Université de Montréal

Système de Vidéosurveillance et de Monitoring

par

Mohamed Dahmane

Département d'informatique
et de recherche opérationnelle

Faculté des arts et sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M. Sc.)
en informatique

Octobre 2004

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

Système de Vidéosurveillance
et de Monitoring

présenté par :

Mohamed Dahmane

a été évalué par un jury composé des personnes suivantes :

Sébastien Roy
(président-rapporteur)

Jean Meunier
(directeur de maîtrise)

Max Mignotte
(membre du jury)

Mémoire accepté le

21 décembre 2004

Sommaire

En vidéosurveillance numérique, la reconnaissance du mouvement humain est souvent l'objectif final, ceci est le résultat de l'analyse du contenu du flux vidéo, souvent en temps réel afin de tirer l'information pertinente. Dans ces systèmes automatiques, globalement les simples détections de mouvement généralement basées sur de simples différences d'images ne sont pas, forcement, suffisantes et ne prennent pas en charge l'aspect symbolique de l'information visuelle que procurent les séquences d'images prises par les caméras vidéos.

L'approche à mettre en œuvre vise à reproduire automatiquement le processus de vidéosurveillance tel que effectué par l'opérateur humain en imitant son propre processus de reconnaissance portant sur la détection, le suivi et le cas échéant le déclenchement d'alarme. Le but fondamental est d'implanter un système "autonome" intégrant des modules de reconnaissance de comportements, de gestion d'alarmes et d'enregistrement intelligent, impliquant ainsi des solutions basées sur des systèmes temps réel capables d'interagir intelligemment, facilement et efficacement avec l'environnement.

Dans ce mémoire, on décrit une approche pour la reconnaissance de comportements humains qui s'inscrit dans un projet de télésurveillance médicale dont le but est de pouvoir détecter des situations atypiques à risque tout en minimisant les fausses alarmes. Les méthodes et algorithmes utilisés dans un tel système sont issus de reconnaissance de formes et d'apprentissage. Les résultats obtenus (un taux de succès de 97.47% de bien-détectés et 95.23% de reconnaissance correcte) sont encourageants mais encore assez limités par rapport à un vrai système intelligent.

Mots clés : vidéosurveillance, détection de mouvement, segmentation temps-réel, poursuite de cibles, trajectoires spatio-temporelles, réseaux de neurones.

Abstract

In digital visual surveillance, the recognition of human movement is often the final purpose; this is the result of the analysis of the contents of the video flow, often in real-time streaming in order to draw-out relevant information. In these automatic systems, globally the simple detection of movement, generally based on simple image subtraction, is not necessarily sufficient and does not deal with the symbolic aspect of the visual piece of information given away by the video sequences. The approach to be implemented aims to reproduce automatically the process of video surveillance such as carried out by a human operator by imitating his own process of recognition wearing on detection, keeping track and if necessary alarm setting.
The fundamental purpose is to implant an "autonomous" system integrating modules of behavior recognition, management of alarms and intelligent recording, thus implying solutions based on real-time systems able to interacting intelligently, easily and effectively with the environment.

In this thesis, we describe a framework to human behavior recognition, which will go with in a generic telecare architecture and for which the purpose is to be able to detect atypical situations in realistic scenarios while minimizing the false alarms. The methods and algorithms used in such system are directly arising from pattern recognition techniques and learning algorithms. The obtained results (a success rate of 97.47% of true-detection and 95.23% of correct recognition) are quite encouraging but still enough restricted with regard to a veritable intelligent system.

Keywords : automated visual monitoring, movement detection, temporal segmentation, people tracking, spatiotemporal trajectories, neural networks.

Table des matières

Sommaire i

Abstract ii

Table des matières iii

Liste des tableaux vi

Table des figures vii

Remerciements x

Dédicaces xi

1 Introduction **1**

1.1 Motivation 1

1.2 Présentation du mémoire 5

2 Système de vidéosurveillance **6**

2.1 Détection 7

2.1.1 Prétraitements 8

2.1.2 La segmentation d'objets mobiles 8

2.1.3 Détection de l'ombre 12

2.1.4 La validation du masque de changement 14

2.1.5 La mise à jour du modèle de l'arrière-plan 15

2.1.6 La classification des entités en mouvement 16

2.2 Suivi temporel 17

2.2.1 Suivi par modèle 18

2.2.2 Suivi par région 19

2.2.3 Suivi par contours actifs 19

2.2.4 Suivi par attributs 20

2.3 Analyse et reconnaissance d'activités 20

2.3.1 Algorithme de recalage temporel 20

2.3.2 Les modèles de Markov cachés 21

2.3.3 Les réseaux de neurones artificiels 22

3 Systèmes Connexionnistes **24**

3.1 Propriétés des réseaux de neurones 26

3.1.1 Topologie d'interconnexion 26

3.1.2 Propriétés des unités 28

3.1.3 Lois d'apprentissage 29

3.2 Les réseaux de neurones incorporant la mémoire 30

3.2.1 Les réseaux de Hopfield 31

3.2.2 Les réseaux d'Elman 32

3.2.3 Les réseaux de neurones à codage temporel 32

3.3 La carte auto-organisatice de Kohonen 33

3.3.1 Principe 34

3.3.2 Propriété topologique 36

4 Méthodologie **38**

4.1 Estimation de l'arrière-plan 38

4.1.1 Le modèle du fond 39

4.1.2 La soustraction du fond 41

4.1.3 La mise à jour du fond 43

4.2 Étiquetage en composantes connexes 44

4.2.1 Algorithme de gestion de partitions 45

4.2.2 L'étiquetage 47

4.3 Poursuite de cibles 49

4.3.1 Notion de matrices de mise en correspondance 49

4.3.2 Évolution temporelle des blobs et des évènements 51

4.4 Reconnaissance par approche comportementale 52

4.4.1 Paramétrisation de trajectoires 53

4.4.2 Détection de comportements anormaux par la SOM 56

5 Résultats et discussion **58**

5.1 La base de données des scénarios 58

5.2 Résultats 59

5.2.1 Détection et classification des changements 59

5.2.2 Suivi 60

5.2.3 Analyse de comportements 64

6 Conclusion et perspectives **69**

Liste des tableaux

4.1 Différentes configurations des séries de correspondance. 51

5.1 Exemples de séquences normales. 59

5.2 Exemples de séquences anormales. 59

5.3 Performances moyennes du modèle conique. 60

5.4 Paramètres de la structure hiérarchique. 64

5.5 Matrice de confusion. 65

5.6 Paramètres des deux SOMs. 67

5.7 Matrice de confusion. 67

Liste des figures

2.1 Pipeline typique d'un processus de vidéosurveillance. 6

2.2 Schéma de détection. 7

2.3 Types de changements dans une scène. 8

2.4 Distorsion en chrominance. 13

2.5 Algorithme classique de mise à jour. 15

3.1 Processus de reconnaissance de formes. 25

3.2 Réseau multicouche récurrent 27

3.3 Réseaux de neurones à codage temporel 33

3.4 Principe du modèle de Kohonen 34

3.5 Carte auto-organisatrice de Kohonen 35

4.1 Distorsion (**a**) de brillance - (**b**) chromatique 40

4.2 Plan d'équibrillance $BCK_{\perp}(x)$ associé à $BCK(x)$ dans le cube RGB. 40

4.3 Le volume d'ombre Γ. 41

4.4 Effet "Jitter" de la caméra en partie dû à des vibrations provoquées par une bouche d'aération tout près. 45

4.5 Représentation d'une partition. 47

4.6 Exemple d'étiquettage - Sauvegarde(*) et Résolution(**) des équivalences . . 48

4.7 Exemple d'évolution temporelle de blobs. 50

4.8 Séquences de segments à différentes résolutions temporelles. 54

4.9 Approximation elliptique de contour. 55

5.1 Résultats de détection / Séquence de test *Plancher/Corridor/Diro/UdeM.* . . 61

5.2 *Intelligent room*-http ://cvrr.ucsd.edu/aton/shadow (*)Statistique Paramétrique (section 2.1.3.A) : relativement, la personne a été bien détectée, dans notre cas on a moins de pixels *ombre* ceci est dû à un τ_1 plus faible (0.75) 61

5.3 Séquence de test *Laboratory raw* (http ://cvrr.ucsd.edu/aton/shadow). . . . 62

5.4 Résultats de détection / Séquence de test *Escalier/Corridor/Diro/UdeM.* . . 62

5.5 Séquence de test *Lab. vision 3D/Diro/UdeM.* 62

5.6 Suivi de cibles (groupement et séparation). 63

5.7 Trajectoires anormales des centroides des cibles (FIG.5.6). 64

5.8 Exemples de scénarios normaux correctement reconnus. 65

5.9 Cas de séquences présentant de fausses alarmes. 66

5.10 Cas d'omission nécessitant une plus haute résolution. 66

5.11 Exemples de comportements atypiques correctement détectés. 67

5.12 Cas de séquences présentant de fausses alarmes. 68

Remerciements

Au terme de ce travail, j'exprime mes remerciements et ma gratitude, les plus profonds, à toutes et à tous qui ont contribué à la préparation et à l'aboutissement de ce mémoire :

Monsieur Jean MEUNIER, professeur titulaire et Directeur du groupe imagerie au département d'informatique et R.O, pour avoir accepté de diriger ce travail. Sa disponibilité constante, son soutien financier et ses critiques associées à ses conseils ont largement contribué à l'aboutissement de ce travail. L'apport de ses orientations et ses remarques à la fois rigoureuses et objectives, a été plus qu'indispensable à l'amélioration progressive du contenu de ce manuscrit.

Monsieur Sébastien ROY, Directeur du laboratoire de vision-3D, pour l'honneur qu'il me fait en présidant le jury de ce travail.

Monsieur Max MIGNOTTE, professeur adjoint au DIRO, pour avoir accepté d'être membre de jury de ce mémoire.

Je leur exprime mes remerciements, les plus vifs.

Un remerciement, tout particulier, à Jean-Philippe (qui se reconnaîtra) qui n'a pas ménagé son temps pour la réalisation de l'ensemble de scénarios.

Un grand merci à tous mes collègues du laboratoire de vision-3D, sans lesquels le climat de travail n'aurait pas été aussi harmonieux et stimulant.

Dédicaces

Je dédie ce mémoire à ceux qui m'ont donné le jour pour ensuite me tendre la main et m'aider à marcher dans la lumière, à mes frères et sœurs qui n'ont jamais quitté ma pensée.

Chapitre 1

Introduction

Cette thèse présente une implémentation d'un système de vidéosurveillance automatique, destiné à faire partie d'un projet de télésurveillance médicale, dans le cadre du concept de salle clinique "intelligente". Le travail consiste en la conception, l'implémentation et l'expérimentation du système, en temps réel, à l'aide d'une caméra fixe installée dans une salle. Avant de décrire ces aspects, la présente section présente la motivation du sujet de cette thèse et expose son organisation.

1.1 Motivation

De nos jours, la télésurveillance a pris une place de plus en plus importante dans la société, ceci est principalement dû à l'intérêt qu'elle procure dans la prévention, la sécurité et plus particulièrement dans les études cliniques. La technologie a atteint une étape où le montage d'un système de cameras pour la capture d'images vidéo est bon marché, cependant chercher les ressources humaines pour effectuer la tâche "fatidique" d'observation est coûteux. Généralement, ces

systèmes enregistrent périodiquement le flux d'information sur des supports appropriés. Ce n'est qu'après que le fait soit accompli que les chargés de sécurité passent en revue les archives vidéo pour voir ce qui c'est effectivement passé. L'intégration de techniques de reconnaissance de mouvement à la vidéosurveillance est une option ouverte pour éviter de telles situations, en permettant de déclencher une alarme, au moment opportun, grâce à un contrôle et une analyse continuels et en temps réel du flux vidéo ; plutôt que des solutions simplistes de détection de mouvement basées sur de simples différences d'images ou de soustraction de fond, dont les performances sont souvent altérées par des fausses alarmes dues, particulièrement, aux effets environnementaux externes tels que le vent, le changement d'éclairage, ... etc.

La vidéosurveillance est en pleine mutation : des solutions simples basées sur les CCTV (Closed Circuit Television) analogiques conventionnels vers les systèmes de 3^e génération à "caméras intelligentes" [60]. Cette évolution qu'a permis la recherche est aussi bien due aux motivations qu'aux exigences dans divers domaines d'application tels que le contrôle des accès aux bâtiments (gouvernementaux, privés, publics, ...), la surveillance des parkings, des stations de métro, des ATMs, le contrôle de la circulation des véhicules et des piétons, voire même le soutien opérationnel des officiers dans les affrontements militaires.
Le but commun de toutes ces finalités est de procéder à un filtrage automatique de l'information pour ne présenter à l'opérateur que des données représentant des situations suspectes en ce qui concerne le sujet de la surveillance.
En outre, dans un système de surveillance traditionnel, l'opérateur humain contrôle, en permanence, des colonnes d'écran CCTV, pour appréhender des entités ou des événements spécifiques. Ce mode de fonctionnement, jugé très pénible, devient aussitôt impraticable lorsqu'il s'agit de couvrir de grandes étendues, en raison de la quantité d'information. Dans ces situations, les systèmes de vidéosurveillance intelligents sont plus adaptés, en raison de la fonctionnalité intégrée qu'ils offrent

grâce au mode coopératif des senseurs multiples, permettant ainsi d'effectuer des suivis automatiques d'un site à un autre. Ceci améliore, considérablement, la fiabilité, car c'est à travers la succession temporelle des événements qu'on peut déterminer si une entité particulière constitue une cible potentielle ou non [8].
D'autre part, les progrès enregistrés dans les systèmes d'acquisition de vidéo de qualité supérieure disponibles à des coûts raisonnables, ont ravivé tant l'intérêt du public que celui des chercheurs pour ouvrir de nouveaux horizons pour la vidéosurveillance automatique. Actuellement, deux projets de grande envergure sont développés : le projet VSAM [1] aux E.U, dont l'objectif est de fournir un système permettant d'analyser des scènes vidéo, incluant des algorithmes de détection et de poursuite de cibles en temps réel à partir de plate forme de caméras stationaire ou aéroportée, classification d'objets (humains, véhicules), analyse de gestes ... etc, pour des applications de surveillance et de contrôle de sites et d'assistance aux militaires. Un projet similaire est développé en Europe intitulé IMPROOFS [2] incluant des procédures d'extraction de mesures à partir d'images (exp. hauteur d'une personne), reconstruction de scène 3D, analyse de mouvement à partir de séquences vidéo ... etc.

Cette thèse s'inscrit dans une démarche de conception d'un système de vidéo-surveillance, basé sur la reconnaissance de comportements, qui sera capable de reconnaître des mouvements par un apprentissage à partir d'un ensemble d'exemples sans l'incorporation d'aucune forme de connaissance à priori concernant le comportement humain. Le système sera en mesure de minimiser les fausses alarmes, à même de distinguer entre activités autorisées et non autorisées, habituelles et inhabituelles. Le système développé consiste en trois modules : un module de détection, un module de poursuite de cibles et d'extraction de caractéristiques

[1] Video Surveillance And Monitoring. [8]
[2] http ://www.robots.ox.ac.uk/ improofs/.

et enfin un module de décision. La reconnaissance est basée sur des attributs globaux du mouvement extraits à partir du flux vidéo.

Notre étude porte, essentiellement, sur deux expérimentations :

- Une architecture SOFMNN [3] à trois niveaux pour l'apprentissage des caractéristiques des trajectoires normales, et pour en détecter les nouvelles. Ces caractéristiques concernent le vecteur flux du mouvement du centre de gravité du blob dont les composantes sont la position et les informations du 1° ordre (vitesse de déplacement). Les niveaux de SOFMNN expriment la résolution temporelle d'inspection.
- Dans une deuxième expérimentation, on considère des propriétés locales du mouvement (position et vitesse instantanée de déplacement), et d'autres plus globales qu'on exprime par les descripteurs elliptiques de Fourier.

Dans ces expérimentations, les informations traitées relèvent purement du 2D en utilisant une QuickCam Pro 4000 de Logitech®, qui offre une mise en œuvre d'une grande simplicité et d'un moindre coût grâce à son capteur CCD VGA haute qualité (jusqu'à 640x480 pixels). Sa performance est principalement due à un débit d'images qui peut atteindre les 30 images/sec qui seront directement analysées dispensant ainsi du temps nécessaire à la compression/décompression et à la sauvegarde. Grâce à sa connexion USB, une norme qui permet un chaînage de périphériques, il est possible de connecter au même poste plus d'une caméra.

[3] Soft Self Organising Feature Map Neural Network.

1.2 Présentation du mémoire

Le chapitre II décrit un système typique de vidéosurveillance et présente ses différents aspects. On énonce chaque module dans une section, tout en se penchant sur les problèmes majeurs qui surgissent et les différentes techniques qui sont adoptées en abordant brièvement quelques travaux récents, qui se montrent plus intéressants et plus originaux.

Le chapitre III nous permet d'aborder les systèmes connexionnistes, en signalant les avantages qu'ils présentent par rapport aux techniques traditionnelles de reconnaissance de formes. Cette section recouvre également quelques réseaux de neurones incorporant une mémoire étant donné le contexte spatio-temporel dans lequel est censé évoluer le système.

Cette étude bibliographique nous a permis d'explorer la matière pour dégager une méthodologie dont les notions importantes et les solutions adoptées seront détaillées à travers les sections du chapitre IV.

