

En ce qui concerne les limitations de l'approche, on pourrait dire que la limitation principale est lié à la nature même de l'approche. En effet, cette approche étant basée sur l'instanciation de modèle similaire à des systèmes de contraintes, c'est à dire que les comportement sont définis dans le domaine de la logique et donc toute forme de concept du domaine arithmétique est difficile à modéliser. En particulier, tout les comportements basés sur la notion de comptage sont exclus. Par exemple, dans la cadre "bureau" ou "parking", on aurait souhaité pouvoir modéliser des comportements lié à un comptage des personnes de la scène.

Chapitre 9 Conclusion

Nous avons présenté dans ce manuscrit, VSIS, le résultat de nos recherches menées sur la reconnaissance de comportements. Le but de ces recherches est l'élaboration d'un programme capable de reconnaître certains comportements humains. Le principe général consiste à se doter, outre d'un flux vidéo, d'un ensemble de descriptions des comportements que l'on souhaite reconnaître et d'une description du décor dans lequel est filmée la scène. Ainsi, on extrait du flux vidéo à temps fixe, les indices du mouvement des personnes, afin de calculer une description de ceux-ci. C'est à dire estimer leur taille ou leur volume et les replacer dans le décor. Avec cette description complète de la scène (personnes + décor) à un instant donné, on calcule la façon dont la scène a évolué afin d'obtenir une description spatiale et temporelle de la scène. A partir de cette description spatiale et temporelle les différents comportements prédéfinis pourront être reconnus.

Ainsi, la première partie de ce document dont le chapitre 3 fait l'objet, a pour but de définir un cadre formel à ces travaux. Nous avons vu que ce problème pouvait être modélisé comme le calcul incrémental d'un graphe (appelé graphe d'interprétation) dont les sommets représentent les éléments physiques et abstraits de la scène au cours du temps (les personnes, les éléments du décor et les manifestations des comportements). Le problème de reconnaissance de comportements correspond alors à la mise à jour du graphe d'interprétation à partir d'une image issue du flux vidéo, du modèle de scène et des modèles de comportements. En d'autres termes, ce problème correspond au calcul d'un ensemble de sommets valués et arcs en minimisant les différences entre la représentation donnée et le phénomène physique réel.

Les avantages de ce cadre formel sont multiples: le premier avantage est de considérer le problème de la reconnaissance de comportements comme un problème global, tout en gardant la spécificité des différents sous-problèmes. Le second avantage de cette modélisation est de permettre de donner une définition précise aux concepts importants du problème tels que la notion d'individu, la notion de manifestation d'un comportement, la notion de référence d'un concept à un autre, la connaissance *a priori* sur l'environnement et surtout la notion de processus d'interprétation. Enfin le troisième avantage de cette modélisation est de garder la granularité temporelle apparente dans la structure même du graphe d'interprétation. Deux améliorations de cette modélisation sont envisageables.

La première amélioration est liée au caractère temporel unique des concepts. En effet,

nous avons vu que dans ce cadre un sommet du graphe était défini par un attribut *time* unique. C'est à dire qu'à chaque concept ne peut être associé qu'un unique niveau de temps. On pense alors à la difficulté de gérer des informations arrivant de plusieurs sources différentes qui ne sont pas forcement synchronisées les unes avec les autres. Cette amélioration, bien que l'ajout d'un attribut supplémentaire à tous les sommets du graphe semble assez bénin, soulève tout de même le problème de faire coexister dans une structure fortement temporelle (le graphe d'interprétation) plusieurs niveaux de temps.

La seconde amélioration est liée au caractère instantané des concepts du graphe et tout particulièrement le caractère instantané des comportements reconnus. En effet, comme nous l'avons vu, seul un unique sommet du graphe représente un comportement. Une amélioration intéressante de cette modélisation serait de considérer la manifestation d'un comportement comme une suite de sommets (de la même façon qu'un individu est une suite de sommets *personne*). Cette amélioration, qui n'est pas en contradiction avec le reste de la modélisation, peut être géré facilement soit au niveau des modèles de comportements, soit à un niveau algorithmique comme un suivi de comportements.

Comme il a été dit plus haut, l'un des avantages de cette modélisation est de considérer le problème de la reconnaissance de comportements comme un problème global, tout en gardant la spécificité des différents sous-problèmes. Comme nous l'avons vu, ces sous-problèmes sont au nombre de trois et le chapitre 4 a pour objet le premier d'entre eux: le problème de la reconnaissance de personnes.

La méthode proposée dans le chapitre 4 consiste à résoudre ce problème en trois étapes. La première étape a pour but de calculer une image idéale de la scène vide; c'est à dire une image ayant les mêmes conditions que l'image courante du flux mais dépourvue de la projection des humains. De façon plus précise, la première étape consiste en le calcul d'une image composite à partir d'une image idéale de l'instant précédent et d'une image réelle de l'instant précédent.

La seconde étape a pour objectif d'extraire sous forme structurée les pixels issus de la différence entre l'image idéale de la scène vide et l'image courante. La structure adoptée est une structure de région connexe de points appelée blob. La solution retenue consiste en une séquence d'opérateurs arithmétiques et morphologiques: différence absolue, seuillage, érosion, dilatation, analyse des composants connexes.

Enfin, la dernière étape a pour double but, d'une part d'identifier les blobs relatifs à une quelconque forme de bruit, et d'autre part de regrouper les blobs issus de la projection de même personnes. La méthode de résolution de cette étape consiste en un parcours heuristique de l'ensemble des partitions possibles dont l'heuristique est basée sur un modèle de personne hybride 2D/3D/densité.

