UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE 1
U.F.R. d'Informatique, Électronique, Électrotechnique et Automatique
Laboratoire d'Automatique LAGIS
Laboratoire d'Informatique Fondamentale LIFL
INSTITUT NATIONAL DE RECHERCHE SUR LES TRANSPORTS ET LEUR SÉCURITÉ
Laboratoire d'Electronique, Ondes et Signaux pour les Transports INRETS

Numéro attribué par la bibliothèque : 3940

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE LILLE 1

Discipline : Automatique et Informatique Industrielle

Présentée et soutenue publiquement

par

Thomas LECLERCQ

le 15 décembre 2006

Contribution à la comparaison de séquences d'images couleur par outils statistiques et par outils issus de la théorie algorithmique de l'information.

JURY :

J.-G. Postaire	Professeur émérite à l'U.S.T.L.	Président de Jury
S. Grigorieff	Professeur à l'Université de Paris 7	Rapporteur
G. Rabatel	D.R. au C.E.M.A.G.R.E.F de Montpellier	Rapporteur
D. Muselet	M.C.F. à l'I.S.T.A.S.E de Saint-Etienne	Examinateur
L. Macaire	M.C.F. H.D.R. à l'U.S.T.L.	Co-Directeur de recherche
J.-P. Delahaye	Professeur à l'U.S.T.L.	Co-Directeur de recherche
L. Khoudour	I.R. à l'I.N.R.E.T.S. L.E.O.S.T	Co-Directeur de recherche

Contribution à la comparaison de séquences d'images couleur par outils statistiques et par outils issus de la théorie algorithmique de l'information.

-

Application à la vidéosurveillance et l'aide à l'exploitation des sites de transports publics.

Thomas Leclercq

15 décembre 2006

Table des matières

Table des figures

Liste des tableaux

Abréviations et notations

Abréviations

ACP	Analyse en Composantes Principales
ADP	Aéroports De Paris
AFNOR	Association Française de NORmalisation
CCD	Charged Couple Device
CIE	Commission Internationale de l'Eclairage
CNIL	Commission Nationale de l'Informatique et des Libertés
CNS 2T	Communication, Navigation et Surveillance pour les Transports Terrestres
CROMATICA	CROwd MAnagement with Telematic Imaging and Communication Assistance
EPIC	Etablissement Public Industriel et Commercial
INRETS	Institut National de REcherche sur les Transports et leur Sécurité
LAGIS	Laboratoire d'Automatique Génie Informatique et Signal
LEOST	Laboratoire d'Electronique, Ondes et Signaux pour le Transport
LIFL	Laboratoire d'Informatique Fondamentale de Lille
PCRDT	Programme Cadre pour la Recherche et le Développement Technologiques

PCC	Poste de Commande Centralisée
PRISMATICA	Pro-active Integrated Systems for Security Management by Technological, Institutional and Communication Assistance
RATP	Régie Autonome des Transports Parisiens
SNCF	Société Nationale des Chemins de Fer
USTL	Université des Sciences et Technologies de Lille

Notations

Notations relatives aux données de la base (séquences-personnes)

$\mathcal{A} \dots \mathcal{H}$	Les huit personnes de la base de données de test
$\mathcal{A}^{(1)} \dots \mathcal{A}^{(4)}$	Les quatre séquences-personnes correspondant aux quatre passages de la personne $\mathcal{A}$
$n_{\mathcal{A}^{(1)}}$	Nombre d'images de la séquence-personne $\mathcal{A}^{(1)}$
$\mathcal{A}_i^{(1)}$	i^{eme} image de la séquence-personne $\mathcal{A}^{(1)}$
	$\mathcal{A}^{(1)} = \{\mathcal{A}_1^{(1)}, \dots, \mathcal{A}_{n_{\mathcal{A}^{(1)}}}^{(1)}\}$

Notations relatives aux images

I, J	Images numériques couleur
$H_{RGB}[I]$	Histogramme couleur normalisé de l'image I calculé dans l'espace couleur (R, G, B) (Pour chaque canal, les composantes couleur des pixels sont représentées sur 16 niveaux)
$H_{RGB}[I](x, y, z)$	Cellule de l'histogramme couleur normalisé $H_{RGB}[I]$ correspondant au point-couleur de coordonnées $(R = x, G = y, B = z)$ (La valeur de cette cellule est un réel compris entre 0 et 1)
$M_{rg}[I]$	Matrice de co-occurrences chromatique de l'image I, relative aux canaux r et g, échantillonnés sur 16 niveaux chacun
$M_{rg}[I](x, y)$	Cellule de la matrice de co-occurrences chromatique de l'image I, relative aux canaux r et g, échantillonnés sur 16 niveaux chacun (La valeur contenue dans cette cellule est un entier positif ou nul)
$C_{rg}[I]$	Matrice de co-occurrences chromatique **normalisée** de l'image I, relative aux canaux r et g, échantillonnés sur 16 niveaux chacun
$C_{rg}[I](x, y)$	Cellule de la matrice de co-occurrences chromatique **normalisée** de l'image I, relative aux canaux r et g, échantillonnés sur 16 niveaux chacun (La valeur de cette cellule est un réel compris entre 0 et 1)