Le chapitre V présente nos résultats à partir de séquences vidéo typiques. Nous traiterons également dans ce chapitre de l'évaluation expérimentale de notre démarche qui nous permettra, finalement à travers le chapitre VI, d'entrevoir des perspectives d'amélioration de performance et d'évolution de capacité de reconnaissance du système.

Chapitre 2

Système de vidéosurveillance

Dans cette section, on décrit le processus de vidéosurveillance. En signalant, à chaque étape, un passage en revue des principales stratégies adoptées [1]. Avant de présenter les expérimentations et les résultats obtenus, on définit premièrement les procédures de détection, du suivi et le processus de reconnaissance.

Le rôle d'un système de vidéosurveillance automatique est de détecter un comportement humain suspicieux. Pour ce faire, le système doit procéder comme un filtre, en détectant des comportements hors du commun pour attirer l'attention de l'opérateur humain sur de tels évènements.
Ce rôle se résume sous forme d'un pipeline modulaire (FIG.2.1).

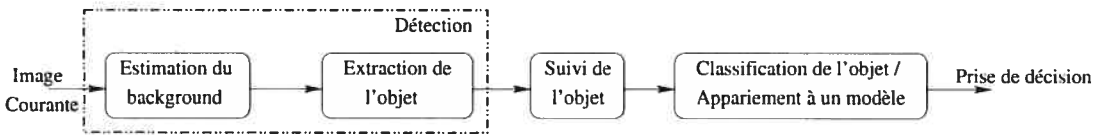


FIG. 2.1 – Pipeline typique d'un processus de vidéosurveillance.

[1] pour une revue de littérature détailée voir [72]

2.1 Détection

La première étape d'un système de surveillance est l'identification des objets mobiles, ainsi que, les objets statiques qui ne font pas partie d'une scène de référence. Cette étape est constituée de modules de bas niveau (FIG.2.2), elle concerne la segmentation des objets mobiles et leur classification. C'est une tâche

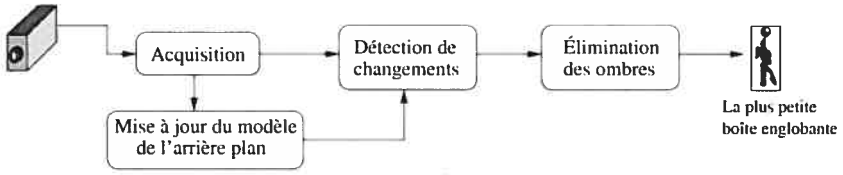


FIG. 2.2 – Schéma de détection.

difficile, surtout dans le cas d'objets formés de plusieurs parties aux mouvements différents comme c'est le cas lorsque l'humain constitue le sujet de la surveillance.

La détection des entités mobiles implique la détection des variations temporelles de l'intensité lumineuse. Cependant les changements dus aux défauts d'illumination et plus particulièrement les effets dus à l'ombre (FIG.2.3) altèrent sévèrement le processus de segmentation, lorsque la scène n'est observée que par une seule caméra. Dans la littérature, les techniques de détection de changement sont divisées en deux classes :

- Les techniques basées sur un seul pixel.
- Les techniques locales utilisant un bloc de pixels.

Un algorithme classique de détection de changement génère, à partir d'une séquence d'images, une image binaire $B : R \rightarrow \{0, 1\}$ appelée le masque de changement selon la règle générique :

$$B(x, y) = \begin{cases} 1 & \text{s'il y changement significatif} \\ 0 & \text{sinon} \end{cases}$$

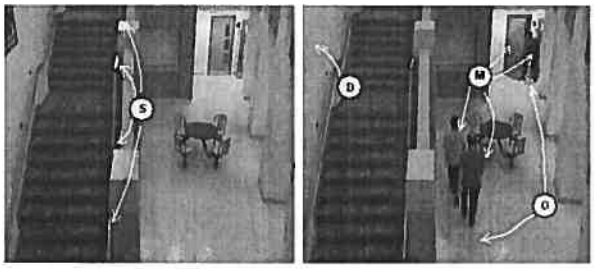


FIG. 2.3 – Types de changements dans une scène.

2.1.1 Prétraitements

Plusieurs approches de détection de changement sont précédées de prétraitements nécessaires pour la suppression de changements jugés *moins importants* qui impliquent, généralement, des rectifications géométriques et des réajustements d'intensité [58].

2.1.2 La segmentation d'objets mobiles

La segmentation des images acquises est une étape nécessaire pour l'extraction des entités mobiles présentes dans une scène afin de pouvoir étudier leur comportement dynamique. Les techniques de segmentation spatio-temporelle sont nombreuses et dépendent des caractéristiques de la scène observée, les plus générales, sont [61] :

A. Le flux optique

Cette technique utilise le champ de vecteur des vitesses apparentes des objets de la scène sur le plan image pour extraire les structures mobiles [18, 73, 35]. Les méthodes d'extraction du flux optique sont répertoriées en deux grandes familles :

1. **Méthodes différentielles** [24]
   - ajout d'une hypothèse de régularité.
   - premier ordre : vitesses inférieures à 1 pixel/intervalle.
   - ordres supérieurs : complexité croissante et sensibilité au bruit.
2. **Algorithmes de mise en correspondance de régions** [3]
   - robustesse.
   - complexité croissante de manière quadratique en fonction de la vitesse maximale recherchée.

Plusieurs formes d'amélioration ont été apportées à cette technique en limitant la zone de recherche, en substituant la recherche dans l'espace par une recherche dans le temps et aussi en limitant la vitesse maximale recherchée. Cependant généralement, elle est sensible au bruit, son calcul est complexe et sa mise en œuvre nécessite, pour les applications en temps réel, un *hardware* spécialisé [64].

B. La différence Simple

Cette approche considère les différences, $D_t(x,y) = I_t(x,y) - I_{t-1}(x,y)$, entre deux ou trois images successives dans la séquence. C'est une technique très simple à mettre en œuvre pour les applications en temps réel et s'adapte bien dans le cas d'environnement dynamique. Cependant, le seuillage de $D_t(x,y)$ nécessite que l'objet soit texturé ou qu'il se déplace plus rapidement autrement, il ne sera pas extrait dans sa totalité en générant des trous à l'intérieur du blob correspondant (l'effet du recouvrement), d'où la nécessité d'un traitement de complètement supplémentaire. Le problème de l'écho dû au dé-couvrement est résolu en incluant une troisième image et en considérant la différence $D_{t+1}(x,y)$. Le masque de changement $B_t(x,y)$ est obtenu par le seuillage :

$$B_t(x,y) = \begin{cases} 1 & \text{si } |D_t(x,y)| > \tau \\ 0 & \text{sinon} \end{cases}$$

Les techniques de détermination automatique du seuil sont nombreuses [63], mais souvent empiriques rendant cette méthode très sensible au bruit et moins robuste aux variations d'illumination. En outre, la "Différence Simple" n'exploite pas les propriétés locales de cohérence du masque de changement.

C. La modélisation de l'arrière-plan

L'image de l'arrière-plan de la scène (*background*) est une représentation quasi-parfaite de l'environnement dans lequel évoluent les objets, elle est utilisée comme image de référence pour extraire les objets en mouvement dans chaque image de la séquence. C'est la méthode la plus utilisée, plus particulièrement dans le cas de fond statique. Cependant, elle est extrêmement sensible aux changements dans une scène dynamique [42].

Dans sa version originale, la détection s'effectue par une simple opération de soustraction entre l'image courante et celle de la partie stationnaire de la scène.

Des résultats intéressants sont obtenus en considérant pour chaque pixel le triplet (min, max, différence max) [23] correspondant respectivement à la valeur minimale et maximale de l'intensité et la différence maximale d'intensité entre les images successives. C'est une technique robuste, cependant une mise à jour régulière de ses paramètres est nécessaire par un simple filtrage adaptatif, ce qui permet au modèle d' " absorber " le bruit dû aux fluctuations de la lumière. Le renforcement de la technique de soustraction du fond par l'information contour [27, 30] améliore la robustesse, la stabilité et la qualité du résultat. Cependant, le temps nécessaire au calcul des gradients directionnels en chaque pixel satisfait moins les exigences strictes du temps réel.

Plusieurs approches établissent un modèle de la scène de référence, pour ensuite procéder par un appariement des images acquises avec ce modèle [16] :

C.1. Les modèles prédictifs

a. Les modèles spatiaux : Différentes techniques considèrent l'information complémentaire du contexte spatial permettant d'aller au-delà des limites de l'information d'intensité et de réduire l'influence des sources de bruit :

1. La méthode Geo-Pixel : Cette méthode compare des blocs nxn pixels et calcule une différence $D(x, y)$ à partir d'une métrique L_{ij} basée sur la moyenne et la variance.
2. Modélisation polynomiale : La métrique utilisée considère les surfaces de tendance d'ordre 0,1 et 2. C'est une modélisation locale sous une forme quadratique de la distribution des niveaux de gris dans le voisinage du pixel (x, y), donnée par la valeur du polynôme :

$$I(x, y) = a_{00} + a_{10}x + a_{01}y + a_{11}xy + a_{20}x^2 + a_{02}y^2$$

   Les blocs qui ne présentent pas de changement ont approximativement les mêmes coefficients de polynômes.
3. Les méthodes dérivatives : Elles permettent de modéliser l'image sous forme de mosaïque de blocs de pixels où la distribution des niveaux de gris est représentée à l'aide des dérivées partielles spatiales de la surface quadratique.
4. Le gradient local d'intensité : Les zones de changement potentielles sont identifées par un gradient $G(x, y) = \min(D(x, y), D(x \pm 1, y \pm 1))$ important. La règle de décision est basée sur la moyenne et la variance de chaque bloc de nxn pixels du masque $G(x, y)$.

b. Les modèles temporels : Dans ce type de modèle, la suite des valeurs des pixels de l'image permet d'exploiter la cohérence temporelle, en considérant l'intensité des pixels comme un processus autorégressif.

C.2. Le modèle d'illumination

Le modèle d'illumination exploite le fait que l'intensité d'un point donné sur un objet est le produit de l'illumination et d'un coefficient d'atténuation. Pour

la détection de changement, le ratio $R(x, y)$ est comparé à un seuil déterminé empiriquement ou statistiquement [49].

$$R(x, y) = \frac{I_1(x, y)}{I_2(x, y)}$$

Ici, on suppose que l'observation d'une variation d'intensité dans l'image traduit nécessairement un mouvement dans la scène. Cela correspond à une hypothèse d'éclairement quasi-constant sur la scène, à défaut le modèle ne serait plus valide [16].

C.3. Le modèle statistique

Cette modèlisation permet de détecter les changements à partir de tests d' hypothèses, l'hypothèse nulle H_0 : il n'y a pas de changement au pixel (x, y) et l'hypothèse complémentaire H_1. La probabilité que la différence $D(x, y)$ soit non nulle, sachant H_0, suit alors une distribution de χ^2. Un seuil de *signification* de test t_α est défini, pour déterminer si un changement à un pixel donné est dû au bruit ou effectivement à un objet en mouvement [70].
Aussi, la probabilité d'observer l'intensité $I_t(x, y)$ à l'instant t et à la position (x, y) peut être exprimée par une somme pondérée de distributions gaussiennes [66, 30].

2.1.3 Détection de l'ombre

La détection et l'élimination des effets de l'ombre sont critiques pour segmenter correctement les objets en mouvement dans une scène puisqu'elles affectent sévèrement le processus de reconnaissance. Plusieurs approches de segmentation permettent de détecter les ombres voire même les pénombres, chacune présente ses propres avantages mais aussi des inconvénients [46]. Les principes utilisés peuvent être regroupés en quatre catégories [55] :

A. Méthodes statistiques non paramétriques

Ces méthodes supposent, par exemple, que la couleur est le produit de l'irradiance et de la reflectance et utilise deux mesures (FIG.2.4) : la distorsion de la chrominance et la distorsion de la brillance. La composante E_i représente la couleur du pixel i dans l'image de référence, I_i est sa couleur dans l'image courante. La différence CD_i exprime la distorsion en chrominance entre E_i et I_i. Cette

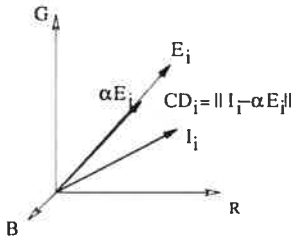


FIG. 2.4 – Distorsion en chrominance.

méthode n'est pas adaptée à un apprentissage en ligne pour la mise à jour du modèle à cause du procédé de normalisation de ses paramètres, nécessaire, pour rendre les comparaisons possibles [25].

B. Méthodes statistiques paramétriques

Ici, on utilise deux sources d'information l'une basée sur l'intensité du pixel, l'autre est d'ordre spatiale. La probabilité que le pixel appartienne à l'ombre est calculée à l'aide d'une matrice de transformation linéaire D qui permet d'estimer les composantes $v' = [R', G', B']$ du pixel sous l'ombre à partir de ces composantes en l'abscence d'ombre. L'information spatiale est exploitée en effectuant une relaxation probabiliste itérative. L'inconvénient majeur de cette méthode est qu'une segmentation manuelle d'un certain nombre d'images de la scène est requise pour collecter les statistiques et la formation de la matrice D. Dans le cas

de scènes constituées de plusieurs surfaces, un zonage est nécessaire, impliquant un plus grand nombre de matrices [44].

C. Méthodes déterministes sans modèle I

Un exemple de ces techniques fait référence à l'espace de couleur HSV en exploitant l'invariance photométrique des composantes *H et S* [7], en supposant que l'ombre ne change pas la teinte *H*, ni la saturation S du fond d'une manière importante mais que, par contre, elle diminue sensiblement son intensité *V* [11].

D. Méthodes déterministes sans modèle II

En exploitant l'information spatiale, la détection de l'ombre se base sur les critères suivants [65] :

- La présence d'une région uniforme plus sombre : En supposant que le rapport entre l'intensité de référence et la valeur d'intensité actuelle est localement constant.
- La présence d'une différence significative en luminance par rapport à l'image de référence.
- Et la présence de contours.

Bien que cette approche traite de la pénombre, les suppositions faites ne sont pas toujours vérifiées.

2.1.4 La validation du masque de changement

Les algorithmes de détection de changements effectuent un post-traitement afin d'éliminer les pixels isolés (bruit) à l'aide d'opérateurs de filtrage.

Pour contraindre le masque de changement résultant selon les exigences et le but de l'application, la taille des blobs est prise en considération. Éventuellement, un lissage des contours des objets détectés, de même qu'un remplissage des trous à l'intérieur de ces blobs peuvent être effectués par des opérateurs morphologiques.

2.1.5 La mise à jour du modèle de l'arrière-plan

Dans le but de maintenir le système fonctionnel le plus longtemps possible, une réactualisation du *background* est nécessaire pour la prise en charge de la dynamique de l'image en particulier les variations d'illumination. L'emploi d'une modélisation adaptative fournit une méthode systématique de mise à jour, en utilisant souvent :

- la moyenne temporelle (plus simple) pour approximer la scène statique courante (FIG.2.5) :

$$B_{k+1}(x,y) = I_k(x,y) + \alpha \cdot |B_k(x,y) - I_k(x,y)|$$

  Cette technique utilise l'effet d'apprentissage et de mémorisation qui consiste à faire contribuer chaque image de la partie statique de la scène par une somme pondérée.

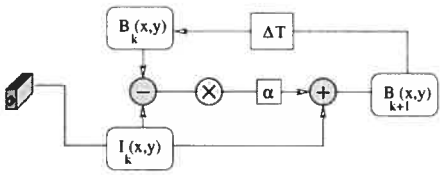


FIG. 2.5 – Algorithme classique de mise à jour.

- la médiane : Cette mesure est plus robuste car elle est moins influencée par les valeurs extrêmes exceptionnelles.
- les filtres de Kalman (linéaire/dynamique) : Les distorsions sont compensées itérativement en supposant qu'elles résultent d'un bruit blanc dont on peut estimer la covariance.

2.1.6 La classification des entités en mouvement

En vidéosurveillance, le but de la classification des objets mobiles est d'extraire correctement les régions correspondant aux humains, à partir des blobs extraits de la précédente phase de segmentation, car l'analyse de leurs activités en est fortement corrélée. Dans les applications où les entités mobiles correspondent exclusivement aux humains, cette étape n'est pas nécessaire.

A. Classification basée sur la forme

La classification d'objets utilise l'information des régions en mouvement tels que le point, la silhouette, le blob ou la boîte englobante. D'autres propriétés caractéristiques, telles que la dispersion exprimée par $périmètre^2/surface$, la $surface$, le ratio $largeur/hauteur$ ..., peuvent être utilisées pour décider si un blob donné représente une personne, un groupe de personne, un véhicule ou simplement une fausse alarme [8, 41].
Les contours des objets mobiles, extraits des images successives, peuvent être comparés avec différents *templates* (appariement de gabarits déformables), ces mesures de similarité sont ensuite injectés dans un réseau de neurones, lequel doit décider si la forme correspond à une forme humaine ou non [68].

B. Classification basée sur le mouvement

Ces méthodes exploitent le caractère cyclique du mouvement humain [13]. Elles font référence au flux résiduel pour analyser la rigidité des entités mobiles et la périodicité de leurs configurations possibles. D'autres sont basées sur les matrices de co-occurrence temporelle, où de meilleures performances sont obtenues en se servant des attributs couleur et vitesse. Les contraintes géométriques

sur les configurations du corps humain : contraintes sur les angles possibles des articulations et sur les vitesses angulaires, constituent également des propriétés caractéristiques visibles pour classer l'objet comme humain ou non-humain. Cependant, les mouvements irrégulièrement cycliques limitent les performances de ce type de classification.