L'avantage de cette approche est d'une part sa rapidité. En effet, le problème étant très rapidement extrait du domaine de l'image, le coût en temps de calcul en est très diminué. Le second intérêt de l'approche réside dans l'apport d'information *a priori*. L'information du modèle de scène autorise le raisonnement 3D. L'information du modèle de personne autorise un raisonnement générique. Le troisième avantage de l'approche (en particulier en ce qui concerne le parcours heuristique) réside dans le caractère générique de l'approche. Ce cadre étant suffisamment vaste, nombre de perspectives d'améliorations peuvent être envisagées.

Tout d'abord, comme il a été dit dans ce chapitre, le problème dur de cette étape est la gestion du bruit dans les mesures issues de la segmentation. Ainsi, une première piste consisterait à étendre le modèle de mesures arithmétiques simples à un modèle basé sur une algèbre floue. Certaines expérimentations non relatées dans ce manuscrit ont été menées dans ce sens. L'idée de ces expérimentations fut de considérer l'hypothèse d'un bruit minimal de 1 pixel sur toutes les mesures issues de l'image afin de définir des distributions de possibilité représentant ces mesures. Ainsi, en étendant l'ensemble des applications utilisant ces mesures à des applications floues, il nous a été possible d'obtenir des descriptions floues de personnes (c.à.d des descriptions gérant leur propre incertitude). Deux avantages majeures furent mis en évidence par cette étude.

Premièrement, cette gestion simple de l'incertitude propagée jusqu'à la reconnaissance mettait en évidence que certaines zones de certaines scènes sont impropres à la reconnaissance de certain comportement. Par exemple, prenons le cas d'un comportement de base tel que "s'approche de" dont la structure est basée sur une modification de la position d'une personne de quelques centimètres par rapport à un élément du décor. Une incertitude minimale de 1 pixel sur la position d'une personne dans la scène impliquait une imprécision de plusieurs mètres dans certaines parties de la scène (en l'occurrence, les parties suffisamment loin de la caméra). Ainsi, même si un comportement "s'approche de" était reconnu dans ces conditions, l'incertitude serait suffisamment grande pour rendre cette reconnaissance inutilisable.

Le second gain que pourrait apporter une gestion des incertitudes basée sur une algèbre floue, a trait à la gestion des occultations statiques. En effet, par définition, les zones de la scène susceptibles de causer des occultations sont des zones de l'image où la projection des personnes est incertaine. Néanmoins, cette incertitude peut être connue à l'avance et son influence limitée. Par exemple, prenons le cas d'une table au centre de la scène derrière laquelle les personnes sont susceptibles de pouvoir passer. Une personne détectée derrière cette table pose le problème de l'ambiguïté sur sa position au sol. Une gestion de la position des personnes basée sur des points 3D flous permettrait de borner l'ensemble des positions

possibles.

Malgré tout, l'extension de cette approche à une approche floue a un coût. En premier lieu, la conversion de fonctions classiques en fonctions floues tend évidemment à multiplier le temps de calcul (au moins par 4). Deuxièmement, le problème du passage de descriptions floues vers une description classique reste un problème ouvert.

Une seconde piste d'amélioration de notre approche de reconnaissance de personnes est liée aux hypothèses que nous avons fixées sur le modèle de scène. En effet, il a été dit dans ce chapitre que le modèle de scène était constitué de façon *a priori* et était considéré constant. Ceci apparait rapidement réducteur et source d'erreurs dans des scènes composées d'éléments non-fixes tels que les portes, les ascenseurs, les chaises, etc ... L'extension du modèle de scène constant à un modèle de scène non constant soulève la problématique suivante: si le modèle de scène n'est pas constant, il doit alors être mis à jour. Or si les informations utilisées pour ce faire sont de même nature que les informations utilisées pour reconnaître les personnes, ces deux algorithmes risquent d'être concurrents. En d'autres termes, comment discriminer une porte qui s'ouvre avec une personne devant cette même porte sur la simple donnée d'un ensemble de blobs?

La troisième perspective concernant la méthode de reconnaissance de personne présentée dans le chapitre 4 est liée au parcours heuristique calculant la partition de l'ensemble des blobs. Comme nous l'avons vu, cette méthode offre un cadre générique de résolution de ce problème dont l'inconvénient majeur est la complexité algorithmique. En effet, cette complexité dépendant fortement du nombre de blobs qui est difficile à prévoir, le temps de parcours est difficile à prévoir également. La problématique d'une amélioration de cette méthode est alors de réussir à stabiliser le temps de parcours sans pour autant limiter les performances de la méthode.

Certaines expériences non décrites ici, laissent entrevoir que la convergence de cet algorithme est relativement rapide d'abord, puis se ralentit doucement. C'est à dire qu'une bonne solution est trouvée rapidement puis raffinée pendant longtemps. On imagine alors limiter la durée du parcours sur l'évaluation du gain à chaque itération. De façon globale, beaucoup d'expérimentations peuvent encore être menées dans la cadre de ce parcours heuristique.

Le chapitre 5 a eu pour objet le second problème de la reconnaissance de comportement qu'est la reconstruction incrémentale de l'évolution de la scène (c'est à dire le problème de mise en correspondance temporelle). La méthode proposée dans ce chapitre consiste en le calcul d'un diagnostic optimal de l'évolution du système entre deux frames. A partir d'un ensemble de primitives, appelées fonctions de mise en correspondance, on construit à chaque frame, un ensemble de diagnostics possibles dont on garde le plus vraisemblable. Ce

diagnostic optimal conditionne alors l'évolution de la description du système.

L'avantage de l'approche réside dans la gestion unifiée et simultanée de l'ensemble des caractéristiques du problème (calcul de similarité entre deux individus, problèmes d'entrées/sorties, problème d'occultation dynamique, problème de bruit et de perte de détections).