Notations spécifiques aux vecteurs d'attributs d'images

Cas général

$\mathbf{x}[I]$	Le vecteur d'attributs d'une image I lorsqu'aucune précision supplémentaire n'est nécessaire
v	Le nombre de composantes d'un vecteur d'attributs $\mathbf{x}[I]$ lorsque ce nombre n'est pas précisé
$\mathbf{x}[I](j)$	j^{eme} composante du vecteur d'attributs $\mathbf{x}[I]$

Histogrammes

$\mathbf{x}_{H_{RGB}}[I]$	La représentation sous forme de vecteur d'attributs de l'histogramme couleur normalisé d'une image I
$v_{H_{RGB}}$	Le nombre de composantes du vecteur d'attributs $\mathbf{x}_{H_{RGB}}[I]$ (**valeur** : $16 \times 16 \times 16 = \mathbf{4096}$)

Matrices de co-occurrences

$\mathbf{x}_{C_{rg}}[I]$	La représentation sous forme de vecteur d'attributs de la matrice de co-occurrences normalisée C_{rg} d'une image I
$v_{C_{rg}}$	Le nombre de composantes du vecteur d'attributs $\mathbf{x}_{C_{rg}}[I]$ (**valeur** : $16 \times 16 = \mathbf{256}$)

Indices de texture

$\mathbf{x}_{L_{rg}}[I]$	Vecteur d'attributs formé des indices de textures calculés à partir de la matrice de co-occurrence M_{rg} d'une image I
$v_{L_{rg}}$	**valeur** : 4

Indices de complexité

$\mathbf{x}_{K_{rg}}[I]$	Vecteur d'attributs formé des indices de complexités calculés à partir des canaux r et g de l'image I
$\mathbf{x}_{K_r}[I]$	Vecteur d'attributs formé des indices de complexités calculés à partir du canal r de l'image I
$\mathbf{x}_{K_g}[I]$	Vecteur d'attributs formé des indices de complexités calculés à partir du canal g de l'image I
$v_{K_{rg}}$	**valeur** : 11
v_{K_r}	**valeur** : 11
v_{K_g}	**valeur** : 11

Signatures d'images

$\phi[I]$	signature d'une image
$\phi_C^{(r,g)}[I]$	signature d'une image par vecteurs d'attributs représentant les matrices de co-occurrences normalisées calculées à partir de l'espace couleur (r, g). $\phi_{(r,g)}^{C}[I] = \{\mathbf{x}_{C_{rr}}[I], \mathbf{x}_{C_{rg}}[I], \mathbf{x}_{C_{gg}}[I]\}$
$\phi_L^{(r,g)}[I]$	signature d'une image par vecteurs d'attributs représentant les indices de texture. $\phi_L^{(r,g)}[I] = \{\mathbf{x}_{L_{rr}}[I], \mathbf{x}_{L_{rg}}[I], \mathbf{x}_{L_{gg}}[I]\}$
$\phi_K^{(r,g)}[I]$	signature d'une image par vecteurs d'attributs représentant les indices de complexités. $\phi_K^{(r,g)}[I] = \{\mathbf{x}_{K_r}[I], \mathbf{x}_{K_{rg}}[I], \mathbf{x}_{K_g}[I]\}$

Notations relatives aux séquences d'images

S	Une séquence d'images $S = \{I_i\}$ avec $i = 1 \ldots n$ si $n = Card(S)$
$\Phi[S]$	Séquence des signatures des images de S $\Phi[S] = \{\phi[I_1], \ldots, \phi[I_n]\}$
S_{Req}	Séquence d'images requête avec $Card(S_{Req}) = n_{Req}$
S_{Can}	Séquence d'images requête avec $Card(S_{Can}) = n_{Can}$

Notations relatives aux descripteurs de séquences d'images

Descripteurs basés sur l'histogramme

$\mathbf{avg}_{H_{RGB}}[S]$	Vecteur d'attributs à $v_{H_{RGB}}$ composantes correspondant à l'histogramme moyen de la séquence d'image S
$\mathbf{med}_{H_{RGB}}[S]$	Vecteur d'attributs à $v_{H_{RGB}}$ composantes correspondant à l'histogramme médian de la séquence d'image S
$\mathbf{min}_{H_{RGB}}[S]$	Vecteur d'attributs à $v_{H_{RGB}}$ composantes correspondant à l'histogramme intersection de la séquence d'image S
$F_{H_{RGB}}[S]$	Descripteur de la séquence d'images S basé sur l'histogramme couleur $F_{H_{RGB}}[S] = \{\mathbf{avg}_{H_{RGB}}[S], \mathbf{med}_{H_{RGB}}[S], \mathbf{min}_{H_{RGB}}[S]\}$