2.2 Suivi temporel

Le suivi d'objets multiples est défini comme l'estimation temporelle des trajectoires relatives à chaque objet. C'est un problème complexe en vidéosurveillance automatique, dans lequel on doit résoudre des événements tels que l'apparition d'objets, la disparition, l'occultation, le regroupement, la séparation et la correspondance, souvent par la mise en correspondance d'un *frame* au suivant de primitives comme les points, segments, bandes, blobs ... etc. Ceci revient à résoudre conjointement deux problèmes : l'*estimation* et l'*association* d'observations données aux objets dont elles sont issues qui, en pratique, n'est pas une tâche facile. Là, encore faut-il que la détection d'objets, résultat de la phase de segmentation, soit correctement faite. Pour ce faire, différents outils sont utilisés :

Filtre de Kalman C'est un estimateur d'états basé sur la distribution Gaussienne [66], cependant cette modélisation n'est pas toujours satisfaisante. Il n'est pas l'estimateur optimal dans le cas de distributions multimodales (*Ex.* : présence d'occlusions, entités suivies semblables ...).

Filtrage particulaire Pour pallier les contraintes de "non linéarité" ou de "non gaussianité ", l'algorithme de condensation consiste à approximer la loi de probabilité conditionnelle des états aux observations par une somme fine pondérée de lois de Dirac centrée en des éléments de R^n appelés particules d'où "Filtrage particulaire" . Cette méthode de propagation de la densité

conditionnelle, basée sur l'échantillonnage et la propagation des échantillons itérativement aux images successives, est robuste [26] mais requiert, relativement, un grand nombre d'exemples.

Le suivi peut être effectué suivant le corps tout entier, ou selon les parties du corps.

2.2.1 Suivi par modèle

Les méthodes de poursuite de cibles à base de modèle cherchent à établir une correspondance entre les images avec le modèle de l'objet à reconnaître.

Modèle de squelette Ces techniques utilisent une approximation du corps humain par une combinaison de segments liés par des jointures. Ce modèle est moins robuste dans les situations où les objets sont sous forme de points épars ou de petite taille.

Modèle de silhouette La projection directe sur le plan est aussi une représentation du corps humain pour effectuer le suivi, l'analyse et la reconnaissance de la marche humaine, en utilisant des patchs planaires ou des *patterns* spatio-temporels. L'avantage des silhouettes réside dans leur extraction qui est plus facile.

Modèle Volumétrique L'inconvénient du modèle 2-D est sa restriction par rapport à l'angle de vue de la caméra. La modélisation 3-D des différentes parties du corps permet d'éviter cet inconvénient et supporte mieux les occlusions, mais requiert trop de paramètres et nécessite beaucoup de temps de calcul. Ces modèles sont plus utilisés en haute et moyenne résolution, dans le cas où les personnes sont assez proches de la camera. La construction et le suivi d'un modèle 3-D est une tâche complexe nécessitant une importante masse de calcul, un modèle hybride 2-D/3-D semble être un compromis plutôt raisonnable [48, 50].

2.2.2 Suivi par région

Le corps humain peut être considéré comme une forme globale combinant un ensemble de blobs représentant les différentes parties du corps. Le principe du suivi par région est d'identifier des régions connectées et de les associer à chaque objet mobile dans chaque image et puis effectuer le suivi au cours du temps en utilisant une mesure d'inter-corrélation. Le critère mettant en correspondance deux régions (blocs), appelée *block-matching*, est généralement la somme en valeur absolue des différences (SAD : Sum of Absolute Difference). La difficulté du suivi par région réside dans deux importantes situations : la présence de zones d'ombre étendue et le cas de scènes encombrées.

2.2.3 Suivi par contours actifs

L'idée est d'avoir une représentation du pourtour de l'objet et de le modeler dynamiquement au cours du temps. Plusieurs travaux basés frontière utilisent un support statistique, d'autres plus récents utilisent les contours actifs géodésiques dans une formulation *level-set* [53] qui s'avère être coûteuse en temps machine, une approche multi-échelle s'apprêterait bien à une détection temps réel et un renforcement de l'approche frontière par un module basé région améliorerait la robustesse [48]. L'avantage du suivi par contour est la diminution de la complexité du calcul. Cependant, la difficulté réside dans la phase d'initialisation, surtout dans le cas d'objets à articulations complexes.

2.2.4 Suivi par attributs

Il est moins facile dans une scène de faire correspondre d'une image à la suivante des attributs de haut niveau telles que des régions que d'établir une correspondance entre des caractéristiques de bas niveau comme les points, dont l'extraction est plus simple. Souvent, dans ces méthodes on considère le centre de gravité du cadre minimum englobant la personne en mouvement pour effectuer le suivi. Dans ce cas, l'intégration de la composante vitesse permet de mieux gérer les occlusions partielles. Le suivi peut être effectué par filtre de Kalman, un outil assez bien développé en vision, en exploitant la pente des segments et la position des points de la silhouette.

2.3 Analyse et reconnaissance d'activités

Cette étape concerne la reconnaissance *globale* de comportements "basse résolution" ou une description plus *locale* de gestes "haute résolution" [48]. Dans un cas comme dans l'autre, le problème de reconnaissance peut être considéré comme un problème de classification. Les techniques générales utilisées sont :

2.3.1 Algorithme de recalage temporel

Basée sur la programmation dynamique, cette technique [2] générale d'appariement permet de déterminer la similarité d'un *pattern* donné avec un motif de référence lorsque les échelles temps ne sont pas parfaitement alignées. Elle a été initialement utilisée en reconnaissance de la parole [45], ensuite appliquée à la reconnaissance de gestes humains [14] et à la reconnaissance de la marche humaine

[2] DTW : Dynamic Time Warping.

puisque cet algorithme s'apprête bien aux variations *naturelles* de la vitesse de marche, au court du recalage, par son procédé de normalisation non-linéaire [12].

2.3.2 Les modèles de Markov cachés

Ce sont des machines à état stochastique non déterministes bien adaptées pour l'analyse de données variables dans le temps. La modélisation par Markov cachés permet de décrire une activité à partir d'une série de mouvements [15] à l'aide de relations de probabilité.

Comme dans le domaine de la reconnaissance de la parole [57], les HMMs [3] ont été appliqués à l'analyse de simples activités humaines dans un corridor (marcher, entrer et sortir d'un bureau) [47], à la reconnaissance de la marche humaine [6], aussi à la reconnaissance de mouvement de Tai-chi [5] et des coups de tennis [74]. Généralement, ils ont des limites dues à l'hypothèse faite sur le processus en supposant toujours qu'il est du premier ordre (l'état courant du modèle ne dépend que de l'état précédent). Une autre limite est due au maximum local qui peut être atteint lors de l'apprentissage qui consiste à déterminer des paramètres à partir d'un algorithme basé sur l'estimation par maximum de vraisemblance (EM), lequel maximise localement.

Cependant, ces modèles peuvent être adaptés pour pouvoir coder les dépendances temporelles sur une échelle temps variable et longue, à l'aide de modèles d'ordre supérieur, en l'occurrence les VLMM (Variable Lenght Markov Models), qui offrent une représentation interne plus efficace du comportement en vision [20]. En décrivant des interrelations entres les HMMs individuels, les HMMs couplés augmentent la performance du système, seulement, l'augmentation du nombre de CHMMs accroîtrait exponentiellement la complexité [48].

[3] HMM : Hidden Markov Model.

Une approche encore plus intéressante puisqu'elle est issue d'un couplage DTW et HMM, considère la reconnaissance comme un processus de fusion de décisions [12].

2.3.3 Les réseaux de neurones artificiels

Les réseaux de neurones ont fait l'objet de beaucoup d'attention ces derniers temps dans un domaine assez similaire de celui de la vidéosurveillance, à savoir la détection des intrusions et de comportements soupçonneux dans les systèmes informatiques [21, 40]. Le grand nombre de données que nécessite la représentation de l'information temporelle n'a pas dissuadé les chercheurs à opter pour l'approche connexionniste dont l'application à la modélisation et l'analyse du comportement humain vise à exploiter son pouvoir de généralisation à partir des données bruitées et/ou incomplètes. Un réseau de type RBF [4] a été développé pour la reconnaissance d'émotions [62]. Aussi, avec une architecture comportant deux couches à apprentissage compétitif et une couche de neurones perméables "leaky neurons" on a pu approximer les fonctions de densité de probabilité des vecteurs flux des trajectoires [32]. Un réseau de type carte de Kohonen a aussi été utilisé pour distinguer dans une scène les comportements anormaux en effectuant un codage de trajectoires à l'aide des vecteurs flux [51]. Semblablement, pour disposer d'une mémoire à plus long terme, une structure hiérarchique de réseaux de Kohonen a été développée pour la détection des situations anormales [52]. Comme, on s'est servi d'une carte de Kohonen pour modéliser les interactions entre objets [34].

Grâce à leur codage interne de l'information temporelle, les réseaux de neurones partiellement récurrents peuvent modéliser correctement des trajectoires spatio-temporelles [56]. Ces types particuliers de réseaux de neurones sont de

[4] Radial Basis Function

puissants outils de prédiction pouvant servir, en particulier, à la prédiction du mouvement [67].

Chapitre 3

Systèmes Connexionnistes

Le but de la reconnaissance de formes (R de F) est de classifier un objet dans la classe la plus proche du point de vue similarité. Généralement, cette classification est un processus automatique de reconnaissance qui implique un apprentissage à partir d'un ensemble de prototypes. L'apprentissage a deux formes : si les classes des exemples sont connues, l'apprentissage est dit supervisé. Dans ce cas, la classification correcte permet une évaluation des performances du système. Si les classes ne sont pas connues à priori, alors on parle d'apprentissage non supervisé qui permet d'identifier les classes elles mêmes. Un processus de R de F est établi en trois phases (FIG.3.1). La première permet l'extraction des données et le prétraitement. La deuxième implique leur représentation : l'observation $\hat{x}$ d'un *pattern* est transformée en un vecteur $\hat{y}$ dont les composantes sont appelées attributs. Le troisième aspect est la classification (prise de décision) des vecteurs attributs. De bons attributs possèdent les quatre caractéristiques suivantes [36] :

Discrimination : les attributs doivent avoir différentes valeurs pour des objets appartenant à des classes différentes ;

Fiabilité : les attributs doivent avoir les mêmes valeurs pour des objets appartenant aux mêmes classes ;

Indépendance : les attributs utilisés doivent être non corrélés les uns aux autres ;

Nombre minimal : la complexité de reconnaissance de formes augmente rapidement avec la cardinalité des vecteurs attributs du système.

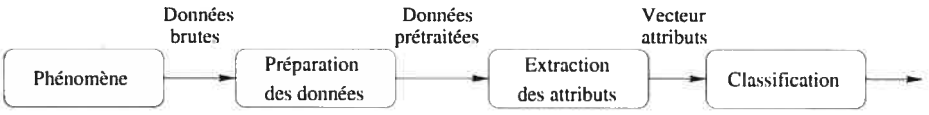


FIG. 3.1 – Processus de reconnaissance de formes.

Les approches les plus connus de reconnaissance de formes sont [28] :

1. *l'appariement de gabarit (template matching) ;*
2. *les modèles statistiques ;*
3. *les modèles syntaxiques ou structurels (grammaires et primitives) ;*
4. *les réseaux de neurones.*

Ces derniers temps, l'utilisation éprouvée des réseaux de neurones artificiels (RNA) et la puissance des plates-formes disponibles ont permis de tester leurs véritables performances en les comparant aux autres modèles stochastiques de reconnaissance de formes. Ils se sont avérés très efficaces en classification et notamment en prédiction [1]. Ces techniques présentent des propriétés plus intéressantes en matière de classification [10] :

- Les RNAs peuvent apprendre : étant donnée un ensemble assez large de données d'apprentissage, les paramètres du réseau peuvent alors être calculés pour minimiser un critère d'erreur donnée.
- Ils sont capables de générer n'importe quel type de fonction non linéaire.
- En général, aucune hypothèse n'est faite sur les distributions statistiques des données.
- Les RNAs sont des structures régulières massivement parallèles, pouvant offrir de hautes performances lorsqu'ils sont utilisés dans des architectures parallèles.

3.1 Propriétés des réseaux de neurones

Ce sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur calcule une sortie unique sur la base des informations qu'il reçoit. Réellement, le neurone artificiel ne peut pas simuler le neurone biologique mais la motivation première demeure. Ceci est dû à leur faculté à apprendre et à généraliser pour de nouveaux cas.
Les RNAs sont particularisés par trois aspects :

1. La topologie d'interconnexion ;
2. Les propriétés des unités ;
3. La lois d'apprentissage.

3.1.1 Topologie d'interconnexion

Selon les règles de connexions, on distingue deux modèles de structure de réseaux :

- Les réseaux à couches :
    1. Les réseaux non récurrents
    2. Les réseaux récurrents
- Les réseaux sans systèmes de couches

A. Les réseaux à couches

Une structuration en couches tend à rendre compte des différents traitements que l'on peut effectuer en cascade sur un ensemble d'informations, proposées sur une couche extrême, appelée couche d'entrée ; elles sont ensuite traitées par un nombre variable de couches intermédiaires ou couches cachées. Le résultat apparaît sur l'autre couche extrême, la couche de sortie. Deux types d'architecture caractérisent les réseaux à couches à savoir : les réseaux non récurrents (feedfor-

ward networks) et les réseaux récurrents (feedback networks).

A.1. Les réseaux non récurrents : La propagation de l'information dans ces réseaux est vers l'avant, les neurones d'une couche donnée reçoivent des informations uniquement de la couche antérieure de façon unidirectionnelle (connexion directe) et chaque neurone émet des informations vers les autres neurones de la couche postérieure (connexion totale), il n'existe pas d'échange latéral d'information entre neurones d'une même couche. Le perceptron multicouche, le réseau à fonctions de base radiale (RBFN : Radial basis function network), le réseau CMAC (Cerebellar model arithmetic computer) s'inscrivent dans ce type d'architecture.

A.2. Les réseaux récurrents : L'architecture de ces réseaux utilise des effets de retour (FIG.3.2). Chaque neurone peut recevoir comme entrée les sorties des autres neurones, et également de lui-même. Le réseau multicouche récurrent (Recurrent Backpropagation Network) est un exemple des réseaux récurrents. Les ART (Adaptive Resonance Theory) définis par Grossberg [22] font aussi partie de ce type de réseaux. Ils sont organisés en deux couches, chacune est totalement connectée par des arcs bidirectionnels.

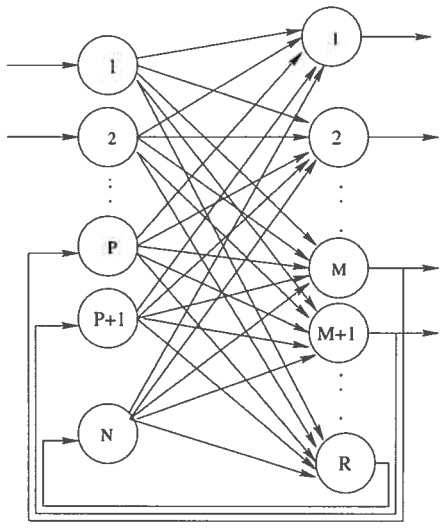


FIG. 3.2 – Réseau multicouche récurrent

B. Les réseaux sans systèmes de couches

Ces réseaux désignent généralement des réseaux totalement interconnectés, où chaque neurone peut être connecté à n'importe quel autre neurone du réseau et également avec lui-même, les connexions récurrentes sont également typiques de ce modèle. Le modèle de Hopfield et les cartes topologiques de Kohonen font partie de ce type de réseaux.

3.1.2 Propriétés des unités

Au niveau des unités, les principales différences résident dans le type d'activation employée (linéaire, sigmoïde exponentielle asymétrique, sigmoïde tangentielle symétrique, gaussienne) et dans la fonctionnalité de ces unités, on peut distinguer deux types majeurs de réseaux :

Réseaux à prototypes : Ce type de réseaux utilise des unités qui servent à représenter des motifs d'exemples appris : les unités ont une représentation interne regroupant les caractéristiques typiques d'un ensemble d'exemples. Les réseaux à prototypes ont normalement un apprentissage non supervisé, ce qui permet de faire appel à une sorte d'inspiration biologique, mais il peuvent aussi avoir un apprentissage supervisé.

Réseaux de type Perceptron : Il s'agit d'un des modèles d'unités les plus utilisés actuellement. Il est à l'origine de plusieurs autres réseaux de neurones avec apprentissage supervisé par correction d'erreur. Le modèle du Percepton Multicouche est devenu très connu, tout en étant associé à la règle d'apprentissage de la rétro-propagation.

3.1.3 Lois d'apprentissage

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. C'est une étape d'adaptation qui fait appel à des exemples caractérisant ce comportement. Les règles d'apprentissage fixent des poids initiaux et indiquent ensuite leur stratégie d'adaptation durant la phase d'apprentissage, jusqu'à atteindre une certaine performance, le facteur le plus important car elle définit la capacité de généralisation du réseau. Cette stratégie peut être mise en œuvre par :

Apprentissage supervisé : Les poids sont ajustés sur la base de comparaison entre les réponses réelles du réseau et les réponses attendues (correctes). De cette façon on apprend au réseau le comportement désiré. La procédure la plus utilisée dans ce type d'apprentissage est la rétro-propagation de l'erreur qui, comme son nom l'indique, consiste à propager l'erreur commise en sortie vers les couches antérieures, pour modifier les poids synaptiques.