Les perspectives que l'on envisage sont de deux natures. En premier lieu, l'amélioration du mode d'évaluation des diagnostics semblerait pouvoir augmenter les performances de l'approche. Deux points se distinguent alors quant à l'amélioration de l'évaluation d'un diagnostic: l'amélioration de l'évaluation des prédicats d'entrées/sorties, d'occultation et de bruit d'une part, et l'amélioration du calcul de similarité entre deux individus d'autre part.

On envisage pouvoir passer d'un modèle de prédicat booléen à un modèle probabiliste. Le prédicat *IsInIo* au lieu de valoir vrai ou faux en fonction de la localisation d'une personne pourrait être défini par une probabilité calculée en fonction de la localisation de la personne. De la même façon, l'évaluation du prédicat NOISE que l'on avait défini comme étant toujours possible (c.à.d e(NOISE)=1), pourrait être calculée comme une probabilité d'être du bruit (c.à.d de ne pas être une personne) par comparaison avec le modèle de personne présenté dans le chapitre 4.

La souplesse apportée par un mode d'évaluation probabiliste plutôt que par un mode d'évaluation booléen tendrait à rendre cette approche plus robuste.

L'amélioration du calcul de similarité entre deux personnes demande quant à lui plus d'investigation. En effet, pour améliorer ce calcul il apparait clairement que de l'information supplémentaire doit être ajoutée. Les sources d'informations supplémentaires que l'on peut ajouter sont au nombre de deux: l'information de forme et l'information de couleur. Comme nous l'avons déjà dit, nous ne pensons pas que l'information de forme soit suffisamment robuste dans le cas général pour suffire à apporter de la robustesse au calcul de similarité. En revanche, l'information de couleur semble prometteuse bien que difficile. L'extension du calcul de similarité basé sur l'information localisation/taille 2D/3D à un calcul de similarité basé sur l'information localisation/taille 2D/3D plus couleur pose le problème de l'extension de la description des personnes à une description comprenant l'information de couleur. La problématique de l'incorporation de l'information de couleur dans la description des personnes se pose de la façon suivante: premièrement, pour que cette information soit utilisable il faut que pour une personne réelle donnée l'information de couleur extraite soit relativement constante. Deuxièment pour que cette information soit utilisable il faut que pour deux personnes données, l'information couleur extraite reflète les différences entre ces deux personnes. De plus, pour que cette information ait un sens, il faut aussi qu'elle

puisse être abstraite. Certaines expérimentations non décrites dans ce manuscrit, laissaient entrevoir la difficulté de l'extension de la description des personnes à une description de couleurs. L'idée de ses expérimentations fut d'estimer une répartition des couleurs dans l'espace RGB de personnes prédéfinies par un ensemble de noyaux obtenus par "nuées dynamiques". Les résultats obtenus montraient une stabilité très relative des descriptions pour une personne donnée au cours du temps, ainsi qu'un caractère discriminant très moyen entre deux personnes à un instant donné.

Une autre perspective intéressante de cette approche consisterait à unifier les méthodes de reconnaissance de mise en correspondance temporelle. L'idée consisterait à étendre le parcours heuristique présenté dans le chapitre 4 jusqu'au calcul de diagnostic optimal. En d'autres termes, il s'agit introduire dans l'heuristique κ l'évaluation du diagnostic optimal obtenu grâce à la partition définie par l'état courant du parcours. Le premier gain de cette unification est d'éliminer l'interface entre les deux algorithmes qui est en fait le filtrage des sous-ensembles de la partition trop éloignée du modèle de personne. Le second gain de cette unification serait de donner plus de marge de manoeuvre à la méthode de mise en correspondance temporelle dans la mesure où plusieurs configurations de correspondance serait testées.

Nous avons tenté à travers les méthodes proposées dans les chapitres 4 et 5 de montrer que le problème du suivi de personnes pouvait trouver une solution générique (c'est à dire une solution qui n'a pas besoin d'être modifiée pour chaque environnement) et robuste (c'est à dire une solution dont les résultats sont réellement utilisables par d'autres algorithmes).

Comme nous l'avons vu, l'un des principaux inconvénients de notre approche est sa paramétrisation. Le chapitre 6 propose une méthode de paramétrisation par apprentissage. On se place pour cela dans le cadre des algorithmes génétiques. Dans ce paradigme, un jeu de paramètres est vu comme un individu appelé chromosome. Un ensemble de chromosomes est appelé une population. Le principe général d'apprentissage par algorithme génétique consiste à faire évoluer cette population jusqu'à un état suffisamment proche de la solution cherchée, c'est à dire une population contenant un chromosome associé à un jeu de paramètres efficaces. Le principe d'évolution consiste à évaluer la qualité de chaque jeu de paramètres (c'est à dire le chromosome) afin de n'en garder, pour la population suivante, qu'une combinaison des plus efficaces. De façon plus précise, notre approche consiste à se doter d'une séquence vidéo connue associée à un résultat idéal (un graphe optimal) que l'on souhaiterait obtenir par le suivi de personne et de faire évoluer une population en comparant le résultat obtenu par les chromosomes de cette population au résultat idéal.