Descripteurs basés sur d'autres vecteurs d'attributs

$F_{C_{rg}}[S]$	Descripteur de la séquence d'images S basé sur la matrice de co-occurrences normalisée relative aux canaux r et g $F_{C_{rg}}[S] = \{\mathbf{avg}_{C_{rg}}[S], \mathbf{med}_{C_{rg}}[S], \mathbf{min}_{C_{rg}}[S]\}$
$F_{L_{rg}}[S]$	Descripteur de la séquence d'images S basé sur les indices de texture relatifs aux canaux r et g $F_{L_{rg}}[S] = \{\mathbf{avg}_{L_{rr}}[S], \mathbf{med}_{L_{rg}}[S], \mathbf{min}_{L_{gg}}[S]\}$
$F_{K_{rg}}[S]$	Descripteur de la séquence d'images S basé sur les indices de complexité relatifs aux canaux r et g $F_{K_{rg}}[S] = \{\mathbf{avg}_{K_{rg}}[S], \mathbf{med}_{K_{rg}}[S], \mathbf{min}_{K_{rg}}[S]\}$

Notations relatives aux signatures de séquences d'images

Signatures par descripteurs

$\Psi_{C^{(rg)}}[S]$	signature par descripteurs de la séquence d'images S à partir des matrices de co-occurrences chromatiques $\Psi_{C^{(rg)}}[S] = \{F_{C_{rr}}[S], F_{C_{rg}}[S], F_{C_{gg}}[S]\}$ $\Psi_{L^{(rg)}}[S] = \{F_{L_{rr}}[S], F_{L_{rg}}[S], F_{L_{gg}}[S]\}$ $\Psi_{K^{(rg)}}[S] = \{F_{K_r}[S], F_{K_{rg}}[S], F_{K_g}[S]\}$

Publications

T. Leclercq, L. Macaire et J-G. Postaire. Comparaison de séquences d'images couleur par analyse en composantes principales. *Actes INRETS 98, Rencontre des laboratoires ESTAS, LEOST, LIVIC, LTN (2005)*, pages 105-117, Villeneuve d'Ascq, France, 24 février 2005.

T. Leclercq, L. Khoudour, L. Macaire et J-G. Postaire. Compact Color Video Signature by Principal Component Analysis. *In Proceedings of the International Conference on Computer Vision and Graphics (ICCVG 2004)*, pages 814-819, Warsaw, Poland, 22-24 septembre 2004.

T. Leclercq, L. Khoudour, L. Macaire et J-G. Postaire. Suivi d'objets déformables dans des séquences d'images couleur. Application à la surveillance de sites dans les transports urbains. *Actes INRETS 94, Rencontre des doctorants du LEOST (2003)*, Villeneuve d'Ascq, France, 4 février 2003.

L. Khoudour, T. Leclercq, J-L. Bruyelle, A. Flancquart. Local camera network for surveillance in Newcastle airport. *Poster session in IDSS 2003 - The IEE*, London, 26 février 2003.

Contribution au chapitre 7 de l'ouvrage : Intelligent Distributed Video Surveillance Systems, A distributed multi-sensor surveillance system for public transport applications, *IEE (Institution of Electrical Engineers) Professional applications of computing series 5*, pages 185-224, ISBN 0-86341-504-0, novembre 2005. Auteurs : J.L. Bruyelle, L. Khoudour, D. Aubert, T. Leclercq et A. Flancquart.

Introduction

J'ai intégré l'Institut National de Recherche pour les Transports et leur Sécurité (INRETS) en janvier 2001 pour occuper le poste d'Ingénieur d'Etudes au Laboratoire d'Elecronique, Ondes et Signaux pour les Transports (LEOST). Le développement d'applications destinées à la vidéosurveillance de sites de transports publics, m'a permis de découvrir le domaine du traitement d'images.

Durant cette période, j'ai également eu la possibilité de suivre le DEA d'Automatique et Informatique Industrielle à l'Université des Sciences et Technologies de Lille. Cela m'a donné l'occasion d'effectuer, au Laboratoire d'Automatique, Génie Informatique et Signal, un stage de recherche qui constituait un prolongement de mon travail d'ingénierie.

Par la suite, l'INRETS m'a octroyé une bourse d'études afin d'accomplir cette thèse, conjointement avec les laboratoires LEOST et LAGIS.