Apprentissage non supervisé : Ce mode d'apprentissage est utilisé lorsque la seule information disponible se trouve dans la corrélation entre les données. Le réseau devra donc créer des groupes à partir de ces corrélations. Contrairement au précédent, ce type d'apprentissage repose sur un *critère interne* de conformité du comportement du réseau par rapport à des spécifications générales et non sur des observations externes.

Les règles d'apprentissage les plus utilisées sont :

- Les méthodes de correction d'erreur, telle que la descente du gradient sur une surface d'erreur par exemple le neurone adaptatif (l'Adaline) ou la rétro-propagation.
- Les méthodes d'apprentissage par renforcement telle que AHC (Adaptive Heuristic Critic), ARC (Association Reinforcement Comparaison).
- Les méthodes d'apprentissage par compétition ou par auto-organisation comme les cartes auto-organisatrices de Kohonen, ART1 (Adaptive Resonance Theory).

- Les méthodes d'apprentissage par création de noyaux par exemple le RBF (Radial Basis Function), ART1, ART2.
- Les méthodes d'apprentissage basées sur des mémoires associatives (auto-associatives ou hétéro-associatives) comme le modèle de Hopfield, BAM (Discrete Bidirectional association Memory).
- Les méthodes d'apprentissage temporel des réseaux récurrents telles que : SRN (Simple Recurrent Network), BPTT (Back Propagation Through Time), RTRL (Real Time Recurrent Learning).

D'autres variantes consistent à améliorer le fonctionnement et à rendre dynamique la structure du réseau, comme la RPROP (Resilient Propagation), QuickProp et le gradient conjugué (Descente de gradient du deuxième ordre), Cascor (Cascade correlation) et les méthodes ontogéniques d'apprentissage qui sont basées sur l'évolution de l'architecture du réseau pendant l'apprentissage. Actuellement, aucune architecture ne s'est distinguée par rapport à une autre. L'architecture non récurrente et l'algorithme d'apprentissage de la rétro-propagation associé sont largement utilisés car ils sont les plus connus.

Une analyse d'une scène en vision informatique implique un traitement de séquences spatio-temporelles. Cependant dans l'analyse des systèmes temporels par une approche connexionniste, l'aspect critique réside dans la façon dont sont présentées les entrées antérieures "l'effet mémoire" et la manière dont cet historique affecte les réponses des entrées actuelles.

3.2 Les réseaux de neurones incorporant la mémoire

D'un côté les exigences des systèmes dynamiques, d'un autre, les limitations des RNAs classiques à introduire dans leur comportement une dimension temporelle ont permis de développer de nouvelles architectures de réseaux de neurones.

Une approche commune consiste à mémoriser des entrées du passé récent pour les présenter, en même temps, avec les entrées courantes [17]. Généralement, l'effet mémoire est intégré sous forme de :

1. **Architectures bouclées**
   - Mémoire associative (Hopfield, réseaux à attracteurs, ...).
   - Les réseaux temporels récurrents obtenus par des extensions des réseaux non récurrents à l'aide de boucles de retour (FIG.3.2, page 27).
2. **Architectures classiques**
   - Les réseaux non récurrents avec fenêtre glissante dans le temps qui permet, pour chaque attribut, de présenter ses n dernières valeurs comme entrée du réseau.
   - Codage temporel des entrées : En créant, par exemple, des entrées de la forme u, du/dt, d^2u/dt^2 ...

3.2.1 Les réseaux de Hopfield

Ces réseaux tentent de stocker un ensemble spécifique de points d'équilibres, appelés états permanents ou stables au sens de Lyapunov correspondants aux exemples stockés. Le comportement du réseau peut être expliqué en terme d'énergie. Un état permanent est atteint par une diminution de cette énergie, correspondant au processus de rappel de la mémoire. Mise à part l'initialisation des états des neurones, il n'y a pas d'entrées externes. En mode fonctionnement, les sorties sont réinjectés en entrées. Ces réseaux sont utilisés comme mémoire associative. Leur inconvénient est dû au minimum local qui peut être facilement atteint lors de la relaxation.

3.2.2 Les réseaux d'Elman

Ils contiennent des connexions récurrentes, leurs permettant de détecter et de générer des motifs temporels. Les réseaux d'Elman sont capables d'approximer n'importe quelle entrée/sortie avec un nombre fini de discontinuités grâce à deux couches sigmoïde/linéaire. Certaines activations en provenance des unités cachées sont injectées à l'entrée via un ensemble supplémentaire appelé unités contextuelles permettant la prise en charge du phénomène *mémoire* [17]. Un autre type de réseaux, appelé réseaux de Jordan, peuvent apprendre eux même l'influence des entrées qui précèdent un instant donné.
À la différence des réseaux d'Elman, les unités supplémentaires d'entrée dans les réseaux de Jordan, appelés unités d'états reçoivent une copie des unités de sortie.

3.2.3 Les réseaux de neurones à codage temporel

Ces architectures font intervenir le contexte temporel en opérant en deux phases (FIG.3.3) :

- Codage temporel
- Architecture non récurrente

Parmi les techniques de codage temporel proposées, on trouve celles qui utilisent les lignes à retard à longueur variable (Tapped Delay Line), dont la sortie est un vecteur à N dimensions, comprenant le signal d'entrée à l'instant t, le signal d'entrée à l'instant *t-1* ...etc.

Une attention particulière a été apportée aux réseaux dédiés au traitement de séquences vidéo où les informations du premier ordre (position et vitesse de déplacement) sont considérées comme des propriétés caractéristiques pour le codage spatio-temporel des trajectoires [33]. Pour la détection et le stockage des

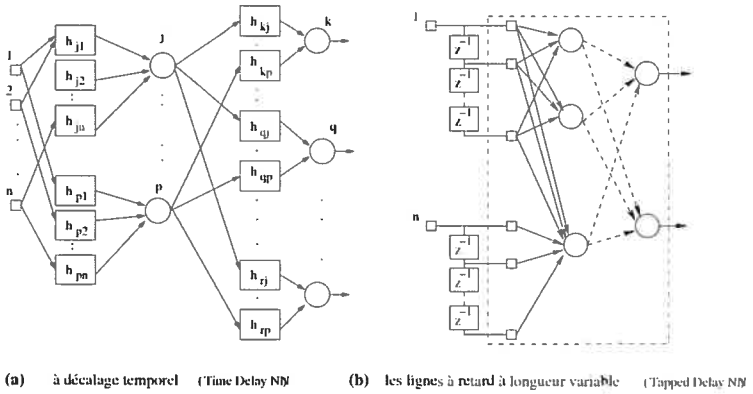


FIG. 3.3 – Réseaux de neurones à codage temporel

corrélations entre entrées successives dans le temps, ces techniques utilisent le vecteur flux $[x, y, dx, dy]$, ou son vecteur augmenté F :

$$F = [x, y, s(x), s(y), s(dx), s(dy), s(|d^2x|), s(|d^2y|)]$$

Les différences du premier et second ordre sont données par $dx = x_t - x_{t-1}$ et $d^2x = x_t - 2 \cdot x_{t-1} + x_{t-2}$. La fonction de lissage $s(x_t) = (\nu)(x_{t-1}) + (1-\nu)(x_t)$ définit une fenêtre mobile moyenne. Le vecteur F est ensuite utilisé comme entrée d'un réseau de type Kohonen.

Une évaluation comparative de certains types de réseaux de neurones appliqués à la vidéosurveillance démontre une performance réelle des cartes de Kohonen pour une moindre complexité [2].

3.3 La carte auto-organisatice de Kohonen

La carte auto-organisatrice de Kohonen SOM [1] convient mieux à notre cas d'étude dans le sens où dans l'impossibilité de recenser tous les comportements anormaux, le système doit chercher des cohérences à partir des corrélations présentes

[1] Self Organizing Map

dans les données issues seulement de comportements normaux, où un phénomène telle qu'une séquence d'observation de mouvements est transformé en une séquence de vecteur "attributs". Ainsi, on identifie comme comportement *anormal* tout profil présentant une déviation plus ou moins significative par rapport au plus proche profil *normal.*

3.3.1 Principe

Turing en 1952 stipule que : *"un ordre global peut être produit par des interactions locales"*. C'est sur ce concept que Kohonen introduisit l'un des meilleurs

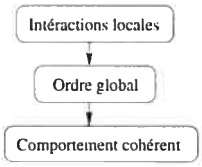


FIG. 3.4 – Principe du modèle de Kohonen

RNAs connus dans la catégorie des non supervisés, qui se base sur les principes suivants :

Principe 1 : Une amélioration des poids synaptiques à tendance à s'auto-amplifier.

Principe 2 : La limitation des ressources du réseau entraîne une compétition entre les synapses.

Principe 3 : Les modifications synaptiques doivent coopérer.

Ce réseau appartient à la classe des algorithmes de codage de vecteurs. Dans ce type d'algorithmes, il s'agit de placer un nombre fixe de vecteurs, appelés prototypes dans l'espace d'entrée, lequel est souvent un espace réel R^d de grande dimension. L'espace d'entrée est représenté par un ensemble d'apprentissage $(x_1, \ldots, x_n)$ $\backslash x_i \in R^d$. La dimension 'd' dépend du problème traité. Chaque prototype correspond à une partie de l'espace d'entrée : l'ensemble des points les plus proches

de ce prototype, en terme de distance. Un tel ensemble est convexe et ses bordures sont définies par les intersections d'hyperplans. Ce qui produit dans l'espace une tessélation de Vonoroï. Le critère général est de placer, pour chaque ensemble de Voronoï, des prototypes représentatifs d'une façon à minimiser les distances moyennes entre ces prototypes et les points d'entrée appartenant à cet ensemble (EQU. 3.1). Ceci est effectué par des algorithmes d'apprentissage non supervisés et dirigés par les données.

$$(E)^2 = \sum (w_i - x_i)^2 \tag{3.1}$$

Chaque prototype représente le vecteur poids d'un neurone. Les neurones sont organisés en grille à une, deux ou plusieurs dimensions tel que chaque neurone possède un voisinage (FIG.3.5). Le but de l'apprentissage ne consiste pas seulement à rechercher les *patterns* qui représentent au mieux l'ensemble des données d'entrée au sens des moindres carrées, mais aussi à trouver, en même temps, un *mapping* topologique entre l'espace d'entrée et la grille de neurones.

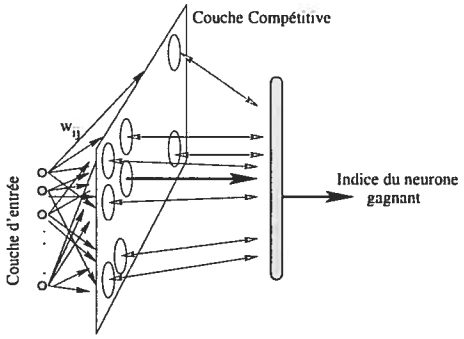


FIG. 3.5 – Carte auto-organisatrice de Kohonen

Dans sa version topologique, le neurone gagnant i vérifiant l'équation (3.2) incite ses voisins à modifier leurs poids dans le même sens selon l'équation (3.3). Ce qui permet à la couche entière de s'auto-organiser de manière topologique. L'unité i ayant le vecteur poids w_i est appelée *the best matching unit* (MBU) du vecteur x, et l'indice $i = i(x)$ est considéré comme la réponse de la SOM. Pour

une unité i fixe, l'ensemble de points x satisfaisant l'équation (3.2) représente l'ensemble de Vonoroï associé à cette unité.

$$\|x - w_i\| = \min_{j=1...l} \|x - w_j\| \tag{3.2}$$

$$w_j = w_j + \gamma \cdot h_r(x - w_j) \tag{3.3}$$

γ représente le facteur gain et h_r est une fonction de la distance r, $(r = \|MBU - j\|)$

3.3.2 Propriété topologique

Si deux neurones i et j sont voisins alors leurs ensembles de Vonoroï dans l'espace d'entrée possèdent des bordures communes, ceci est valide si la propriété du *mapping* topologique peut être vérifiée pour toutes les unités, cependant, cela dépend de la cardinalité de l'espace d'entrée et de la grille de neurones : car au sens mathématique, il n'est pas possible de préserver la topologie entre deux espaces de cardinalités différentes, une couche de neurones à deux dimensions ne peut que, localement, suivre deux dimensions de l'espace multidimensionnel d'entrée. En pratique, l'espace d'entrée est de grande dimension mais les nuages de données $(x_1, \ldots, x_n)$ utilisées pour l'apprentissage se trouvent concentrés sur des dimensions inférieures que la carte peut suivre au moins approximativement [37].

Le fait que le *mapping* possède une propriété topologique, a l'avantage de le rendre plus tolérant aux erreurs : Si $\tilde{x}$ est un vecteur obtenu par une perturbation sur un vecteur donné x, alors la réponse de la carte $i(\tilde{x})$ ne peut être qu'au voisinage de $i(x)$. Cette propriété est essentielle pour une grande performance (faible erreur). Dans une architecture SOM, le rôle de la couche de sortie *Tout Au Vainqueur* est de comparer les réponses de la couche compétitive (les distances $\|x - w_i\|$) et de repérer l'indice du MBU. Une fonction d'activation peut être définie pour chaque neurone i telle que sa sortie soit une fonction monotonique

décroissante de $||x - w_i||$. La forme générale de l'algorithme d'auto-organisation est donnée comme suit :

1. *Choisir aléatoirement des valeurs initiales pour les vecteurs poids.*
2. *Répéter les étapes 3 et 4 jusqu'à convergence de l'algorithme.*
3. *Choisir aléatoirement un exemple x à partir des exemples d'entrée et chercher son MBU selon l'équation (3.2)*
4. *Ajuster les vecteurs poids de toutes les unités selon l'équation (3.3)*

La fonction de voisinage h_r et le gain γ doivent décroître lentement en fonction du temps pour assurer la convergence de l'algorithme, dont le rôle est de construire des vecteurs poids optimaux dans l'espace des attributs, ce qui correspond à une forme de compression.

Dans le cas de masses importantes de données, cette forme de compression peut être utilisée pour une visualisation. Pour une meilleure représentation graphique, des systèmes hiérarchiques peuvent être construits dans lesquels les sorties des cartes de niveau inférieur sont aussi utilisées comme entrées de celles des niveaux subséquents et ainsi de suite [71].

Chapitre 4

Méthodologie

4.1 Estimation de l'arrière-plan

La détection des *objets en mouvement* (MVO) [1] est basée sur la soustraction du fond, une technique très efficace lorsqu'elle est adaptée aux scènes dynamiques. L'idée de base est de comparer les données acquises en mode fonctionnement à un modèle préenregistré de la scène pour extraire les régions d'intérêt. Dans sa forme la plus générale, le modèle de segmentation de l'avant-plan comprend les étapes suivantes :

- La modélisation de l'arrière-scène : revient à construire une image référence représentant la partie stationnaire de la scène ;
- La soustraction : permet de classifier un pixel donné comme faisant partie d'un MVO ou du *fond* ;
- La mise à jour de l'arrière-plan.

[1] **Moving Visual Object** : On désignera par **objet** en mouvement, le sujet de la surveillance : **un humain**.

4.1.1 Le modèle du fond

La délimitation de l'ombre est un enjeu primordial d'une bonne détection d'objets en mouvement. Vu les problèmes que peut causer les pixels de l'ombre désormais classés comme avant-plan, conduisant à des formes illusoires, pire encore entrainant des fausses connectivités entre blobs complètement indépendants. Exploitant l'information **brillance** et celle de la **chrominance**, l'approche que nous avons adoptée pour contrer ces difficultés, vise à tirer parti d'une caractéristique assez singulière de l'ombre qui est à l'origine de plusieurs travaux [25, 11] dont le principe repose sur le fait que l'ombre couvre un pixel en diminuant sensiblement son intensité, par contre en gardant pratiquement invariante sa chromacité, en affectant les composantes RGB d'une façon proportionnée [39], avec une quasi-préservation de couleur dominante produisant l'effet de semi-transparence.

A. La brillance

Soit $< I_R(x), I_G(x), I_B(x) >$ la représentation RGB du pixel x de l'image courante et $< BCK_R(x), BCK_G(x), BCK_B(x) >$ celle de son correspondant dans l'image référence. On définit alors la distorsion en brillance $\delta Br(x)$ du vecteur $I(x)$ relativement au vecteur $BCK(x)$ comme la différence entre $BCK(x)$ et la projection $I'(x)$ de $I(x)$ sur le support de $BCK(x)$ (FIG.4.1.a). Formellement on écrit :

$$\delta Br(x) = \|BCK(x)\| - \frac{I(x) \cdot BCK(x)}{\|BCK(x)\|}$$

Trois cas se présentent : $\delta Br(x)$ est nulle si aucun changement de brillance dans la scène n'est observé, elle est inférieure à zéro pour les régions devenues plus brillantes et possède une valeur positive pour les pixels assombris, on parlera dans ce cas, plutôt, du niveau d'assombrissement.