Les perspectives dans ce domaine sont nombreuses. Dans le cadre de notre approche tout d'abord, deux pistes sont envisagées. La première piste de recherche consiste à aug-

menter le nombre de paramètres du chromosome, afin d'en accroitre les potentialités. A ce titre, certaines expérimentations non décrites dans ce manuscrit ont été menées. L'idée fut d'une part de s'affranchir de l'hypothèse $\beta = 0$ (c'est à dire de passer d'un modèle de suivi ne dépendant que des conditions initiales) à β quelconque (c'est à dire un modèle de suivi autonome) et d'autre part d'éliminer l'hypothèse de linéarité entre les λ_i. Dans ces expérimentations, le chromosome était composé de 25 paramétres au lieu de 17 pour le modèle de chromosome presenté dans ce manuscrit. Les résultats de ces expérimentations furent tout aussi encourageants que les résultats présentés dans le chapitre 6 (c'est à dire que l'augmentation du nombre des paramètres n'est pas une limite de l'apprentissage).

Toujours dans le cadre de notre approche, une seconde piste de recherche est liée à l'amélioration de la fonction d'évaluation (la fitness). En effet, nous avons vu que, bien que la fonction utilisée dans le protocole présenté dans ce manuscrit offrait un certain nombre d'avantages tels que sa simplicité et son interprétation directe, cette fonction a certaines limitations que l'on imagine pouvoir dépasser (notamment en ce qui concerne la prédominance de la recherche d'isomorphisme des graphes par rapport à la minimisation de la distance spatiale). On pense en fait qu'un mode d'évaluation global basé sur du matching de graphe serait plus à même de s'affranchir de ce problème.

D'une façon plus générale, la paramétrisation d'algorithmes semble être une voie à part entière dont le préalable serait une étude sur la sémantique des parametres. En effet, une rapide classification nous amène à penser que tous les paramètres n'ont pas le même genre d'influence sur les performances d'un algorithme donné. On entrevoit trois catégories de parametres: les paramètres de balances dont la valeur va influencer un certain type d'erreur par rapport à un autre (par exemple α: la valeur de seuillage de notre segmentation), les paramètres de tolérance dont l'existence n'a de sens que par rapport à une faiblesse du modèle (par exemple ν: la valeur de durée de vie initiale) et les faux paramètres dont la sémantique s'apparente plus à une constante dont on ne connait pas la valeur (par exemple γ: la valeur de l'influence de la densité de pixels mobiles sur l'heuristique κ).

L'un des enjeux d'une étude sur la paramétrisation de façon générale (c'est à dire comprendre comment un paramètre donné d'un algorithme donné influence les résultats de cet algorithme) est de passer d'un schéma où le but de l'apprentissage est de trouver les meilleures valeurs possibles à un schéma où le but de l'apprentissage est de trouver des règles permettant de calculer les meilleures valeurs.

Nous avons tenté à travers les méthodes proposées dans le chapitre 6, de montrer que l'apprentissage pour la paramétrisation d'algorithmes de vision était une problématique réelle susceptible de trouver une solution réaliste.

Enfin, la dernière partie de ce manuscrit a eu pour objet le problème de reconnais-

sance des comportements proprement dite. Comme nous l'avons vu, ce problème pose deux problèmes distincts: la représentation des comportements et la reconnaissance des comportements. Nous avons proposé dans le chapitre 7, un formalisme permettant de représenter les comportements comme des ensembles de prédicats booléens dont les variables sont des sommets du graphe d'interprétation. Nous avons proposé en outre un algorithme de reconnaissance. Le principe de cet algorithme est de convertir chaque modèle de comportement en un ensemble de modèles dont la reconnaissance peut être résolue comme un problème de satisfaction de contraintes. Le chapitre 8 a montré que cette approche était suffisamment générique et efficace pour s'appliquer à des environnements, des contraintes et des objectifs très différents.

Comme il l'a déjà été dit, l'un des points clefs de cette approche réside dans le fait que tout type de comportements est représentable de la même façon. En d'autres termes, le formalisme de description d'un événement est le même que le formalisme de description d'un scénario long terme. L'enjeu de cette unification est relatif à certaines expérimentations non décrites dans ce manuscrit dont l'objectif fut la manipulation formelle de modèles de comportements pour l'optimisation de leur reconnaissance. En effet, si les formalismes de description de comportements diffèrent en fonction de leur nature, les seules optimisations possibles doivent être faites "à la main" par l'expert. En revanche, si le formalisme de description est unique, alors ces optimisations peuvent être automatisées.

L'étude menée sur la manipulation de modèle de comportement a mis en évidence différents types de manipulations possibles. En premier lieu, on pense aux manipulations sur un modèle donné que l'on apparente à de la vérification de modèle: vérification syntaxique, vérification logique et vérification arithmétique.

Mais la piste la plus intéressante concernant la manipulation de modèles de comportements est liée aux manipulations de plusieurs modèles. On met dès lors en évidence deux types d'opérateurs manipulant plusieurs modèles en même temps: le développement et la factorisation. Le développement a pour objectif de convertir deux modèles A et B, A nécessitant une occurrence de B en un modèle C tel que le modèle C ait les mêmes conditions de réalisation que B. Prenons comme exemple les deux évènements "une personne entre" et "une personne sort". On souhaite alors reconnaître toutes les occurrences du scénario "entre et sort" que l'on construit alors. La base de comportement de cette application est composée de trois comportements: l'évènement "une personne entre", l'évènement "une personne sort" et le scénario "entre et sort", alors que seules les occurrences de "entre et sort" sont intéressantes. Dans ce cas le développement des deux évènements par rapport au scénario amènerait à une base de comportements ne comprenant qu'un unique modèle. De la même façon, pour expliquer l'idée de la factorisation, prenons deux scénarios "une

personne entre et sort" et "une personne entre et s'assoit". Dans ce cas là, une factorisation possible consisterait à extraire des deux modèles, la partie commune correspondant à l'évènement "une personne entre" afin que les prédicats qui le composent ne soient vérifiés qu'une seule fois par frame.