Cette collaboration s'est ensuite enrichie dès lors que nous avons eu la possibilité de travailler en collaboration avec le Laboratoire d'Informatique Fondamentale de Lille (LIFL).

Cette thèse présente notre contribution à la comparaison d'objets déformables dans des séquences d'images couleur basée sur l'analyse des composantes couleur des pixels représentant ces objets.

Dans le cadre de notre application à la problématique transports, ces objets déformables représentent des personnes filmées selon une vue de dessus lors de leur déplacement dans le champ d'observation de caméras placées dans les sites de transports publics.

Une des applications potentielles de ces travaux concerne la surveillance de différentes zones sensibles d'un site public. Une autre application est l'aide à l'exploitation et la régulation de trafic.

Dans les deux cas, la problématique est de retrouver les mêmes personnes

dans différentes séquences d'images couleur saisies par des caméras observant ces différentes zones.

Cette reconnaissance nécessite la définition d'une signature caractéristique de la personne en mouvement basée sur les propriétés spatio-colorimétriques des pixels la représentant dans les différentes images d'une même séquence. nous développons des mesures spécifiques de comparaison de ces signatures caractérisant les séquences d'images couleur.

Cette thèse s'organise en trois parties :

La première partie concerne la problématique transports traitée par l'INRETS.

Dans le chapitre 1 nous exposons les deux problématiques qui seront abordées dans le cadre de la vidéosurveillance et dans celui de l'aide à l'exploitation de sites de transports publics. La localisation automatique de personnes permet à un agent de surveillance de réagir à une situation potentiellement dangereuse une fois que celle-ci a été détectée, tandis que la constitution des matrices origine-destination permet aux opérateurs d'optimiser les flux de passagers.

Dans le chapitre 2, nous proposons un cahier des charges répondant aux besoins des exploitants. Nous montrons en quoi notre travail consistera à développer une méthode permettant de comparer le contenu d'une séquence d'images appelée requête et représentant le passage d'une personne, à une autre séquence d'images appelée candidate, qui représente également le passage d'une personne. Le but est de déterminer s'il s'agit de la même personne ou de deux personnes différentes. Nous décrivons ensuite les données sur lesquelles nous basons notre travail, leur acquisition ainsi les pré-traitements qui leurs sont appliqués. Nous définissons enfin les notions de séquence-personne et de signature.

La deuxième partie constitue un état de l'art non exhaustif de la comparaison de deux séquences d'images.

Le chapitre 3 concerne la comparaison du contenu de deux images numériques couleur. Après avoir présenté différents espace de représentation de la couleur, nous passons en revue un certain nombre de signatures d'images composées de un ou plusieurs vecteur d'attributs. La ressemblance entre deux signatures s'évalue en mesurant la proximité entre les vecteurs d'attributs homologues.

Le chapitre 4 expose différentes méthodes de comparaison du contenu de deux séquences d'images. Nous mettons d'abord en évidence les problèmes causés par

le caractère temporel inhérent aux séquences, puis nous présentons et analysons différentes signatures de séquences d'images rencontrées dans la littérature ainsi que les mesures de proximité qui leur sont appliquées.

La première approche que nous avons employée consiste à projeter les points couleurs présents dans chacune des images de la séquence candidate dans un espace de représentation spécifique à la séquence requête. Cette technique utilise les principes de l'analyse en composantes principales et permet de distinguer le cas où deux séquences représentent la même personne (les nuages de points des deux séquences projetés dans cet espace de représentation se superposent) du cas où les séquences représentent des personnes différentes (les nuages de points des deux séquences projetés dans cet espace de représentation sont bien séparés). Cette approche a fait l'objet d'une publication [LKMP04], cependant nous ne développerons pas cet aspect de notre travail.

La troisième partie concerne notre seconde approche de la comparaison d'objets déformables dans des séquences d'images couleur par analyse de signatures. Cette seconde voie est une démarche plus originale qui intègre la prise en compte de l'information couleur par des méthodes issues de la théorie de la complexité de l'information. L'hypothèse sous-jacente à cette démarche est que l'application d'algorithmes de compression réglés avec les mêmes paramètres à deux séquences similaires fournit des résultats très proches en termes de niveau de compression tandis que l'application à deux séquences différentes fournit des mesures différentes.

Le chapitre 5 concerne la comparaison de deux séquences de signatures caractérisant chacun une séquence-personne. Nous mettons d'abord en évidence les difficultés causées par la nature des données considérées, puis nous proposons une méthode de comparaison de séquences de signatures lorsque celles-ci se composent de matrices de co-occurrences chromatiques. Nous appliquons enfin cette méthode à deux séquences de signatures lorsque celles-ci se composent d'un autre type de vecteurs d'attributs calculés à partir des matrices de co-occurrences : les vecteurs d'indices de textures.