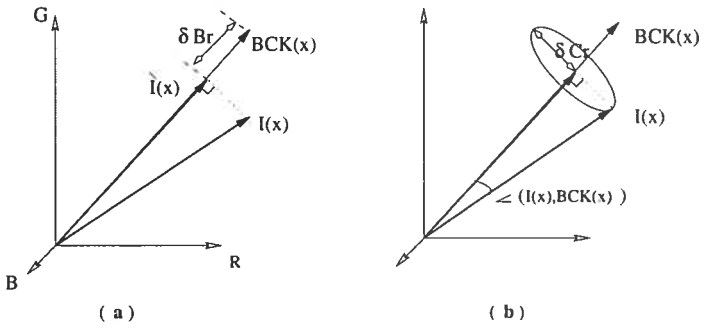


FIG. 4.1 – Distorsion (**a**) de brillance - (**b**) chromatique

B. La chrominance

Un modèle considérant seulement la brillance est incomplet car il ne sera pas capable de détecter des pixels dont la couleur a changé en restant dans le plan d'équibrillance exprimé par $|\delta Br(x)| \approx \epsilon$ (FIG.4.2). Le mode brillance,

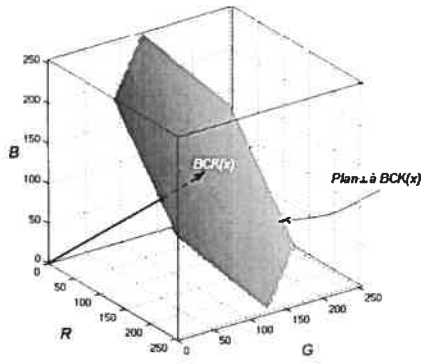


FIG. 4.2 – Plan d'équibrillance $BCK_{\perp}(x)$ associé à $BCK(x)$ dans le cube RGB.

tel que défini ci-avant, opère sur le support du vecteur $BCK(x)$ (FIG.4.1.a). L'établissement d'un modèle de chrominance permet la prise en charge de la couleur en opérant dans le plan $BCK_{\perp}(x)$. On définit alors une distorsion chromatique $\delta Cr(x)$ du vecteur $I(x)$ par rapport au vecteur $BCK(x)$ comme l'angle $\angle\big(I(x), BCK(x)\big)$ (FIG.4.1.b) :

$$\delta Cr(x) = \arccos\left(\frac{I(x) \cdot BCK(x)}{\|I(x)\| \times \|BCK(x)\|}\right)$$

4.1.2 La soustraction du fond

Un pixel x est classé appartenant à l'avant-plan si :

$$\delta Br(x) < 0 \quad \text{ou} \quad \delta Br(x) > \tau_0 \tag{4.1}$$

sinon si sa distorsion chromatique est significative :

$$|\delta Cr(x)| > \theta$$

Les intensités $I(x)$ vérifiant $0 < \delta Br(x) \leq \tau_0$ et $|\delta Cr(x)| \leq \theta$ définissent le volume d'ombre Γ (FIG.4.3).

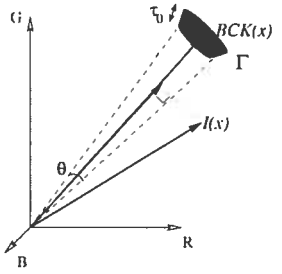


FIG. 4.3 – Le volume d'ombre Γ.

En posant :

$$\tau_0 = (1 - \tau_1)\|BCK(x)\| \tag{4.2}$$

on garantit un seuillage semi dynamique (compression proportionnelle du seuil) selon la norme de $BCK(x)$, en effet la perception des différences d'intensités trop saturées n'est pas la même que pour celles qui le sont moins. L'équation (4.1) devient :

$$\|BCK(x)\|^2 < I(x) \cdot BCK(x) \quad \text{ou} \quad \tau_1\|BCK(x)\|^2 > I(x) \cdot BCK(x)$$

La distribution des pixels les plus saturés a tendance à avoir une plus faible variance, en outre l'ombre n'affecte pas les zones éclairées de la même manière

que celles qui sont plus sombres, dont la chromaticité est moins stable à cause de leur représentation RGB qui est plus proche de l'origine. En conséquence, un clivage *Chromatique* vs. *Achromatique* de la scène, permet de considérer des seuils (τ_1, θ) plus appropriés, à ce dernier, on définit une couleur de l'arrière-plan comme achromatique si : $\max_{c \in \{R,G,B\}} \left(BCK_c(x)\right) < 50$.

Les résultats montrés dans (FIG. 5.1 à 5.5 :page 61) ont été obtenus en fixant les seuils (τ_1, θ) à (0.75, 0.1 rad) pour les pixels achromatiques du fond et à (0.80, 0.04 rad) pour ceux de la partition complémentaire (chromatique). Pour la distorsion de brillance la sensibilité est de 0.15, la valeur du seuil τ_0 s'ajuste par rapport à la norme de BCK (EQU. 4.2). Pour la distorsion chromatique, la sensibilité est de l'ordre de 0.05 rad dans le cas de pixels achromatiques et de 0.02 rad pour les pixels chromatiques (en supposant que les pixels proches de l'origine dans le système RGB, correspondant à des pixels à faible intensité V dans le système HSV, sont moins stables quant à leur teinte).

Les applications en temps réel exigent des algorithmes simples, rapides et fiables. Pour pouvoir satisfaire ces exigences temporelles strictes, on augmente la fréquence de détection, en ne soumettant au modèle que les pixels qui constituent des candidats potentiels, ceci en ne retenant que ceux vérifiant :

$$|I_c(x) - BCK_c(x)| > k \cdot \sigma_c(x) \qquad \text{avec } c \in \{R, G, B\}$$

Les seuils $\sigma_c(x)$ sont déterminés automatiquement et représentent les différences maximales en valeur absolue entre deux images consécutives [23]. En prenant k=2, on se fixe 75% des valeurs dans l'intervalle $[BCK_c(x) - \sigma_c(x), BCK_c(x) + \sigma_c(x)]$, en supposant que $\sigma_c(x)$ est un estimé fiable de l'écart-type [2].

[2] Théorème de Chebyshev.

Pseudo code de soustraction

```
//soit I_c(x) le vecteur intensité du pixel x de l'image courante
// et BCK_c(x) son correspondant dans l'image référence.
// tester si x représente un candidat potentiel.
si (|I_R(x) − BCK_R(x)| > k σ_R(x) ou |I_G(x) − BCK_G(x)| > k σ_G(x)
        ou |I_B(x) − BCK_B(x)| > k σ_B(x))
//tester le niveau d'assombrissement ([3]).
   τ = I(x)·B(x) / ||BCK(x)||²
   si (τ_1 < τ et τ < 1.0) alors
      //x est relativement assombri par rapport à BCK(x)
      //vérifier sa distorsion chromatique.
      si (|δCr(x)| > θ) alors //x a changé de teinte.
         x ∈ MVO
      sinon
         x ∈ ombre
      fsi
   sinon
      x ∈ MVO
   fsi
fsi
```

4.1.3 La mise à jour du fond

Pour garder le système fonctionnel le plus longtemps possible dans le cas de scènes dynamiques, une "maintenance" périodique du *background BCK* est nécessaire. Ce qui est obtenu par une mise à jour d'un *background* "parallèle" qu'on note BP, selon une approximation récursive de la moyenne pondérée des intensités $I_c(x)$:

$$BP_c^{t+\delta t}(x) = \alpha \cdot BP_c^t(x) + (1-\alpha) \cdot I_c^{t+\delta t}(x)$$

Cette expression permet d'estimer une distribution gaussienne non stationnaire de l'intensité. Le facteur $\alpha \in [0,1]$ est initialisé empiriquement en tenant compte de la fréquence d'acquisition de la caméra, en effet cette valeur contrôle le taux

[3] Deux couples de seuils empiriques (τ_1, θ) sont considérés, selon la pré-segmentation de $BCK(x)$ (Chromatique ou Achromatique).

d'adaptation. Ainsi, $BP_c(x)$ finale représente une moyenne avec pondération exponentielle des valeurs prises par le pixel x durant la période de mise à jour Δt. À $t = \Delta t$, l'image référence BCK est réinitialisée par BP où seuls les pixels non visités par des MVOs valides, seront considérés, ceci en gardant leur trace à partir du processus de poursuite de cibles.

Grâce au *background* "parallèle", la mise à jour sélective (EQU. 4.3) permet de ne pas compromettre le modèle du background "original".

$$BCK^{t+\Delta t}(x) = \begin{cases} BCK^{t}(x) & \text{si } \exists t' \in [t, t+\Delta t] \text{tq } x \in \text{O et O} \in \{MVOs\}^{t'} \\ BP^{t+\Delta t}(x) & \text{sinon} \end{cases} \tag{4.3}$$

La soustraction de l'arrière plan seule ne permet pas un débruitage de l'image. Ceci requiert une phase supplémentaire mettant en œuvre, souvent, des filtres morphologiques qui se prêtent bien au cas pratique de vidéosurveillance, mais généralement, en faisant du système une application dépendante de la scène. Par ailleurs, les auteurs éprouvent de grandes difficultés à trouver la bonne combinaison des opérations morphologiques à appliquer [4]. Dans cette optique, vu la robustesse de notre modèle de détection (FIG. 5.1 : page 61 à 5.5), seul un filtre médian (3×3) est nécessaire pour éliminer les pixels isolés et les effets de bordure dus à la distribution aléatoire des échantillons durant le processus d'acquisition (FIG. 4.4). Les régions d'intérêt retenues sont ensuite labellisées par un algorithme rapide d'étiquetage en composantes connexes et les petits blobs seront ignorés.

4.2 Étiquetage en composantes connexes

L'image résultat de l'étiquetage en composantes connexes, assigne à chaque pixel un identificateur : l'identificateur du blob auquel il appartient. Les blobs détectés permettent de déterminer les objets mobiles. La façon la plus classique

FIG. 4.4 – Effet "Jitter" de la caméra en partie dû à des vibrations provoquées par une bouche d'aération tout près.

est d'effectuer pour chaque pixel un *Flood Fill* [54] utilisant des appels récursifs qui présentent un risque de débordement de la pile. Les meilleursalgorithmes d'étiquetage sont ceux qui procèdent avec une ou deux lignes à la fois, et au maximum en deux passes. Profitant des optimisations du compilateur, leur complexité est en O(n) où n est le nombre de pixels de l'image [59].

4.2.1 Algorithme de gestion de partitions

Pour un étiquetage efficace on crée une représentation interne de la connexité de l'image en utilisant un algorithme de gestion de partitions (*Union-Find Algorithm*) [9]. On appelle partition d'un ensemble E une famille de sous-ensembles non vides $E_1, \ldots, E_k$ appelés *classes* qui soit telle que :

- deux classes quelconque E_r et E_s soient disjointes ;
- l'union de toutes les classes soit l'ensemble E tout entier.

Étant donné une partition de l'ensemble d'entiers $\{0, \ldots, n-1\}$ (n représente le nombre total de pixels contenus dans le masque de changement), on veut résoudre les deux problèmes suivants :

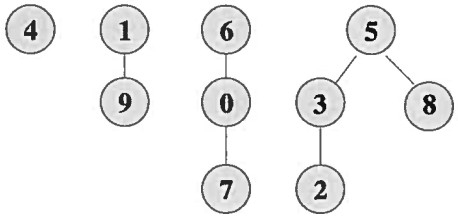
- trouver la classe d'un élément (find) ;

- faire l'union de deux classes (union).

En général, on commence par une partition où chaque classe est réduite à un singleton, qu'on traite à l'aide d'une suite de requêtes de l'une des deux formes ci-dessus. Pour la représentation d'une partition, deux solutions peuvent être envisageables :

- La première consiste à représenter la partition par un tableau *classe* : Chaque classe est identifiée par un entier, et $classe[x]$ contient le numéro de la classe de l'élément x. Trouver la classe d'un élément se fait en temps constant, mais fusionner deux classes prend un temps O(n), puisqu'il faut parcourir tout le tableau pour repérer les éléments dont il faut changer la classe.
- Une deuxième solution consiste à choisir un représentant dans chaque classe : Fusionner deux classes revient alors à changer de représentant pour les éléments de la classe fusionnée.

Il paraît avantageux de représenter la partition par une forêt. Chaque classe de la partition constitue un arbre de cette forêt. La racine de l'arbre est le représentant de sa classe. La figure (4.5.a) montre la forêt associée à une partition.
Une forêt est représentée par un tableau d'entiers *parent* (FIG.4.5.b). Chaque noeud est représenté par un entier, et l'entier $parent[x]$ est le parent du noeud x. Une racine r n'a pas de parent. On convient que, dans ce cas, $parent[r] = r$. Chercher le représentant de la classe contenant un élément donné revient à trouver la racine de l'arbre contenant un noeud donné. L'union de deux arbres se réalise en ajoutant la racine de l'un des deux arbres comme nouveau fils à la racine de l'autre. La complexité est O(h) où h est la hauteur de l'arbre (la plus grande des hauteurs des deux arbres). En fait, lors de l'union de deux arbres, on peut améliorer l'efficacité de l'algorithme par la règle suivante : la racine de l'arbre de moindre taille devient fils de la racine de l'arbre de plus grande taille. Une deuxième stratégie, appliquée cette fois-ci lors de la méthode *Find* présente une amélioration plus marquée, en se basant sur la règle de compression de che-



(a) Forêt associée à la partition {{4},{1,9},{0,6,7},{2,3,5,8}}

x	0	1	2	3	4	5	6	7	8	9
parent [x]	6	1	3	5	4	5	6	0	5	1

(b) Tableau rattaché à la forêt (a)

FIG. 4.5 – Représentation d'une partition.

mins suivante : après être remonté du noeud x à sa racine r, on refait le parcours en faisant de chaque noeud rencontré un fils de r.

4.2.2 L'étiquetage

L'algorithme opère ligne par ligne de gauche à droite. Pour chaque pixel, on vérifie si son voisin gauche possède la même valeur alors on est dans le même blob et donc le pixel courant est directement étiqueté. Si le pixel du haut a la même valeur que celui du gauche mais pas le même identificateur de blob, alors les deux pixels appartiennent à la même région et donc une relation d'équivalence doit être mémorisée, pour simplifier leur fusion. Si le pixel du gauche ne possède pas la même valeur que celle du pixel du haut, alors un blob est créé et un nouvel identificateur lui est assigné. L'algorithme continue ainsi, la création de nouveaux blobs et la fusion des blobs équivalents (FIG.4.6). La supériorité de l'*Union-Find data structure* est de gérer et de garder efficacement la trace des relations d'équivalence entre les blobs, en sauvegardant les labels qui correspondent au même blob dans un arbre qui rend la tâche de résolution des équivalences plus rapide et plus facile.

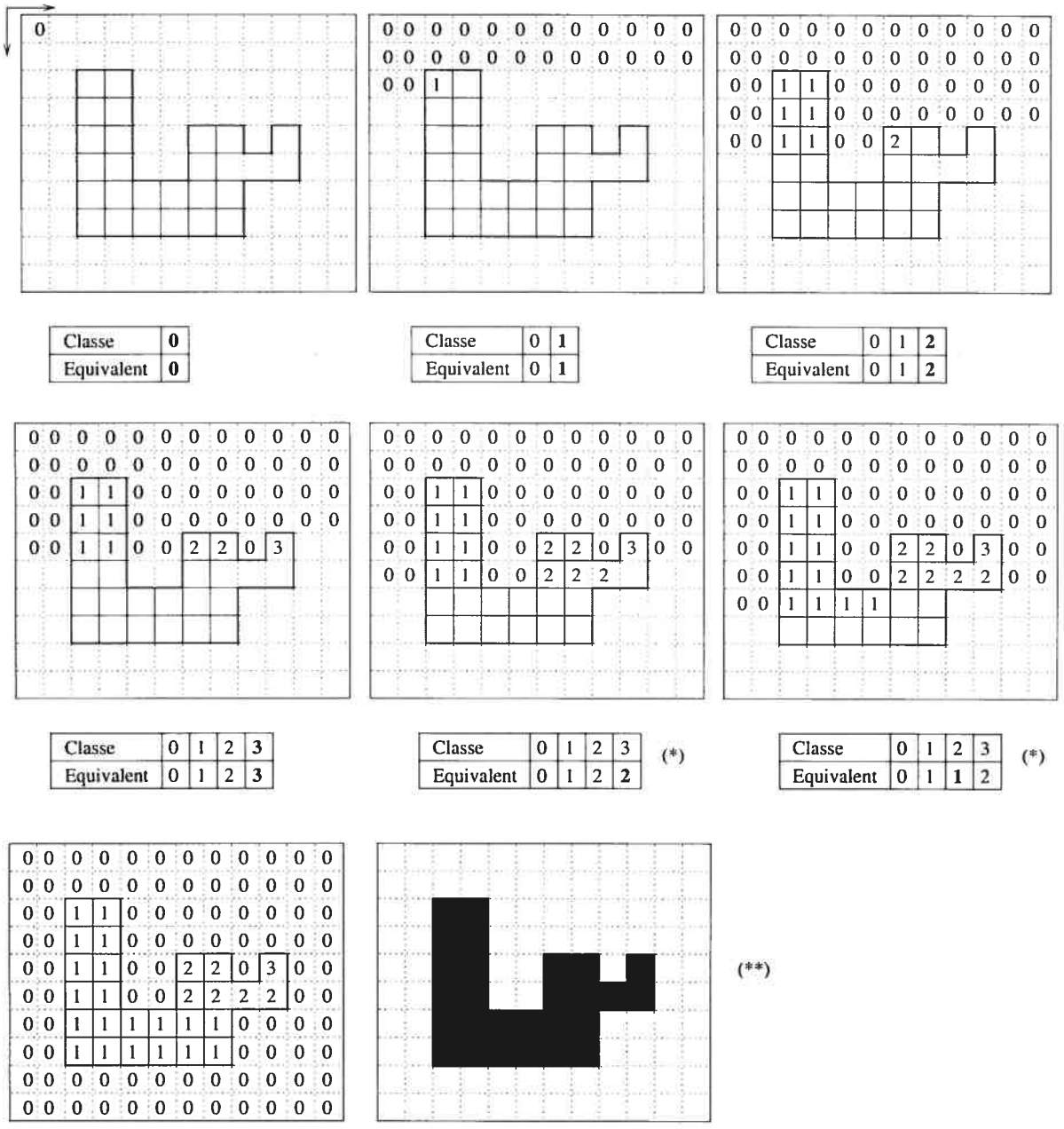


FIG. 4.6 – Exemple d'étiquettage - Sauvegarde(*) et Résolution(**) des équivalences

Dans l'image étiquette, les blobs dont la surface apparente est inférieure à un certain seuil T_{cc} seront ignorés, ceux qui sont retenus seront soumis au module de poursuite de cibles.