L'enjeu de l'utilisation des deux opérateurs factorisation / développement sur une base de comportements est d'optimiser sa reconnaissance. Une étude non décrite dans ce manuscrit laisse entrevoir un certain nombre de conclusions que voici.

En premier lieu, le caractère automatique de ces deux opérateurs est très différent. En effet, l'opération de développement d'un ensemble de modèles en un seul modèle est unique et non-ambiguë. En revanche, l'opération de factorisation d'un ensemble de modèles n'est pas unique. Une infinité de possibilités d'ensembles de modèles différents peut être obtenue par factorisation. En d'autres termes, l'opération de factorisation est plus complexe à automatiser.

En second lieu, le gain en terme de temps de calcul obtenu par factorisation / développement est difficile à évaluer *a priori*. Prenons comme exemple le cas un scénario "une personne s'approche de la machine A" composé d'un évènement "une personne s'approche de". Le résultat du développement de ces deux modèles est particulièrement efficace du point de vue de la reconnaissance. En effet on passe d'une reconnaissance de deux modèles composés respectivement de 7 prédicats à 3 variables et 12 prédicats à 4 variables à la reconnaissance d'un unique modèle composé de 10 prédicats à 2 variables. Dans ce cas, l'opération de développement est particulièrement efficace.

En revanche si l'on prend l'exemple du développement du scénario "entre et sort", celui-ci est très peu efficace dans la mesure où le scénario obtenu est caractérisé par une durée égale à la durée du scénario "entre et sort", mais dont les variables sont des sommets *person*. I.e. un scénario très peu contraint donc coûteux à reconnaître.

Nous avons tenté de montrer au travers des chapitres 7 et 8 qu'il était possible d'apporter une solution réaliste au problème de la reconnaissance de comportements. Nous avons tenté de montrer en outre, que la multiplication des formalismes et des algorithmes n'est pas nécessaire pour obtenir une approche robuste. Au contraire, l'accent mis sur l'unification des formalismes nous a permis de mener à bien un certain nombre d'applications de naturs très diverses.

La philosophie générale de notre approche fut de tenter de fournir une solution globale efficace et générique au problème de la reconnaissance de comportement humain à partir de séquences vidéos. Cette démarche en plus d'avoir montré son efficacité nous a permis d'ouvrir des perspectives sur de futures recherches dans des domaines aussi divers que la segmentation d'images, les méthodes heuristiques, les méthodes de diagnostics, l'apprentis-

sage, la représentation de connaissance ou la reconnaissance de modèles spatio-temporels.

Bibliographie

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, March 1999.

[2] K. Akita. Image sequence analysis of real world human motion. *Pattern recognition*, 17(1):73 – 83, 1984.

[3] E. André, G. Herzog, and T. Rist. On the simultaneous interpretation of real world image sequences and their natural language description: The system soccer. In *8th European Conference of Artificial Intelligence*, pages 449 – 454, Munich, 1988.

[4] A.F. Bobick ans S. Intille and Y. Ivanov. Representation of multi-agent action for recognition. In *Third International Workshop on Cooperative Distributed Vision*, Kyoto, Japan, November 1999.

[5] H. Araujo and C. Brown. A note on lowes tracking algorithm. Technical Report 610, The University of Rochester Computer Science Department, Rochester, New York, April 1996.

[6] D. Ayers and M. Shah. Monitoring human behavior in an office environment. In *Computer Society Workshop on Interpretation of Visual Motion*, 1998.

[7] Bouchon-Meunier B. *La logique floue et ses applications*, volume 1. Addison-Wesley France, 1995.

[8] J.L. Barron, D.J. Fleet, and S.S Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43 – 77, 1994.

[9] A. Baumberg and D. Hogg. Learning flexible models from image sequences. Technical Report 93.36, University of Leeds, October 1993.

[10] A. Baumberg and D. Hogg. An adaptative eigenshape model. In *British Machine Vision Conference BMVC*, Birmingham, 1995.

[11] B. Bennett. Spatial reasonning with propositonal logics. In *4th International Conference on Knowledge Representation and Reasoning*, 1994.

[12] B. Bennett. Towards a decision procedure for the rcc theory of spatial regions. In *AISB workshop on Automated Reasoning*, University of Sheffield, April 1995.

[13] P. Bouthemy. Modèles et méthodes pour l'analyse du mouvement dans une séquence d'images. *Tecnique et Science Informatique*, 7(6):527–546, 1988.

[14] F. Bremond. *Environnement de resolution de problemes pour l'interpretation de sequences d'images.* PhD thesis, INRIA - Universite de Nice Sophia-Antipolis, 1997.

[15] F. Brémond and M. Thonnat. A context representation for surveillance systems. In *ECCV Worshop on Conceptual Descriptions from Images*, April 1996.

[16] H. Buxton and S. Gong. Advanced visual surveillance using bayesian networks. In *Workshop on Context-based Vision*, Cambridge, 1995. IEEE.

[17] H. Buxton and S. Gong. Visual surveillance in dynamic and uncertain world. *Artificial Intelligence Journal*, 78:431 – 459, 1995.

[18] N.J. Byrne, A. Baumberg, and D. Hogg. Using shape and intensity to track non-rigid objects. Technical Report 94.14, University of Leeds, May 1994.

[19] J. Canny. A computational approach to edge detection. *PAMI*, 8:679 – 698, 1986.

[20] C. Castel, L. Chaudron, and C. Tessier. What is going on? a high level interpretation of sequences of images. In *4th European Conference on Computer Vision, Workshop on Conceptual Descriptions from Images*, Cambridge UK, April 1996.

[21] T. Cham and J.M. Rehg. A multiple hypothesis approach to figure tracking. In *Computer Vision and Pattern Recognition*, Fort Collins, Colorado, June 1999.