Le chapitre 6 présente la théorie algorithmique de l'information. Nous définissons la notion de complexité de description de Kolmogorov qui s'appuie sur le principe de description de longueur minimale. Nous exposons enfin les liens entre la complexité de description et la compression de données. Nous abordons ensuite

l'application de cette théorie à la comparaison de données. La distance de transformation est d'abord définie du point de vue théorique, puis nous présentons une manière de s'en approcher par compression relative. Enfin, nous mettons en évidence les limites de cette méthode lorsque celle-ci est appliquées à deux séquences-personnes.

Le chapitre 7 présente nos travaux sur la comparaison de signatures de séquences d'images basées sur un vecteur d'attributs d'image original : le "vecteur d'indices de complexités". Après avoir mis en évidence les liens entre indices de complexités et indices de textures, nous définissons la notion de vecteur d'indices de complexités obtenus par multicompression. Enfin, pour comparer deux séquences-personnes, nous appliquons aux deux séquences de vecteurs d'indices de complexités la méthodes de comparaison de deux séquences de vecteurs d'attributs que nous avons retenue au chapitre 5.

Chapitre 1

Vidéosurveillance et aide à l'exploitation

1.1 Introduction

Dans les sites de transports publics, l'intégrité des biens et des personnes, qu'il s'agisse des usagers ou du personnel d'exploitation, doit pouvoir être assurée par les exploitants. L'une des solutions envisagées pour obtenir cette sécurité est la surveillance du site au moyen de caméras.

Par ailleurs, l'optimisation de la régulation du trafic nécessite de mettre en correspondance l'offre des services de transports avec la demande des usagers. Là encore, l'utilisation de la vidéo peut constituer une aide précieuse.

Un **site de transports publics** désigne l'endroit par lequel les usagers accèdent aux moyens de transport qui sont mis à leur disposition tels que les bus, tramways, métros, trains et avions. Par la suite, lorsqu'aucune précision n'est apportée, le terme **site** désignera de manière générale un site de transports publics, qu'il s'agisse d'une gare ferroviaire, d'une station de métro ou d'un aéroport.

Les **exploitants de sites de transports publics** sont les organismes dont les activités comportent, en partie ou totalement selon le cas, la gestion des sites de transports publics.

Le terme **exploitant** désignera indifféremment les organismes publics et privés concernés par les deux soucis de sécurité et d'aide à l'exploitation relatifs aux sites de transports publics.

L'**aide à l'exploitation** consiste à permettre aux exploitants de transports publics de mieux faire coïncider l'offre des services de transports proposée avec

la demande des usagers. Il s'agit de développer des outils d'optimisation de flux de passagers transitant au sein des sites.

La **sécurité** concerne les incidents ou accidents nuisant à la bonne marche des transports et à l'intégrité des biens et des personnes. Le souci des exploitants est d'être capables de réagir efficacement et rapidement à ces différents événements.

Ce premier chapitre décrit la problématique liée à l'utilisation de la vidéosurveillance appliquée à l'aide à l'exploitation et à la sécurité des sites de transports publics.

Dans le premier paragraphe, nous analysons les besoins des exploitants des sites de transports publics, d'une part en termes d'aide à l'exploitation à travers la notion de **matrice origine-destination** et d'autre part en termes de sécurité à travers la notion de **situation potentiellement dangereuse**.

Dans le deuxième paragraphe, nous présentons un exemple d'infrastructure de vidéosurveillance existant aujourd'hui : le système de vidéosurveillance de la gare *Lille-Flandre*. Nous y découvrons également les limites de cette technique, telle qu'elle est exploitée à l'heure actuelle.

Dans le troisième paragraphe, nous décrivons les travaux antérieurs réalisés par l'INRETS[1] pour détecter automatiquement des situations potentiellement dangereuses grâce à des logiciels spécifiques développés dans le cadre de programmes de recherche.

Dans le quatrième paragraphe, nous décrivons un prototype de système distribué de vidéosurveillance qui regroupe ces fonctionnalités. Ce système permet à un agent de surveillance d'être alerté dès qu'une situation potentiellement dangereuse a été automatiquement détectée.

Enfin, dans le cinquième paragraphe, nous identifions précisément les deux fonctionnalités qu'il serait souhaitable d'intégrer à ce système pour répondre aux nouveaux besoins des exploitants, et qui feront l'objet de nos travaux.

1.2 Analyse des besoins des exploitants

Grâce à des échanges [La96][HKa01] entre les chercheurs de l'INRETS et les responsables d'organismes d'exploitation de transports publics comme la SNCF ou TRANSPOLE, nous avons pu identifier et analyser deux besoins majeurs aux-

[1] Institut National de Recherche sur les Transports et leur Sécurité

quels la vidéosurveillance peut apporter des solutions : l'aide à l'exploitation et la gestion des situations potentiellement dangeureuses.