4.3 Poursuite de cibles

L'algorithme de poursuite de cibles utilisé se base sur une méthode de *matching* direct. Aucun modèle de description de mouvement ou de prédiction, souvent complexe, n'est nécessaire [19]. En vidéosurveillance, le but du suivi est d'extraire les caractéristiques des trajectoires des cibles et la façon dont celles-ci interagissent entre elles. Ces deux aspects sont souvent révélateurs quant au comportement de l'individu dont les mouvements apparents sont faibles par rapport à leurs extensions spatiales [43]. Par conséquent, une simple correspondance de régions à l'aide de boîtes englobantes paraît plutôt suffisante, en considérant les recouvrements mutuels entre deux ensembles de blobs correspondant au couple d'images successives. Chaque blob est représenté par son plus petit rectangle englobant. Le critère de recouvrement est vérifié si le centre d'un rectangle est contenu dans l'autre.

4.3.1 Notion de matrices de mise en correspondance

Pour la mise en correspondance des deux ensembles de blobs relevant des deux images consécutives, deux matrices sont évaluées : la matrice de correspondance des nouveaux blobs $\{B_i(t)\}$ avec les anciens blobs $\{B_j(t-1)\}$, notée M_{t-1}^t.

$$M_{t-1}^t(i,j) = Matching\{B_i(t-1), B_j(t)\}$$

et la matrice de correspondance réciproque M_t^{t-1}

$$M_t^{t-1}(i,j) = Matching\{B_i(t), B_j(t-1)\}$$

Pour simplifier la correspondance, on introduit le concept de *série de correspondance.*

$$S_{t-1}^t(i) = \bigcup_j j \qquad \text{tel que} \qquad M_t^{t-1}(i,j) = 1$$

Par exemple, la colonne $S_{t-1}^t(III)$ désigne l'ensemble des blobs 'k' du frame t correspondant au blob 'III' du frame $t-1$, $\{2,3\}$ dans l'exemple de la figure (4.7).

$$\mathbf{M_{t-1}^t} = \begin{array}{c} \\ I \\ II \\ III \\ IV \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \qquad \mathbf{M_t^{t-1}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} I & II & III & IV \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$\mathbf{S_{t-1}^t} = \underbrace{\begin{bmatrix} 1 & 1 & 2 & 4 \\ & & 3 & \end{bmatrix}}_{I \;\; II \;\; III \;\; IV} \qquad \mathbf{S_t^{t-1}} = \underbrace{\begin{bmatrix} I & & & \\ II & III & III & IV \end{bmatrix}}_{1 \quad 2 \quad 3 \quad 4}$$

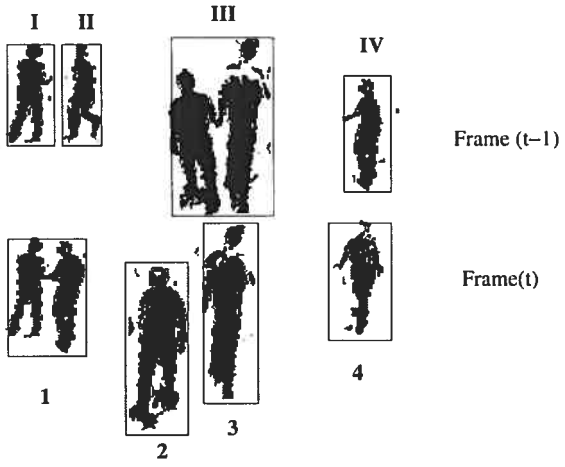


FIG. 4.7 – Exemple d'évolution temporelle de blobs.

4.3.2 Évolution temporelle des blobs et des évènements

L'algorithme supervise l'évolution des blobs d'un frame au suivant en analysant le contenu de leurs séries de correspondance (TAB.4.1). Le système prend en charge des événements simples tels que l'apparition de personnes et leur disparition de la scène, le regroupement et la séparation.

Par exemple :

$$\left.\begin{array}{l} S_{t-1}^{t}(III)=\{2,3\} \\ S_{t}^{t-1}(2)=\{III\} \quad \text{et} \quad S_{t}^{t-1}(3)=\{III\} \end{array}\right\} \Longrightarrow \{III\}\overrightarrow{Division}\{2\}\cup\{3\}$$

Évenement	Séries de correspondance	
Regroupement	$B_i(t-1)\cup B_j(t-1)\equiv B_k(t)$ $\Leftrightarrow$	$S_{t-1}^{t}(i)=S_{t-1}^{t}(j)=\{k\}$; $S_{t}^{t-1}(i)=\{i\}\cup\{j\}$
Séparation	$B_i(t-1)\equiv B_j(t-1)\cup B_k(t)$ $\Leftrightarrow$	$S_{t-1}^{t}(i)=\{j\}\cup\{k\}$; $S_{t}^{t-1}(j)=S_{t}^{t-1}(k)=\{i\}$
Apparition	$B_i(t)\equiv New$ $\Leftrightarrow$	$S_{t-1}^{t}(j)\neq\{i\}\quad\forall j$; $S_{t}^{t-1}(i)=\varnothing$
Disparition	$B_i(t-1)\equiv Leaves$ $\Leftrightarrow$	$S_{t-1}^{t}(i)=\varnothing$; $S_{t}^{t-1}(j)\neq\{i\}\quad\forall j$
Correspondance	$B_i(t-1)\equiv B_j(t)$ $\Leftrightarrow$	$S_{t-1}^{t}(i)=\{j\}$; $S_{t}^{t-1}(j)=\{i\}$

TAB. 4.1 – Différentes configurations des séries de correspondance.

Lorsque le blob forme un groupe, l'algorithme utilise l'information du groupe et non du blob individuel. Dans ce cas, on doit maintenir une information supplémentaire relative à la couleur du blob individuel tout juste avant le regroupement, ce qui permettra de retrouver la personne correspondante lors de sa séparation éventuelle, puisqu'il est difficile de segmenter individuellement des personnes

quand elles sont regroupées. La répartition des couleurs du blob est représentée par une estimation d'une fonction de densité de probabilité (FDP) à partir d'un échantillonnage qu'on a pris uniforme, de pixels recouvrant la personne, grâce à une technique inspirée de l'algorithme *Convergent k-means clustering*, dont la paramétrisation permet d'éviter les problèmes d'initialisation et de synthétiser une distribution des centroïdes avec une grande entropie [31].
La résolution d'une séparation de blobs, dont on connaît les distributions de probabilités de couleur avant le regroupement, revient à chercher le couple de blobs qui maximise la fonction de similitude (EQU. 5.1, page 63) , en supposant que la cible garde une répartition de couleurs plus ou moins identique.

Les MVOs valides, ceux qui ont été correctement traqués pendant un nombre suffisant de frames seront soumis au modèle de reconnaissance de comportements.

4.4 Reconnaissance par approche comportementale

Le système de surveillance effectue un monitoring de trajectoires basé sur l'hypothèse selon laquelle des situations à risque impliquent un comportement anormal, en affichant une attitude assez particulière comme : quelqu'un qui marche en longeant le mur, court trop vite, marche dans le "mauvais" sens ou dans des sites à accès réglementé ou comme celui qui reste immobile "trop" longtemps après une chute par exemple ou dans le cas d'un groupe d'individus qui s'immobilise ou une bousculade ... etc.
Par conséquent, on identifie une situation suspicieuse comme une déviation par rapport à l'ensemble de comportements habituels.

Explicitement, le comportement est déterminé à l'aide d'un ensemble de trajectoires définies dans un espace de comportements. Cette définition dépend de plusieurs suppositions. Premièrement, l'espace des mesures représente l'espace entier des comportements. Ainsi, cet espace dépend étroitement du choix des attributs et de leur cardinalité. Si les attributs sont inappropriés au problème ou si le nombre d'attributs n'est pas assez suffisant, le système ne sera pas en mesure de distinguer les différents comportements. Deuxièmement, les comportements peuvent être modélisés par des patterns représentant des clusters de vecteurs observations dans l'espace des attributs. Cette supposition est due à l'utilisation de la SOM, nécessitant des motifs d'apprentissage, pour modéliser les trajectoires point à point ou semi complètes. En outre on peut supposer que les instances de comportements sont mutuellement exclusives. Cette supposition "forte" permet au processus de décision non seulement de caractériser une trajectoire donnée comme normale mais permet aussi de donner une interprétation en termes symboliques à chacune d'entre elles.

4.4.1 Paramétrisation de trajectoires

A. Le vecteur flux

L'information enregistrée lors du suivi est un ensemble de séquences de points 'centroïdes' permettant de définir pour chaque objet une trajectoire sous forme de vecteurs flux $F = [x, y, \dot{x}, \dot{y}]$, représentant sa position et sa vitesse instantanée. Ce type de mémorisation des activations antérieures permet au système de fonctionner qu'avec des trajectoires partielles [33].
Pour une échelle de temps correspondant à des décisions à plus ou moins long terme une augmentation de la largeur de la fenêtre d'observation est nécessaire.

À la différence de l'approche de moyennage adoptée par [51] pour capturer les propriétés globales des MVOs, dans laquelle les vecteurs flux (F) sont pondérés, proportionnellement, par leurs positions temporelles respectives dans la séquence, nous avons adopté une approche hiérarchique (FIG.4.8) où l'on considère des éléments trajectoire (F) à différentes résolutions temporelles ($F_{r_1}, \ldots, F_{r_n}$).

FIG. 4.8 – Séquences de segments à différentes résolutions temporelles.

Pour absorber les pics de variation et minimiser les effets importuns de discrétisation et de spatialisation du temps, un lissage temporel est effectué sur les valeurs instantanées des vecteurs flux : $SF_{r_i} = [s(x), s(y), s(\dot{x}), s(\dot{y})]$.
La fonction de lissage $s(\cdot)$ est de la forme :

$$s(x) = \mu \cdot x(t) + (1 - \mu) \cdot x(t-1) \quad \text{où} \quad \mu \lesssim 1 \text{ est la constante de lissage}$$

B. Les descripteurs elliptiques de Fourier (DEF)

Pour permettre au système de prendre en charge des trajectoires plus complètes afin de pouvoir extraire des propriétés globales du mouvement, nous avons implémenté une autre approche basée sur les descripteurs elliptiques de Fourier [38] (voir section 5.2.3.B). Dans ce cas, le nombre de points considérés exprime la résolution temporelle.
Supposons que les trajectoires sont des courbes approximées par des segments

générés à partir de P points et que $x(l)$ et $y(l)$ représentent respectivement les projections de ces courbes sur les axes des x et des y, alors les séquences des coordonnées $x(l)$ et $y(l)$, le long de la trajectoire, peuvent être représentées par des fonctions périodiques discrètes. Ces fonctions peuvent, cependant, être approximées par deux séries de Fourier discrètes (FIG.4.9).

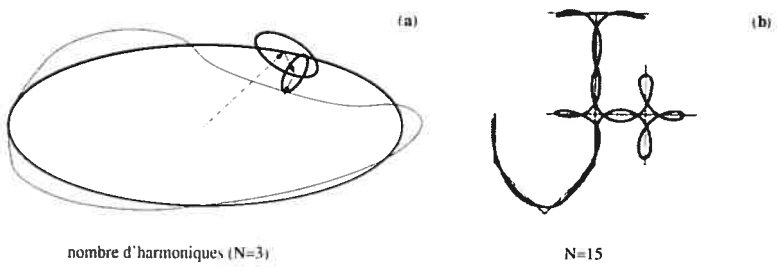


FIG. 4.9 – Approximation elliptique de contour : (a) fermé, (b) ouvert.

Si L est la longueur totale de la courbe fermée [4] et l représente la longueur d'une portion de la courbe joignant le point de départ à un autre point de la courbe, alors ces séries de Fourier peuvent être approximées par :

$$\hat{x}(l) = A_0 + \sum_{n=1}^{N} \left[a_n \cos(\frac{2n\pi l}{L}) + b_n \sin(\frac{2n\pi l}{L})\right]$$

$$\hat{y}(l) = C_0 + \sum_{n=1}^{N} \left[c_n \cos(\frac{2n\pi l}{L}) + d_n \sin(\frac{2n\pi l}{L})\right]$$

Les coefficients de Fourier a_n, b_n, c_n, d_n $\{n = 1..\infty\}$ sont définis par :

$$a_n = \frac{L}{2n^2\pi^2} \sum_{p=1}^{P} \frac{\Delta x_p}{\Delta l_p} \left[\cos(\frac{2n\pi l_p}{L}) - \cos(\frac{2n\pi l_{p-1}}{L})\right]$$

$$b_n = \frac{L}{2n^2\pi^2} \sum_{p=1}^{P} \frac{\Delta x_p}{\Delta l_p} \left[\sin(\frac{2n\pi l_p}{L}) - \sin(\frac{2n\pi l_{p-1}}{L})\right]$$

$$c_n = \frac{L}{2n^2\pi^2} \sum_{p=1}^{P} \frac{\Delta y_p}{\Delta l_p} \left[\cos(\frac{2n\pi l_p}{L}) - \cos(\frac{2n\pi l_{p-1}}{L})\right]$$

[4] On suppose que toute courbe *ouverte* peut être considérée comme une courbe *fermée* qui revient sur elle-même (FIG.4.9.b), ceci ne change en rien la forme globale de trajectoires. Cependant, l'information instantanée du mouvement est codée à part (voir section 5.2.3.B).

$$d_n = \frac{L}{2n^2\pi^2} \sum_{p=1}^{P} \frac{\Delta y_p}{\Delta l_p} \left[\sin(\frac{2n\pi l_p}{L}) - \sin(\frac{2n\pi l_{p-1}}{L}) \right]$$

Où $\Delta\, x_p = x_p - x_{p-1}$, $\Delta\, y_p = y_p - y_{p-1}$ et $\Delta\, l_p = \sqrt{\Delta\, {x_p}^2 + \Delta\, {y_p}^2}$. P représente le nombre total de points de la courbe, x_p et y_p désignent les coordonnées du point p et n est l'ordre des descripteurs. Les descripteurs a_n, b_n, c_n, d_n définissent des ellipses approximant l'allure de la courbe. Ces coefficients dépendent du point de départ (x_1, y_1), de l'aspect et de la grandeur de la courbe ainsi que de son centre de gravité approximé par la position (A_0, C_0).
Dans notre implémentation, les trois premières harmoniques (N=3) ont été retenues (voir section 5.2.3.B).

4.4.2 Détection de comportements anormaux par la SOM

En reconnaissance de formes, dans le cas où aucune hypothèse n'est faite sur la distribution des données, les méthodes statistiques de modélisation ne sont généralement pas efficaces. La tendance est de procéder par une modélisation par réseaux de neurones, lesquels apprennent les distributions des scénarios normaux.

Particulièrement, devant l'extrême variation des scénarios et dans l'incapacité de pouvoir énumérer tous ce qui relève de l'anormalité, la SOM (Section. 3.3-page 33) paraît un avantage déterminant. Par ailleurs, elle offre un outil visuel en cartographiant les comportements types dans l'espace des attributs sous formes de cartes de comportements.
L'algorithme d'apprentissage génère l'ensemble de *motifs de comportements types* en opérant en deux phases : Une étude spontanée des activations du réseau suivie d'un renforcement de cette réponse. Soit x, le contenu du vecteur caractéristique (SF_{r_i} [5] ou DEF [6]), l'entrée de la carte et w_k les poids synaptiques du neurone k (FIG.3.5, page 35), cet algorithme se présente comme suit :

[5] Le lissé du vecteur flux à la résolution r_i.

[6] Vecteur des n premières harmoniques a_i, b_i, c_i, d_i.

1. Spécification de l'architecture de la carte (choix du nombre de neurones et définition des relations du voisinage) et initialisation des poids w_k,
2. Choisir aléatoirement un exemple de la base d'apprentissage et repérer le neurone vainqueur $k \quad : \quad \|w_k - x\| \leq \|w_j - x\| \qquad \forall j$
3. Modification des vecteurs poids de l'unité k et de ses voisins selon la règle :

$$w_j(t+1) = w_j(t) + \alpha(t) \cdot \nu(j,k,t) \cdot \big(x - w_j(t)\big) \quad 0 \leq \alpha(t) \leq 1$$

Possédant une décroissance hyperbolique, le pas d'apprentissage $\alpha(t) = \frac{a}{b+t}$ est une fonction dépendante du nombre d'époques et satisfaisant les conditions de l'approximation stochastique : $\sum_{t=0}^{\infty} \alpha(t) = \infty$ et $\sum_{t=0}^{\infty} \alpha^2(t) < \infty$.
La fonction de voisinage $\nu(j,k,t) = e^{-\frac{d^2(j,k)}{2\sigma^2(t)}}$ est une gaussienne évoluant dans le temps, où $d(j,k)$ représente la distance entre l'élu d'indice k et le neurone d'indice j, et peut être définie de plusieurs manières, la distance de *city-block* en est la plus simple : $d(j,k) = |m_j - m_k| + |n_j - n_k|$, (m,n) correspond à des positions dans la carte. Le rayon d'attraction est donné par : $\sigma(t) = \sigma_{initial}\big(\sigma_{final}/\sigma_{initial}\big)^{t/t_{max}}$, en fixant $\sigma_{initial} = 1.5$ et $\sigma_{final} = 0.5$.