[22] Z. Chen and H.J. Lee. Knowledge-guided visual perception of 3d gait from a single image sequence. *IEEE Transactions on systems man and cybernetic*, 22(2):336–342, mar 1992.

[23] N. Chleq, F. Bremond, and M. Thonnat. *Advanced Video-Based Surveillance Systems*, chapter Image Understanding for Prevention of Vandalism in Metro Station, pages 106 –117. Kluwer Academic Publishers, 1999.

[24] N. Chleq and M. Thonnat. Realtime image sequence interpretation for videosurveillance. In IEEE, editor, *International Conference on Image Processing*, pages 801 – 804, Lausanne, Switzerland, 1996.

[25] O. Chomat and J. L. Crowley. Utilisation de champs réceptifs spacio-temporels pour la reconnaissance de l'apparence locale d'activités. In *ORASIS*, Aussois, France, Avril 1999.

[26] H.I. Christensen, J. Matas, and J. Kittler. Using grammars for scene interpretation. In *International Conference on Image Processing*, 1996.

[27] J.M. Chung and N. Ohnishi. Cue circles: Image feature for measuring 3-d motion of articulated objects using sequential image pair. In *3th International Conference on Face and Gesture Recognition*, pages 474 – 478, Nara, Japan, April 1998.

[28] A. G. Cohn, J.M. Gooday, and B. Bennett. A comparison of structures in spatial and temporal logics. In R Casati and G White, editors, *In Philosophy and the Cognitive Sciences*, pages 409 – 422, Vienna, Austria, 1994.

[29] A. G. Cohn, J.M. Gooday, B. Bennett, and N.M. Gotts. A logical approach to representing and reasonning about space. *Artificial Intelligence*, 9:255 – 259, 1995.

[30] V. ColindeVerdiere and J.L. Crowley. Object recognition using local appearance. In *Reconnaissance des formes et Intelligence Artificielle*, volume 2, pages 129 – 136, Clermont-Ferrand, Janvier 1998.

[31] R. Collins, A. Lipton, and T. Kanade. A system for video surveillance and monitoring. In *Proceedings of the American Nuclear Society (ANS) Eighth International Topical Meeting on Robotics and Remote Systems*, April 1999.

[32] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1):38 – 59, January 1995.

[33] L. Davis, E. Borovikov, R. Cutler, D. Harwood, and T. Horprasert. Multi-perspective analysis of human action. In *Third International Workshop on Cooperative Distributed Vision*, Kyoto, Japan, November 1999.

[34] K.M. Dawson-Howe. Active surveillance using dynamic background subtraction. Technical Report 06, Department of Computer Science, Dublin, Ireland, August 1996.

[35] Q. Delamarre. Modélisation de la main pour sa localisation dans une séquence d'images. Technical Report 0198, INRIA Sophia Antipolis, décembre 1996.

[36] S. Dettmer, A. Seetharamaiah, L. Wang, and M. Shah. Model-based approch for recognizing human activities from video sequences. In *Workshop on Motion of Non-Rigid and Articulated Objects*, June 1998.

[37] C. Dousson. *Suivi d'évolution et Reconnaissance de Chroniques.* PhD thesis, Université Paul Sabatier de Toulouse, LAAS-CNRS 7 avenue du Colonel Roche, 31077 Toulouse Cedex, Septembre 1994.

[38] C. Dousson and M. Ghallab. Suivi et reconnaissance de chroniques. *Revue d'Intelligence Artificielle*, 8(1):22 – 61, 1994.

[39] M.S. Drew, J. Wei, and Z. Li. Illumination-invariant color object recognition via compressed chromaicity histograms of normalized images. Technical Report 09, School of Computing Science Simon Fraser University, Vancouver, B.C. Canada, 1997.

[40] D. Dubois and H. Prade. *Théorie des possibilités.* Masson, 1988.

[41] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*, chapter 3, pages 33 – 68. The MIT Press, 1993.

[42] C. Le Gal, J. Martin, and G. Durand. Smartoffice: An intelligent and interactif environment. In *1st International Workshop on Managing Interactions in Smart Environments*, Dublin, 1999.

[43] J.G Ganascia. *L'intelligence Artificielle.* Michel Serres et Nayla Farouki, FLAMMARION, 1993.

[44] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82 – 98, 1999.

[45] D.M. Gavrila and L.S. Davis. Tracking of humans in action: a 3-d model-based approach. In *ARPA Image Understanding Workshop*, Feb 1996.

[46] M. Gelgon, P. Bouthemy, and T. Dubois. A region tracking method with failure detection for an interactive video indexing environment. In *3rd Int. Conf. on Visual Information Systems*, Amsterdam, June 1999.

[47] J.M. Gooday and A.G. Cohn. Conceptual neighbourhoods in temporal and spatial reasoning. In *Workshop on Spatial and Temporal Reasoning*, pages 57 – 63, Amsterdam, August 1994.

[48] M. Grabisch and A. Nifle. Reconnaissance de scénarios temporels fondés sur la logique possibiliste. In *Reconnaissance de Formes et Intelligence Artificielle 2000*, volume II, pages 295 – 305, Fevrier 2000.

[49] W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *CVPR98*, 1998.

[50] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley Publishing Company, Reading, MA, 1992.

[51] I. Haritaoglu. *A Real Time System for Detection and Tracking of People and Recognizing Their Activities*. PhD thesis, niversity of Maryland at College Park, 1998.

[52] I. Haritaoglu, D. Harwood, and L.S. Davis. Hydra: Multiple people detection and tracking using silhouettes. In *2nd International Workshop on Visual Surveillance*, pages 6 – 13, Fort Collins, Colorado, June 1999.