1.2.1 Aide à l'exploitation et régulation de trafic

D'une manière générale, l'aide à l'exploitation d'un site de transports publics consiste à optimiser la régulation du trafic au sein de ce site. La régulation du trafic peut être optimisée, non seulement en mettant en adéquation l'offre de transport et la demande des usagers, mais aussi en s'adaptant rapidement aux évolutions de cette demande. Par exemple, l'identification des heures de grande affluence sur une ligne de métro permet aux organismes de transports de mettre à la disposition des usagers des rames supplémentaires afin d'éviter des situations de congestion.

Les applications de notre travail se concentrent sur l'aide à l'exploitation d'une plateforme multimodale. Une plateforme multimodale est un site de transports publics où convergent les différentes lignes de transports mises à la disposition des usagers, à savoir lignes de trains, de bus, de métros et de tramways. Afin d'exploiter au mieux ces lignes de transports, il est nécessaire d'établir la matrice origine - destination qui contient les informations sur la répartition des flux entrants et des flux sortants entre les différentes lignes. Par exemple : pour 100 personnes provenant de la ligne de métro n^o 2 entre 8 et 10 heures, 10 personnes ont pris la ligne TER n^o 4675, 20 personnes ont pris le bus n^o 7, 10 personnes ont pris le tramway n^o 3 et enfin 60 personnes ont terminé ici leur trajet et sont sorties du site.

A l'heure actuelle, ces matrices origine - destination sont constituées à partir de sondages ponctuels effectués par des enquêteurs auprès des usagers. Les exploitants de transports ont émis le besoin de disposer d'un outil permettant de constituer automatiquement ces matrices, dans le but d'obtenir des informations complètes et exhaustives sur le trafic des passagers au sein d'une plate-forme multimodale.

1.2.2 Gestion des situations potentiellement dangereuses

Le second besoin identifié porte sur la détection et la gestion des situations potentiellement dangeureuses apparaissant au sein d'un site de transports pu-

blics. Ces situations regroupent les événements pouvant être liés à un groupe de personnes ou à une personne en particulier, dont les conséquences risquent de dégrader la qualité de fonctionnement des transports ou de porter atteinte à l'intégrité des biens et des personnes.

1.2.2.1 Situations liées à un groupe de personnes

Le premier cas de situations potentiellement dangereuses concerne le comportement anormal d'un groupe de personnes. Tout mouvement inhabituel de la foule est le signe d'un événement particulier [YV95].

Un déplacement particulièrement lent des usagers dans un couloir peut être causé par un attroupement, ce dernier pouvant créer une situation de congestion ; un déplacement particulièrement rapide des usagers dans ce même couloir peut résulter d'un accident comme un début d'incendie ; enfin un comportement désorganisé de la foule dans un hall peut résulter d'une altercation ou d'une bagarre.

1.2.2.2 Situations liées à une personne en particulier

Le deuxième besoin est relatif à la gestion des situations potentiellement dangereuses liées au comportement anormal d'une seule personne. Ce comportement est identifié sans ambiguité lorsque, par exemple, une personne s'introduit dans une zone interdite ou agresse une autre personne.

Par contre, le comportement d'une personne est ambigu lorsque, par exemple, celle-ci s'arrête dans une zone de passage : il peut s'agir d'une personne cherchant son chemin, tout comme d'une personne se livrant à une activité illicite comme la revente de stupéfiants.

Une personne peut également se déplacer à contre-sens dans un couloir, ce qui constitue un comportement suspect. Toutefois, il peut simplement s'agir d'une personne qui s'est trompée de chemin.

Enfin, lorsqu'une personne se déplace à une vitesse anormalement élevée, il peut s'agir d'une personne venant de commettre un délit, ou d'une personne désireuse d'emprunter un train, une rame de métro, un bus ou un tramway sur le point de partir.

L'étendue de la plupart des sites de transports publics rend difficile et très coûteuse leur surveillance in-situ par un ou même plusieurs agents de sécurité. Les organismes de transports ont donc largement développé l'utilisation de la vidéosurveillance.

1.3 Un exemple de dispositif de vidéosurveillance

Plutôt que d'effectuer une étude de synthèse sur les différents systèmes de vidéosurveillance installés dans des sites de transports publics, nous avons choisi de décrire le système installé sur le site de la gare *Lille-Flandre*.

Les infrastructures matérielles de vidéosurveillance dans les transports publics reposent aujourd'hui sur des caméras qui ont été disposées à des endroits stratégiques du site, tels que les halls, les couloirs et les escaliers. Les images acquises par les caméras sont transmises à des écrans de contrôle situés dans un Poste de Commande Centralisée (PCC). Dans ce poste, un agent surveille le site en observant les différents écrans de contrôle.