Durant la surveillance, le vecteur caractéristique (SF_{r_i} ou DEF) est soumis à la SOM, aussitôt, le neurone gagnant (vecteur prototype) est identifié, et si la distance euclidienne correspondante excède un seuil alors il n'est pas reconnu et la trajectoire sera considérée comme 'suspecte' et une alarme est émise.
Étant critique pour la reconnaissance, la valeur du seuil est déterminée à la fin du processus d'apprentissage et représente la distance maximale entre les vecteurs d'apprentissage et leurs neurones gagnants respectifs, ce qui garantit une sensibilité maximale aux cas inconnus.

Chapitre 5

Résultats et discussion

5.1 La base de données des scénarios

Pour la mise à l'épreuve du système, des scénarios habituels (TAB.5.1) et inhabituels (TAB.5.2) ont été enregistrés avec une résolution de 320x240 pixels, à partir d'une USB-WebCam, installée dans le corridor de l'entrée principale du pavillon André-Aisenstadt de l'Université de Montréal.
Pour pouvoir tester les vraies performances et les capacités réelles du système, on s'est imposé des conditions de prise de vue assez difficiles telles que :

- la caméra observe la scène avec un angle plongeant.
- la scène est influencée par l'éclairage extérieur provenant d'une porte vitrée située presque en face de la caméra.
- l'éclairage dans la scène est non uniforme.
- des surfaces de nature différente : un plancher très réflectif; un escalier sombre et un peu rugueux.
- l'équilibrage fluorescent des niveaux de blanc a été activé : afin de simuler une source de lumière non complètement blanche.

Séquence	Description
Escalier	montée et descente (rencontre éventuelle (2 personnes)).
Table	s'asseoir (se relever, revenir ou s'éloigner et quitter via la porte).
Marche	apparaître ou quitter via la porte, avec ou sans rencontre.

TAB. 5.1 – Exemples de séquences normales.

Séquence	Description
Accélération	courir.
Position anormale	s'étendre sur la table, monter sur un pilier.
Bousculade	se tirailler.
Malaise	chute dans l'escalier, évanouissement sur le plancher et sur la table.
Mouvement bizarre	marcher en longeant les murs, tourner en rond, effectuer des allers-retours, "forcer" une porte barrée.

TAB. 5.2 – Exemples de séquences anormales.

Les données d'apprentissage consistent en 46 trajectoires normales, contenant 3615 points. L'ensemble de test comprend 19 trajectoires inhabituelles contenant 2174 points. En moyenne, on a 3 trajectoires par séquence.

5.2 Résultats

5.2.1 Détection et classification des changements

Chaque pixel x est classé comme avant-plan selon son intensité $I(x)$, l'intensité moyenne de l'arrière plan $BCK(x)$ et la différence *interframe* absolue maximale $\sigma_c(x)$. La classification de pixel comme appartenant à un MVO ou à

l'ombre est basée sur une propriété physique de cette dernière en utilisant la distorsion chromatique et de brillance, et emploie une pré-segmentation chromatique/achromatique de l'arrière plan à cause des niveaux de variation des intensités du *background* qui diffèrent selon leurs degrés d'illumination.
L'insensibilité du modèle par rapport aux faibles variations d'illumination démontre de bonnes performances (FIG. 5.1 à 5.5) et donne un taux de détection (VP) élevé, une très faible omission, et notamment un pourcentage de commissions (FP) acceptable pour un algorithme en temps réel (TAB.5.3). Dans ces figures, la couleur bleue désigne les détections résultant d'une distorsion de brillance sensible, les pixels en rouge montrent ceux qui présentent une distorsion chromatique significative. Les ombres ont été correctement isolées (les régions en vert).

	VP	**Omissions**	**FP**
Taux de détection	97.47%	2.53%	7.60%

TAB. 5.3 – Performances moyennes du modèle conique.

Les faux positifs (7.6%) sont en majorité dus aux ombres fortes situées tout juste sous les MVOs. En pratique, ce type de détection n'influe pas trop le système puisque l'algorithme de suivi de cibles ne se sert que de la boîte englobante minimale.

5.2.2 Suivi

La fréquence d'acquisition de la caméra est de 30 *images/sec.* Réellement, le système fonctionne à une fréquence inférieure à cause du temps nécessaire pour les différents traitements mais assez suffisante pour ne pas perdre des blobs et pour une mise en correspondance correcte. La vitesse minimale observée est de $16Hz$ sur un processeur $2.66GHz$ grâce un algorithme typique de suivi qui

FIG. 5.1 – Résultats de détection / Séquence de test *Plancher/Corridor/Diro/UdeM.*

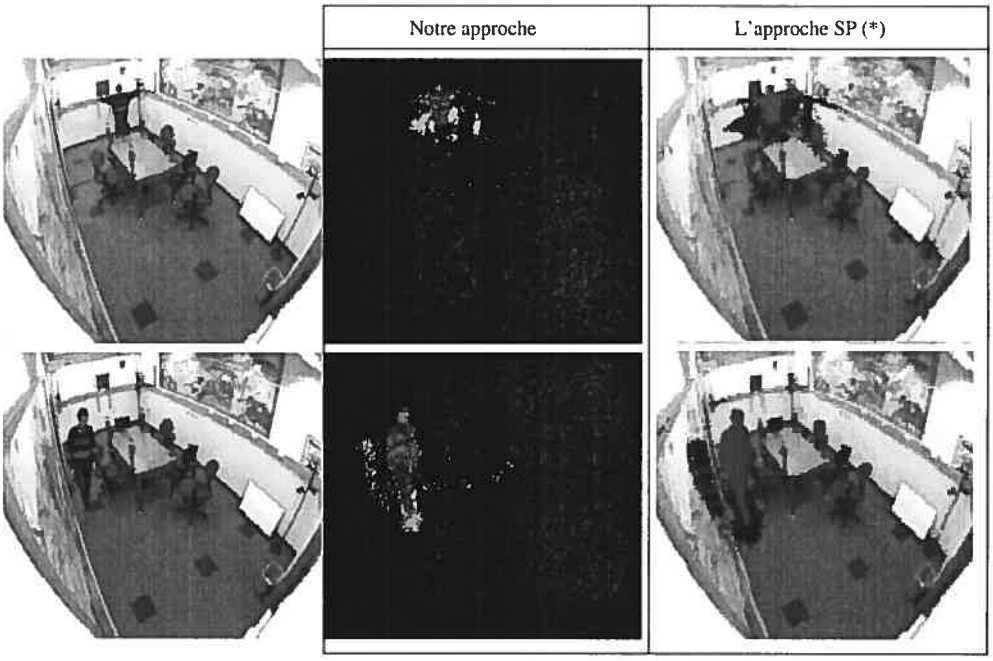


FIG. 5.2 – *Intelligent room*-http ://cvrr.ucsd.edu/aton/shadow (*)Statistique Paramétrique (section 2.1.3.A) : relativement, la personne a été bien détectée, dans notre cas on a moins de pixels *ombre* ceci est dû à un τ_1 plus faible (0.75) .

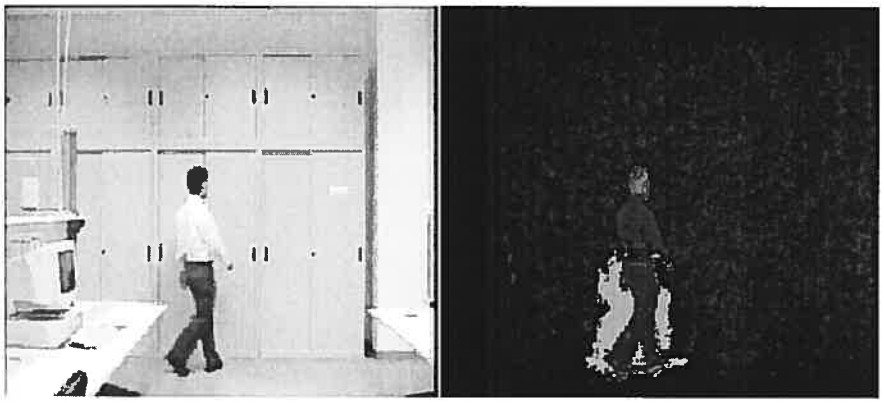


FIG. 5.3 – Séquence de test *Laboratory raw* (http ://cvrr.ucsd.edu/aton/shadow).

FIG. 5.4 – Résultats de détection / Séquence de test *Escalier/Corridor/Diro/UdeM.*

FIG. 5.5 – Séquence de test *Lab. vision 3D/Diro/UdeM.*

dépend uniquement des informations de correspondance entre blobs.

a. Correspondance b. Groupement c. Séparation

FIG. 5.6 – Suivi de cibles (groupement et séparation).

Le système ne conserve pas individuellement le centroïde de l'objet suivi lorsqu'il y a regroupement. C'est le centroïde de tout le groupe qui sera considéré (FIG.5.7 du frame 38 à 44 et de 57 à 68). Après un événement *séparation* (FIG.5.6.c), une procédure de *matching* est lancée pour retrouver les blobs correspondants à ceux d'avant le regroupement, la mise en correspondance est établie par une représentation utilisant une mixture de gaussiennes de la distribution des couleurs du blob établie à partir d'un maillage régulier [1] de points (i, j). Les paires de blobs $(B_{groupé}, B_{séparé})$ maximisant l'équation (5.1) sont considérés comme correspondants à la même personne.

$$Sim(B_{groupé}, B_{séparé}) = \sum_{p(i,j)} \sum_{k}^{n} \exp\left(\frac{-\|x - x_k\|^2}{2 \cdot r_k^2}\right) \tag{5.1}$$

Où $x_{(r,g,b)}$ est la couleur du point $p \in B_{groupé}$. La $k^{ème}$ PDF des n gaussiennes représentant la couleur de $B_{séparé}$ est centrée sur x_k avec un rayon r_k.

[1] Il est toutefois possible d'adopter une stratégie pour avoir un maillage non-uniforme pour une meilleure représentation de la couleur du blob.

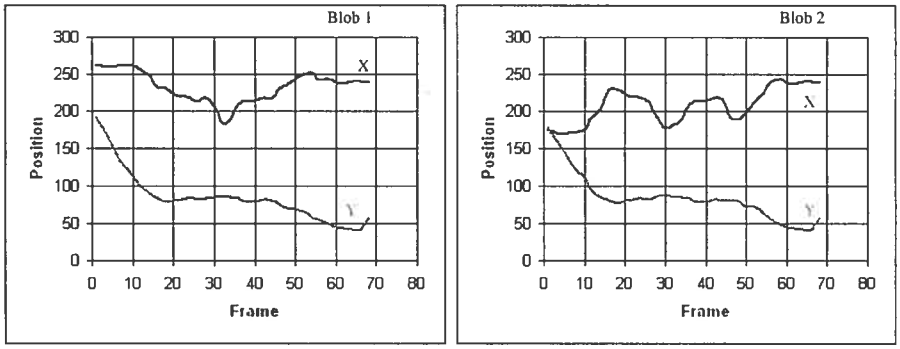


FIG. 5.7 – Trajectoires anormales des centroides des cibles (FIG.5.6).

5.2.3 Analyse de comportements

A. Par l'architecture hiérarchique de SOMs

Dans cette expérience, on considère une structure hiérarchique de SOMs, dont les paramètres sont présentés dans le tableau (5.4), pour pouvoir caractériser les propriétés globales du mouvement. Les données d'apprentissage des niveaux 1, 2

niveau	dimension	entrées
1	30x30	$s(x), s(y), s(\dot{x}), s(\dot{y})$
2	25x25	"
3	20x20	"

TAB. 5.4 – Paramètres de la structure hiérarchique.

et 3 comportent 3615, 1239 et 722 points obtenues à 2 *Hz*, 1/1.5 *Hz* et 1/2.5 *Hz* respectivement, en fixant la constante de lissage temporelle μ à 0.9.

Les résultats peuvent être résumés comme suit :

- classification correcte $39/42 = 92.9\%$
- trajectoires inhabituelles correctement classifiées $18/19 = 94.7\%$
- trajectoires normales correctement classifiées $21/23 = 91.3\%$

	Résultat de la classification	
Séquences	**Normal**	**Anormal**
Normal	21	2
Anormal	1	18

TAB. 5.5 – Matrice de confusion.

Le pourcentage de bien classés (92.9%) révèle l'assez bonne qualité de la description de l'ensemble des scénarios "normaux", relativement au nombre ayant servi à l'apprentissage (23) qui quand même sous-représente l'espace complet de comportements. La figure (5.8) présente quelques scénarios ayant été correctement reconnus.

Les fausses détections sont généralement dues aux scénarios où l'on enregistre une

(a) S'asseoir & quitter via la porte (b) Monter l'escalier (c) S'immobiliser et discuter

FIG. 5.8 – Exemples de scénarios normaux correctement reconnus.

variation assez brutale de la position du centroïde des boîtes englobantes dont les dimensions varient remarquablement. Ces cas surviennent particulièrement dans l'escalier lorsque deux individus se croisent (FIG.5.9) ou lorsqu'il y a séparations-fusions intermittentes de blobs (FIG.5.12).

Des trajectoires inhabituelles désormais classées comme normales sont celles où la personne s'assoit à la table pendant un bout de temps puis tout à coup s'y cogne la tête. Le caractère local de l'événement ne permet pas de lever l'ambiguïté à cause de la similitude qu'il présente avec l'événement *s'asseoir* (FIG.5.10). Par conséquent, son appréhension nécessite une description plus "fine" à haute

FIG. 5.9 – Cas de séquences présentant de fausses alarmes.

résolution, on parlera alors ici de reconnaissance de gestes humains.

(a) Anormale (b) Normale

FIG. 5.10 – Cas d'omission nécessitant une plus haute résolution.

Mis à part ces cas justifiés, toutes les autres trajectoires ont été correctement détectées (FIG.5.11).

B. Par descripteurs elliptiques de Fourier

Dans cette deuxième ébauche d'analyse on se sert de deux cartes topologiques de Kohonen (TAB.5.6). La première carte permet de coder l'information instantanée, tandis que la deuxième mémorise l'aspect global de la courbe en utilisant les descripteurs elliptiques de Fourier [2].

[2] Seules les trois premières harmoniques $(a_i, b_i, c_i, d_i)_{i=1...3}$ ont été retenues.

(a) Évanouissement (b) Se déplacer en longeant le mur (c) Tourner en rond (d) Se tirailler

FIG. 5.11 – Exemples de comportements atypiques correctement détectés.

SOM	dimension	entrées
Contexte ponctuel	30x30	$s(x), s(y), s(\dot{x}), s(\dot{y})$
Aspect global	30x30	$(a_i, b_i, c_i, d_i)_{i=1...3}$

TAB. 5.6 – Paramètres des deux SOMs.

Les données d'apprentissage comportent 3615 points qui sont générées à 2 Hz, en utilisant la même fonction de lissage $s(\cdot) = \mu\, x_t(\cdot) + (1-\mu)\, x_{t-1}(\cdot)$ avec $\mu = 0.9$. Les résultats se résument comme suit :

	Résultat de la classification	
Séquences	**Normal**	**Anormal**
Normal	22	1
Anormal	1	18

TAB. 5.7 – Matrice de confusion.

- classification correcte $40/42 = 95.2\%$
- trajectoires inhabituelles correctement classifiées $18/19 = 94.7\%$
- trajectoires normales correctement classifiées $22/23 = 95.6\%$

Des tableaux (5.5 et 5.7), il ressort que la reconnaissance considérant les descripteurs elliptiques de Fourier présente une diminution des fausses détections, augmentant ainsi le taux de sensibilité du système car, comparativement à la

représentation hiérarchique, le codage par DEF des relations temporelles permet d'éviter le phénomène de spatialisation du temps que la fonction de lissage n'a pas pu pallier à cause des variations très brutales de la position des centroïdes notamment dans les séquences *Escalier* et plus particulièrement lors de fusion et de séparation très longitudinale de blobs (FIG.5.9).

Les fausses détections sont dues à la sensibilité des DEF aux mouvements périodiques de par leur construction. Dans la séquence de la (FIG.5.12), les deux blobs sont tellement proches l'un de l'autre que les positions des centroïdes associés soient sévèrement perturbées par des fusions et des séparations intermittentes déclenchant de fausses alarmes.

FIG. 5.12 – Cas de séquences présentant de fausses alarmes.

Le nombre de séquences inhabituelles ayant été reconnues comme normales (le taux d'omissions) demeurent inchangé, car comme évoquer précédemment cette situation nécessite, tout naturellement, une description plus fine du mouvement.

Chapitre 6

Conclusion et perspectives

Partant du processus de surveillance tel qu'effectué par l'opérateur humain (*appréhender-suivre-reconnaître*) notre objectif final consistait à mettre en œuvre un système de surveillance automatique en détectant des événements suspicieux dans une scène.