[53] J. Heikkila and O. Silven. A real-time system for monitoring of cyclists and pedestrians. In *2nd International Workshop on Visual Surveillance*, Fort Collins, Colorado, June 1999.

[54] G. Herzog. Utilizing interval-based event representation for incrementatal high-level scene analysis. In *4th International Workshop on Semantics of Time, Space, and Movement and Spatio-Temporal Reasonning*, Chateau de Bonas, France, 1992.

[55] D. Hute, J.P. Mazy, and K. Graf. *The prevention of Vandalism in Metro Stations*, chapter 1.4, pages 34 – 43. Kluwer Academic Publishers, 1999.

[56] P. Huttenlocker. Tracking non-rigid object in complex scenes. In *International Conference on Computer Vision*, Berlin, 1992.

[57] P. Huttenlocker, J.J. Noh, and W. Rucklidge. Tracking non-rigid objects in complex scenes. Technical Report 1320, Computer Science Department Cornell University, Ithaca, NY 14853, 1992.

[58] S. Intille and A. Bobick. Visual tracking using closed-world. Technical report, M.I.T Media Laboratory Perceptual Computing Section, Cambridge, MA 02139, November 1994.

[59] S. Intille and A. Bobick. Closed world tracking. In *5th International Conference on Computer Vision*, Cambridge, 1995.

[60] S. Intille and A. F. Bobick. Visual recognition of multi-agent action using binary temporal relations. In *Computer Vision and Pattern Recognition*, Fort Collins, Colorado, June 1999.

[61] S.S Intille. *Visual Recognition of Multi-Agent Action.* PhD thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, September 1999.

[62] Y. Ivanov, C. Stauffer, A. Bobick, and W.E. Grimson. Video surveillance of interactions. In *2nd International Workshop on Visual Surveillance*, pages 82 – 89, Fort Collins, Colorado, June 1999.

[63] R. Jain, W.N. Martin, and J.K. Aggarwal. Segmentation throught the detection of changes due to motion. *Computer Graphics and Image Processing*, 2:13–34, 1979.

[64] J. Kittler, J. Matas, M. Bober, and L. Nguyen. Image interpretation: Exploiting multiple cues. In *International Conference on Image Processing and Applications*, Edinburgh, June 1995.

[65] M.K. Leung and Y.H. Yang. A region based approach for human body motion analysis. *Pattern Recognition*, 20(3):321 – 339, 1987.

[66] A. Lindivat. Des logiciels qui verrons venir le danger. *L'Ordinateur individuel*, 1(105):104–107, Avril 1999.

[67] H. Liu, T. Hong, M. Herman, and R. Chellappa. Accuracy vs. efficiency trade-offs in optical flow algorithms. In *4th European Conference on Computer Vision*, 1996.

[68] A. Marakov. comparison of background extraction based intrusion detection algorithms. In *International Conference on Image Processing*, volume 1, pages 512 – 524, 1996.

[69] S. J. McKenna, S. Jabri, Z. Duric, and H. Wechsler. Tracking interacting people. In *4th International Conference on Face and Gesture Recognition*, pages 348 – 353, Grenoble, France, March 2000.

[70] G. Medioni, I. Cohen, F. Brémond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. In *DARPA Image Understanding Workshop*, Monterey, November 1998.

[71] F. Meyer and P. Bouthemy. Region-based tracking in an image sequence. In *European Conference on Computer Vision*, pages 476–484, 1992.

[72] R. Mohr and T. C. Henderson. Arc and path consistency revised. *Artificial Intelligence*, 28:225 – 233, 1986.

[73] R. Nikoukhah, S.L. Campbell, and F. Delebecque. Kalman filtering for discrete-time linear system. Technical Report 3343, INRIA, Janvier 1998.

[74] J. Owens and A. Hunter. Application of the self-organising map to trajectory classication. In *Third Visual Surveillance*, Dublin, July 2000.

[75] A. Pentland. Machine understanding human action. In *7th International Forum on of Frontier of Telecommunication Technology*, Tokyo, 1995.

[76] J. Picard. Efficiency of the extended kalman filter for non linear systems with small noise. Technical Report 1068, Institut National de Recherche en Informatique et Automatique, Aout 1989.

[77] M.R. Pickering, J.F. Arnold, and M. Frater. An adaptative block matching algorithm for efficient motion estimation. In *International Conference on Image Processing*, 1996.

[78] C. Pinhanez and A. Bobick. Fast constraints propagation on specialized allen networks and its application to action recognitiion and control. Technical Report 456, MIT Media Laboratory, 20 Ames St. - Cambridge, MA 02139, January 1998.

[79] H. Prade. *Modeles mathematiques de l'imprecis et de l'incertain en vue d'application au raisonnement naturel.* PhD thesis, Universte Paul Sabatier de Toulouse, juin 1982.

[80] Y. Raja, S.J. McKenna, and S. Gong. Tracking and segmenting people in varying lighting conditions using colour. In *3th International Conference on Face and Gesture Recognition*, pages 228 – 233, Nara, Japan, April 1998.

[81] R.P.N. Rao. Robust kalman filters for prediction, recognition and learning. Technical Report 645, Computer Science Departement and University of Rochester, Rochester, New York 14627, December 1996.

[82] P. Regemagnino, S. Maybank, R. Fraile, and K. Baker. *Automatic Visual Surveillance of Vehicles and People*, chapter 3.1, pages 95 – 105. Kluwer Academic Publishers, 1999.

[83] G. Rigoll, S. Eickeler, and S. Muller. Person tracking in real-wolrd scenarios using statisical methods. In *4th International Conference on Face and Gesture Recognition*, pages 342 – 347, Grenoble, France, March 2000.