1.3.1 L'ensemble des caméras

La vidéosurveillance repose aujourd'hui sur un ensemble de caméras placées à des endroits stratégiques du site. Dans un lieu fermé, tel que la station de métro ou la gare, les caméras sont placées au bord des voies, dans les couloirs où le flux de passagers est important, à des points de carrefour et à l'entrée de zones interdites. D'autres caméras observent les entrées et sorties du site. Cette couverture télévisuelle est, à l'heure actuelle, l'un des dispositifs utilisés par les autorités locales et les forces de police pour assurer la sécurité des biens et des personnes. Ces caméras acquièrent des images qui sont transmises vers des écrans de contrôle situés dans un Poste de Commande Centralisée (PCC).

1.3.2 Le Poste de Commande Centralisée (PCC)

Le Poste de Commande Centralisée est un lieu dans lequel un agent de surveillance examine les images transmises par les caméras réparties sur le site.

La figure 1.1 représente l'architecture du système de vidéosurveillance actuellement installé dans la gare *Lille-Flandre*. L'agent sélectionne, parmi l'ensemble des caméras, celles qui seront reliées à chacun des moniteurs. Ainsi, l'agent observe plusieurs écrans simultanément, sans pouvoir réellement focaliser son attention en permanence sur l'un d'eux.

Le nombre d'écrans étant inférieur au nombre de caméras disponibles, l'agent ne dispose à chaque instant que d'une représentation partielle des zones observées. S'il veut être efficace, l'agent doit fréquemment modifier la répartition des caméras

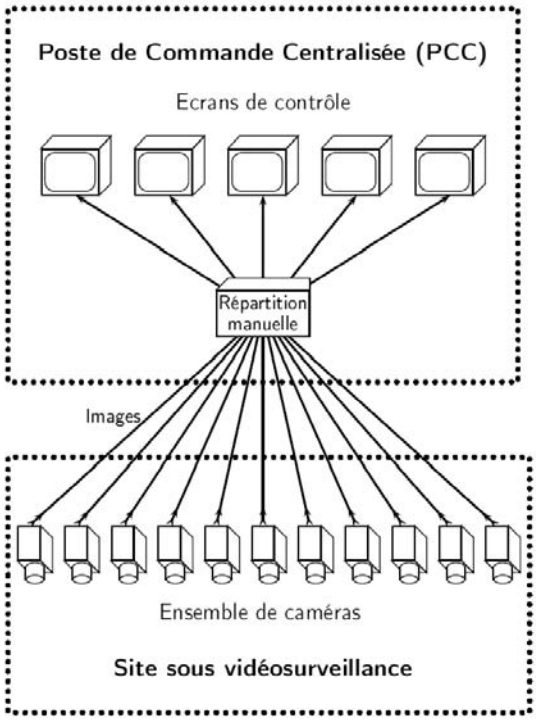


FIG. 1.1 – *Architecture du système de vidéosurveillance actuel installé à la gare de Lille-Flandre.*

vers les écrans de contrôle afin de pouvoir surveiller l'ensemble des zones couvertes par la vidéosurveillance.

1.3.3 Les insuffisances du système

La configuration la plus répandue est le cas où l'agent de surveillance observe des images pouvant provenir d'une centaine de caméras connectées à un nombre d'écrans vidéo variant entre 5 et 10 (voir figure 1.2).

Des recherches, comme celles menées par le ministère britannique de l'intérieur [HKa01], ont montré que l'attention de l'agent de sécurité s'atténue au bout d'une trentaine de minutes et que ce dernier devient généralement "aveugle" aux événements. Par conséquent, il est probable qu'il ne soit pas toujours capable de détecter des situations potentiellement dangereuses.

L'architecture actuelle du système de vidéosurveillance installé à la gare Lille-Flandre ne permet donc pas de répondre parfaitement aux besoins des exploitants

de transports publics.

1.4 Détection de situations potentiellement dangereuses

Compte-tenu de l'importante charge de travail imposée à l'agent de surveillance, il est essentiel d'éviter la surcharge visuelle à laquelle il est actuellement exposé. Au lieu de contraindre celui-ci à examiner successivement toutes les scènes observées par les caméras du site, il serait opportun de n'afficher que les images des scènes représentant un intérêt particulier. Pour ce faire, il est nécessaire de disposer d'un système d'analyse d'images qui détecte automatiquement les incidents et accidents ainsi que les situations potentiellement dangereuses.

Le système installé dans la gare Lille-Flandre est un système classique de vidéosurveillance de sites de transports publics. Avec l'accord du gouvernement Français, les exploitants du site ont souhaité le développement de projets de recherche portant sur la détection automatique de situations potentiellement dangereuses, par l'analyse des images acquises avec les caméras installées sur le site.