La détection des objets en mouvement est basée sur un algorithme de soustraction de fond adaptatif intégrant un modèle flexible et efficace d'élimination de l'ombre basé sur le principe de *color constancy* en modélisant la luminance et la chromacité du pixel et leur distorsions respectives dans le système RGB, un seuillage simple permet de segmenter l'avant-plan du *background* et des ombres. Cependant, les ombres fortes, moins graves dans notre cas, peuvent être éliminées par de simples critères basés sur des considérations géométriques. Les données relatives aux mouvements de chaque personne sont générées à l'aide d'un algorithme de poursuite de cibles très performant en temps réel utilisant des critères de recouvrements directs et inverses entre blobs. Ces données temporelles sont ensuite injectées dans un module de reconnaissance de comportements. Deux ébauches ont été présentées la première consistait en une architecture hiérarchique utili-

sant trois cartes topologiques de Kohonen pour l'intégration du temps. Dans la deuxième démarche on utilise deux réseaux de Kohonen mais en codant cette fois-ci la séquence des données temporelles à l'aide des descripteurs elliptiques de Fourier qui permettent d'approximer l'allure de trajectoires à l'aide d'ellipses. Dans ce cas, on cherche à dégager deux aspects du mouvement : les propriétés ponctuelles et globales.
Dans les deux cas, l'apprentissage est par l'exemple où seule une redondance suffisamment produite dans les données permet aux SOMs d'acquérir une certaine connaissance.

La meilleure façon d'inclure la composante temps est de la codée à l'intérieur du réseau plutôt que d'encoder les données elles-mêmes, nous seront peut-être amenés à nous intéresser à l'utilisation des modèles de SOM spatio-temporels capables d'incorporer les relations temporelles dans le réseau lui-même.
En outre nous nous intéressons à la détection des anomalies à partir de la SOM par seuillage dynamique [69] en utilisant l'information stockée dans les connexions latérales de la carte plutôt qu'un seuil calculé à partir de l'erreur maximale qui ne tient pas compte de la distribution des données dans le voisinage du neurone.
En ce qui concerne l'apprentissage, on suggère d'adopter une approche incrémentale, pour permettre au réseau d'apprendre progressivement des situations potentielles n'ayant pas été présentes dans la base d'apprentissage.

La complexité des tâches de détection, du suivi et de l'analyse pose un certain nombre de problèmes pratiques, empirés davantage par la qualité instable du flux vidéo et le champ de vision plus ou moins restreint du capteur (75°).
Dans les scènes d'intérieur, les problèmes de détection sont dus au changement d'éclairage (une lumière qui s'éteint/s'allume, une fenêtre ou une porte qui s'ouvre/se ferme, des objets très réfléchissants ...). Une amélioration de la stratégie de détection pourrait se faire en ajoutant davantage de boucles de retour à partir

des modules de reconnaissance et de suivi pour exploiter des cohérences spatio-temporelles.

Quant au processus de poursuite de cibles, les occlusions intermittentes sont les principales causes de problèmes qu'on peut résoudre par une stratégie de *handoff/handover* [1] en multi-caméras, cependant le recouvrement entre les champs de vision est une condition nécessaire pour effectuer correctement la poursuite à travers la scène.
Pour remédier aux problèmes de changement brutal des centroïdes des blobs fusionés, il semble plus judicieux de mettre en œuvre un procédé de suivi qui lorsqu'il y a regroupement considère les centroïdes individuels de chaque blobs et non pas un seul centroïde (le représentant de tout le groupe). Cette option peut être pratique par la mise à profit des propriétés de recouvrement de l'algorithme de poursuite de cibles adopté qui permet de limiter considérablement l'espace de recherche seulement à la boîte englobante minimale du groupe.

À un niveau supérieur du système, le module de prise de décision est contraint par des impératifs universels pour assurer la fiabilité de la surveillance : **sensibilité :** proportion d'évènements normaux observés prédits comme évènements normaux ; **spécificité :** proportion d'évènements inhabituels observés prédits comme évènements inhabituels ; **temps de détection :** durée acceptable entre l'occurrence et la détection d'un évènement.

L'étude menée dans cette thèse distingue deux différentes approches. La première, une approche hiérarchique où les tracés des trajectoires sont modélisés à différentes résolutions temporelles, offre une meilleure spécificité, tandis que la seconde, basée sur des descripteurs elliptiques de Fourier où l'allure des trajectoires est approximée par un nombre défini d'ellipses, démontre une sensiblité supérieure pour une spécificité comparable.

[1] Mode coopératif : caméras relais d'un angle d'observation à l'autre.

Pour aller au delà de cette dualité sensibilité/spécificité, d'une part, nous comptons inclure dans notre système des fonctionnalités de grande et de moyenne résolution en intégrant des techniques de reconnaissance de gestes humains afin d'enrichir la modélisation de comportements spatio-temporels. D'une autre part, nous estimons que l'intégration des connaissances à priori avec prise en charge de la forme symbolique et l'aspect incertain des informations, qu'on trouve réunies chez l'opérateur humain sous forme de règles de déduction, présente un potentiel déterminant pouvant améliorer considérablement le processus de reconnaissance. Une représentation neuro-floue [29] constituerait sans doute une solution plus complète qui allie la capacité d'apprentissage des réseaux de neurones et les capacités de raisonnement des systèmes d'inférence.

Bien qu'un grand travail (circonstance oblige) ait été fait dans un domaine aussi jeune que la télésurveillance, beaucoup d'options restent ouvertes comme la segmentation, la modélisation et notamment le traitement des occlusions lors du suivi. Néanmoins, cette étude "préliminaire" nous a permis de soulever les problèmes inhérents à la vidéosurveillance, offrant ainsi un contexte permettant d'orienter des études encore plus ciblées pour la suite du projet qui pourra faire partie d'une architecture plus générique d'un système de télésoins permettant de surveiller la santé d'un patient et à même de poser un diagnostic médical.

Bibliographie

[1] N. Gershenfeld A. Weigend. *Time Series Prediction : Forecasting the Future and Understanding the Past.* Addison Wesley, 1994.

[2] Addison J.F D. & al. Methods for Integrating Memory into Neural Networks Applied to Condition Monitoring. *6th IASTED Int. Conf. Artificial Intelligence and Soft Computing, Canada*, July 2002.

[3] Anandan P. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2 :283–310, 1989.

[4] Bevilacqua A. Effective Object Segmentation in a Traffic Monitoring Application. *Proc. of the 3rd ICVGIP*, pages 125–130, 2002.

[5] Campbell L. & al. Invariant features for 3-D recognition. *Int. Conf. of Face and Gesture Recognitionn*, pages 157–162, 1996.

[6] Campbell L. & al. Learning and Recognising Human Dynamics in Video Sequences. *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 568–574, 1997.

[7] Cavallaro A., Salvador E. and Ebrahimi T. Detecting shadows in image sequences. *Proc. of IEE Conf. on Visual Media Production (CVMP), London (UK)*, March 2004.

[8] Collins R. & al. A system for video surveillance and monitoring. *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA*, CMU-RI-TR-00-12, May 2000.

[9] T.H. Cormen. *Introduction to algorithms.* 2nd ed. Cambridge, Mass. : MIT Press, Boston ; Montréal : McGraw Hill Book, 2001.

[10] T.leonde Cornelius. *Image processing and Pattern Recognition.* Academic Press, San Diego, 1998.

[11] Cucchiara R. & al. Improving shadow suppression in moving object detection with HSV color information. *Proc. of The IEEE 4th Inter. Conference on Intelligent Transportation Systems*, pages 334–339, August 2001.

[12] Cuntoor N., Kale A., Chellappa R. Combining multiple evidences for gait recognition. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 3 :33–36, 2003.

[13] Cutler R., Davis L.S. Robust real-time periodic motion detection, analysis, and applications. *IEEE Tran. on PAMI*, 22(8) :781–796, 2000.

[14] Darrell T. and Pentland A. Space Time Gestures. *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 335–340, 1993.

[15] Dick A., Brooks M.J. Issues in automated visual surveillance. *Proc. of the 7th Int. Con. on Digital Image Computing : Techniques and Applications*, pages 195–204, 2003.

[16] Durucan E., Ebrahimi T. Change detection and background extraction by linear algebra. *Proceedings of the IEEE*, 89(10) :1368–1381, Oct. 2001.

[17] Elman Jeffrey L. Finding structure in time. *Cognitive Science*, 14 :179–211, 1990.

[18] Fejes S., Davis L.S. What can projections of flow fields tell us about the visual motion. *6th Int. Conf. on Computer Vision, ICIP*, pages 979–986, 1998.

[19] Fuentes L.M. and Velastin S.A. People Tracking in Surveillance Applications. *Proc. of the 2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS2001)*, 2001.

[20] Galata A., Johnson N. and Hogg D. Learning Variable Length Markov Models of Behaviour. *Computer Vision and Image Understanding (CVIU)*, 81(3) :398–413, 2001.

[21] Ghosh A. and Schwartzbard A. A study in using neural networks for anomaly and misuse detection. *Proc. of the 8th USENIX Security Symposium Washington, D.C., USA*, August, 1999.

[22] Grossberg S. Adaptive pattern classification and universal recoding : I. parallel development and coding of neural feature detectors. *Biological Cybernetics*, 23 :121–134, 1976.

[23] Haritaoglu I., Harwood D., Davis L. A Fast Background Scene Modeling and Maintenance for Outdoor Surveillance. *Int. Conf. on Pattern Recognition (ICPR'00)*, 4 :179–183, 2000.

[24] Horn BK and Schunck BG. Determining optical flow. *Artificial Intelligence*, 17 :185–203, 1981.

[25] Horprasert T. & al. A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection. *ICCV'99 Frame-Rate Workshop*, 1999.

[26] Isard M. and Blake A. CONDENSATION – conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1) :5–28, 1998.

[27] Jabri S. & al. Detection and Location of People in Video Images Using Adaptive Fusion of Color and Edge Information. *Proc. 15th Int. Conf. on Pattern Recognition*, 4 :627–630, 2000.

[28] Jain A.K., Duin P.W., Mao J. Statistical Pattern Recognition : A Review. *IEEE Trans. on PAMI*, 22(1) :4–37, 2000.

[29] R. Jang. *Neuro-Fuzzy Modeling : Architectures, Analyses, and Applications.* Ph.D. Dissertation, Department of Electrical Engineering and Computer Science, University of California at Berkeley, 1992.

[30] Javed O., Shafique K., Shah M. A hierarchical approach to robust background subtraction using color and gradient information. *Proc. Workshop on Motion and Video Computing*, pages 22–27, December 2002.

[31] J.F. Jodouin. *Les réseaux neuromimétiques.* Hermes, Paris, 1994.

[32] Johnson N. and Hogg D. Learning the Distribution of Object Trajectories for Event Recognition. *Image and Vision Computing*, 14(8) :609–615, 1996.

[33] Johnson N. and Hogg D. Learning the Distribution of Object Trajectories for Event Recognition. *Proc. British Machine Vision Conference*, 2 :583–592, September 1995.

[34] Johnson N., Galata A. and Hogg D. The Acquisition and Use of Interaction Behaviour Models. *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 866–871, 1998.

[35] Karayiannis N.B., Tao G. Extraction of Temporal Motion Velocity Signals from Video Recordings of Neonatal Seizures by Optical Flow Methods. *25th Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, pages 874–877, 2003.

[36] R.Castelman Kenneth. *Digital image processing.* Prentice Hall Inc., New Jersey 07458, 1996.

[37] T. Kohonen. *Self-organization and associative memory.* Springer series in information sciences, Berlin, Springer-Verlag, 1988.

[38] Kuhl F.P. and Giardina C.R. Elliptic Fourier Features of a Closed Contour. *CGIP*, 18 :236–258, 1982.

[39] Kumar P. & al. A comparative study of different color spaces for foreground and shadow detection for traffic monitoring system. *Proc. of The IEEE 5th Inter. Conference on Intelligent Transportation Systems*, pages 100–105, September 2002.

[40] Lichodzijewski P. & al. Host-based intrusion detection using self-organizing maps. *Proc. of the 2002 Int. Joint Conf. on Neural Networks*, pages 1714–1719, 2002.

[41] Lipton A., Fujiyoshi H. and Patil R.S. Moving target detection and classification from realtime video. *In Proceedings of the 1998 Workshop on Applications of Computer Vision*, 1998.

[42] McIvor A., Zang Q., Klette R. The Background Subtraction Problem for Video Surveillance Systems. *Proc. of the Int. Workshop on Robot Vision*, pages 176–183, 2001.

[43] McKenna S.J. & al. Tracking groups of people. *Int. Journal on Computer Vision and Image Understanding*, 80 :42–56, 2000.

[44] Mikic I. & al. Moving shadow and object detection in traffic scenes. *Proc. of Inter. Conference on Pattern Recognition*, 1 :321–324, 2000.

[45] Myers C.S., Rabiner L.R. and Rosenberg A.E. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Trans. Acous., Speech, and Sig. Processing*, 28(6) :623–635, 1980.

[46] Nadimi S., Bhanu B. Physical Models for Moving Shadow and Object Detection in Video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(8) :1079–1087, 2004.

[47] Nair V., Clark JJ. Automated Visual Surveillance Using Hidden Markov Models. *the 15th Vision Interface Conference, Calgary*, pages 88–92, May 2002.

[48] Niu W. & al. Real-time Multi-person Tracking in Video Surveillance. *Proc. of Pacific Rim Multi-media Conference*, 2003.

[49] Ohta N. A statistical approach to background subtraction for surveillance systems. *Eighth IEEE Int. Conf. on Computer Vision ICCV*, 2 :481–486, 2001.

[50] Ong E. and Gong S. A dynamic human model using hybrid 2D-3D representations in hierarchical PCA space. *In British Machine Vision Conference*, 1 :33–42, September 1999.

[51] Owens J. and Hunter A. Application of the Self-Organising Map to trajectory Classification. *Proc.Third IEEE International Workshop on Visual Surveillance*, ISBN 0-7695-0698-4 :77–83, July 2000.

[52] Owens J., Hunter A. and Fletcher E. Novelty Detection in Video Surveillance using Hierarchical Neural Networks. *ICANN 2002*, 3 :1249–1254, August 2002.

[53] Paragios N., Deriche R. Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects. *IEEE Trans. on PAMI*, 22(3) :266–280, 2000.

[54] I. Pitas. *Digital Image processing Algorithms and applications.* New York; Toronto : Wiley & Sons, ISBN 0-471-37739-2, 2000.

[55] Prati A., Mikic I., Trivedi M.M., Cucchiara R. Detecting moving shadows : algorithms and evaluation. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 25(7) :918 – 923, 2003.

[56] Psarrou A., Gong S. and Buxton H. Modelling Spatio-Temporal Trajectories and Face Signatures on Partially Recurrent Neural Networks. *Proc. IEEE Int. Conf. on Neural Networks*, 5 :2226–2231, 1995.

[57] Rabiner L. A Tutorial on Hidden Markov Models and Selected Appli-cations in Speech Recognition. *Proceedings of the IEEE*, 77(2) :257–286, 1989.

[58] Radke R.J. & al. Image Change Detection Algorithms : A Systematic Survey. *Submitted to IEEE Transactions on Image Processing*, http ://www.ecse.rpi.edu/Homepages/rjradke/papers/radketip04.pdf, April 2004.

[59] Rahimi A. at mit.edu. Fast Connected Components on Images. *http ://www.ai.mit.edu/~ rahimi/connected/.*

[60] Regazzoni C., Ramesh V. and Foresti G. Special issue on video communications, processing, and understanding for third generation surveillance systems. *Proceedings of IEEE*, 89(10) :1355–1367, 2001.

[61] Regazzoni C.S., Foresti G.L. New trends in Video Communications, Processing and Understanding in Surveillance Applications. *ICIP*, 2001.

[62] Rosenblum M., Yacoob Y. and Davis L. Human Emotion Recognition from Motion Using a Radial Basis Function Network Architecture. *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 43–49, 1994.

[63] Rosin P.L. Thresholding for Change Detection. *Computer Vision and Image Understanding*, 86(2) :79–95, 2002.

[64] Röwekamp T. & al. Specialized Architectures for Optical Flow Computation : A Performance Comparison of ASIC, DSP, and Multi-DSP. *Proc. 8th*

Int. Conference on Signal Processing Applications & Technology, San Diego, USA, pages 829–833, 1997.

[65] Stauder J. & al. Detection of Moving Cast Shadows for Object Segmentation. *IEEE Transactions on Multimedia*, 1(1) :65–76, 1999.

[66] Stauffer C., Grimson E. Learning Patterns of Activity Using Real-Time Tracking. *IEEE Trans. on PAMI*, 22(8) :747–757, 2000.

[67] Sumpter N. and Bulpitt A. Learning Spatio-Temporal Patterns for Predicting Object Behaviour. *In Proc. British Machine Vision Conference*, 1998.

[68] Tate S., Takefuji Y. Video-Based Human Shape Detection by Deformable Templates and Neural Network. *6th Int. Conf. on Knowledge-Based Intelligent Info. Eng. Syst. and Allied Tech.*, pages 280–285, September 2002.

[69] Taylor O. & al. Improved Classification for a Data Fusing Kohonen Self Organizing Map Using A Dynamic Thresholding Technique. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 2 :828–832, 1999.

[70] Toth D., Aach T. and Metzler V. Bayesian spatio-temporal motion detection under varying illumination. *European Signal Processing Conference EUSIPCO)*, September 2000.

[71] Vesanto J., Alhoniemi E. Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks*, 11(3) :586–600, 2000.

[72] Wang L., Hu W. & Tan T. Recent Developments of Human Motion Analysis. *Pattern Recognition*, 36(3) :585–601, 2003.

[73] Wixson L. Detecting salient motion by accumulating directionally-consistent flow. *IEEE Trans. on PAMI*, 22(8) :774–780, 1998.

[74] Yamato J., Ohya J. and Ishii K. Recognizing Human Action in Time-Sequential Images using Hidden Markov Model. *In IEEE Conf. on Computer Vision and Pattern Recognition*, pages 379–385, 1992.