[84] L. Robert. *Perseption stereoscopique de courbes et de surface tridimensionnelles. Application a la robotique mobile.* PhD thesis, Ecole Polytechnique, 1993.

[85] K. Rohr. Toward model-based recognition of movement in image sequences. *Computer Vision, Graphique and image processing: image understanding*, 59(1):94–115, jan 1994.

[86] N. Rota, R. Stahr, and M. Thonnat. Tracking for visual surveillance in vsis. In *First Workshop on Performance Evaluation of Tracking and Surveillance*, Grenoble, March 2000.

[87] N. Rota, R. Stahr, and M. Thonnat. Ai techniques for vsis human tracker. Technical Report 4138, INRIA, March 2001.

[88] N. Rota and M. Thonnat. Activity recognition from video sequence using declarative models. In *European Conference on Artificial Intelligence*, Berlin, August 2000.

[89] N. Rota and M. Thonnat. Video sequence interpretation for visual surveillance. In *Third Visual Surveillance*, Dublin, July 2000.

[90] N. A. Rota. Système adaptatif pour le traitement de séquences d'images pour le suivi de personnes. Master's thesis, DEA IARFA Laboratoire d'informatique de Paris IV, septembre 1998.

[91] D. Salber, J. Coutaz, D. Decouchant, and M. Riveill. De l'observabilite et de l'honnetete: le cas du controle d'acces dans la communication homme-homme mediatise. In *Interacion Homme Machine*, pages 27 – 34, Toulouse, octobre 1995.

[92] A. Sato, K. Mase, A. Tomono, and K. Ishii. Pedestrian counting system robust against illuminastion changes. In *Visual Communication and Image Processing*, Massachusetts, 1993.

[93] E. Sender. C'est arrivé demain : Big brother parmi nous. *Sciences et avenirs*, 1(635):74–77, Janvier 2000.

[94] M.C. Shin, D. Goldgof, and K.W. Bowyer. Comparison of edge detectors using an object recognition task. In *Computer Vision and Pattern Recognition*, Fort Collins, Colorado, June 1999.

[95] T. Strat. Employing contextual information in computer vision. In *DARPA93*, pages 217–229, 1993.

[96] C. Tessier. Reconnaissance de scènes dynamiques à partir de données issues de capteurs : le projet Perception. In *Agard-CP-595 - Multi-sensor systems and data fusion for telecommunications, remote sensing and radar*, pages 14–1–9, Lisbonne, Portugal, 1997.

[97] H. Wang and M. Brady. Real-time corner detection algorithm for motion estimation. *Image and Vision Computing*, 13:695–703, 1995.

[98] G. Welch and G. Bishop. An introduction to kalman filter. Technical Report 95-041, Departement of Computer Science and University of North Carolina, Chapel Hill NC 27599-3175, December 1995.

[99] O.S. Wenstop. Motion detection for image information. In *3th Scandinavian conference on image analysis*, pages 381 – 386, Tromso Norway, 1983.

[100] C.R. Wren and A.P. Pentland. Dynamic models of human motion. In *3th International Conference on Face and Gesture Recognition*, pages 22 – 27, Nara, Japan, April 1998.

[101] M. Yamada, K. Ebihara, and J. Ohya. A new robust real-time method for extracting human silhouettes from color images. In *3th International Conference on Face and Gesture Recognition*, pages 528 – 533, Nara, Japan, April 1998.

[102] Z. Zhang. Parameter estimation technique: A tutorial with application to conic fitting. Technical report, Institut National de Recherche en Informatique et Automatique, Octobre 1995.

Resume

Les recherches dont ce document fait état sont relatives à un programme appelé VSIS. Le but de ces recherches est l'élaboration d'un programme capable de reconnaître certain comportements humains à partir d'un flux vidéo. Le principe général consiste à se doter, outre le flux vidéo issu d'une caméra fixe, d'un ensemble de descriptions des comportements que l'on souhaite reconnaître d'une part, et d'autre part d'une description du décor dans lequel est filmé la scène. Ainsi, on extrait du flux vidéo à temps fixe, les indices du mouvement des personnes, afin de calculer une description de ceux-ci. C'est à dire estimer leur taille ou leur volume et les replacer dans le décor. Ce problème est résolu par parcours heuristique de l'ensemble des descriptions possibles. Avec cette description complète de la scène (personnes + décor) à un instant donné, on calcule la façon dont la scène a évoluée afin d'obtenir une description spatiale et temporelle de la scène. La méthode utilisée pour cela est basée sur le calcul d'un diagnostic le plus vraisemblable de l'évolution de la scène. A partir de cette description spatiale et temporelle les différents comportements prédéfinis pourront être reconnus. La méthode retenue pour cela est basée sur la résolution d'un ensemble de problèmes de satisfaction de contraintes obtenus à partir des modèles de comportements prédéfinis.

Abstract

This document presents research related to a program called VSIS. The aim of those research is the elaboration of a program able to recognise human behaviours from video sequences. The main principle is to build a set of descriptions of the behaviours, we want to recognise, and a description of the environment where the scene takes place. Then, we extract from the video stream, at constant time, the clues of the human motion in order to compute a description of each human. (E.g location, height, volume). This problem is solved by an heuristic search among all the possible descriptions of the scene. With this complete description of the scene, we compute the evolution of the scene between two instant in order to obtain a spacial and temporal description of the scene. The proposed method is based on the computing of the most likely diagnosis of the evolution of the scene. With the spacial and temporal description of the scene the predefined behaviours can be recognised. The proposed method is based on the solving of a set of constraints satisfaction problems obtained by translation of the predefined behaviours models.