Grâce à leur participation à plusieurs projets de recherche, les chercheurs de l'INRETS ont développé des algorithmes de détection de situations potentiellement dangereuses que nous présentons dans un premier temps.

FIG. 1.2 – *Le poste de travail d'un agent de sécurité, et ses nombreux écrans.*

1.4.1 Algorithmes de détection automatique

La vidéosurveillance fait l'objet de nombreux travaux depuis plusieurs années, et régulièrement des revues lui consacrent des volumes entiers [MT00, CLK00, MT04]. Les chercheurs s'appliquent à concevoir des algorithmes capables de détecter des événements particuliers à partir de l'analyse des images acquises par des caméras de surveillance [Pra03]. Les situations potentiellement dangereuses citées précédemment sont un exemple d'événements qu'il s'agit de détecter automatiquement.

Dans le cadre des projets *CROwd Management with Telematic Imaging and Communication Assistance* (CROMATICA) [VAB+98] du 4ème PCRDT[2] et *Proactive Integrated Systems for Security Management by Technological, Institutional and Communication Assistance* (PRISMATICA) [VKa02] du 5ème PCRDT, l'INRETS a développé différents algorithmes d'analyse de scènes visant à détecter automatiquement certaines situations potentiellement dangereuses que nous présentons ici.

Bien que ces algorithmes ont été testés dans des conditions réelles[3] [Kho03], ils n'ont pas, à ce jour, été mis en ligne au sein des systèmes de vidéosurveillance des sites de transports publics.

1.4.2 Exemples de situations détectées

1.4.2.1 Situation de *stationnarité*

La détection de stationnarité [KDB00][DVD96] d'une personne dans une zone de passage se fait par analyse de la différence entre les images acquises et une image de référence représentant la scène. Parmi les parties de l'image considérée qui sont différentes de celles de l'image de référence, l'algorithme distingue celles qui représentent une personne ou un objet en mouvement de celles qui sont statiques. Ces zones statiques représentent alors une personne qui s'est arrêtée ou un objet qui a été déposé, comme l'illustre la figure 1.3.

[2] Programme Cadre pour la Recherche et le Développement Technologiques en Europe

[3] Le projet PRISMATICA comprenait une validation sur site réel dans l'aéroport de Newcastle (Ecosse).

FIG. 1.3 – *Détection de stationnarité dans une zone de passage (en rouge).*

1.4.2.2 Situation de déplacement *à contresens*

La détection d'une personne se déplaçant à contresens [KDB00][DVD96] dans un couloir à sens unique est basée sur l'approche "flot optique" de Horn et Shunk [HS81] qui estime les directions des mouvements. L'algorithme implémenté permet de calculer les vecteurs de mouvement d'une personne et, en fonction de la direction de ces vecteurs, d'indiquer ainsi le sens de déplacement de cette personne [W.90]. Un seuil sur le nombre de vecteurs de mouvement considérés en contresens peut être fixé : ainsi le système peut être configuré pour ne déclencher une alarme que lorsque plusieurs personnes se déplacent simultanément en contresens.

1.4.2.3 Situation d'*intrusion* ou de chute sur les voies

La détection d'intrusions en zones interdites et de chutes sur les voies se fait grâce à un algorithme spécifique utilisé pour détecter les contours des objets en mouvement dans l'image [Vie88]. Un processeur câblé [Cab92] autorise un traitement à la fréquence d'acquisition de 25 images par seconde.

Cet algorithme est basé sur l'analyse des différences de luminance entre pixels de mêmes coordonnées dans des images successives de la séquence vidéo originale. Il consiste essentiellement en une extraction de contours sur les différences d'images successives afin de ne retenir que les contours des objets mobiles dans l'image. Une amélioration de cette méthode [Ka99] utilise trois images successives pour éliminer des artefacts apparaissant lorsque le fond de la scène est fortement

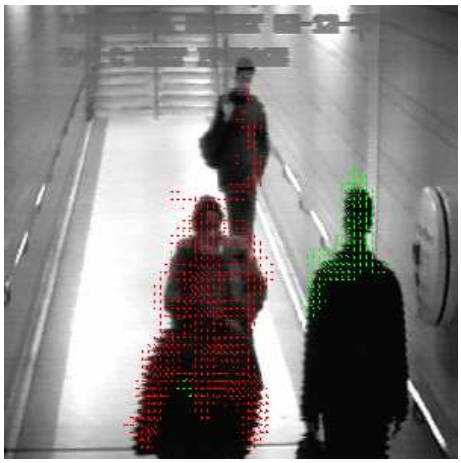


FIG. 1.4 – *Détection d'individus se déplaçant à contresens (en rouge). L'individu en vert se déplace dans le bon sens.*

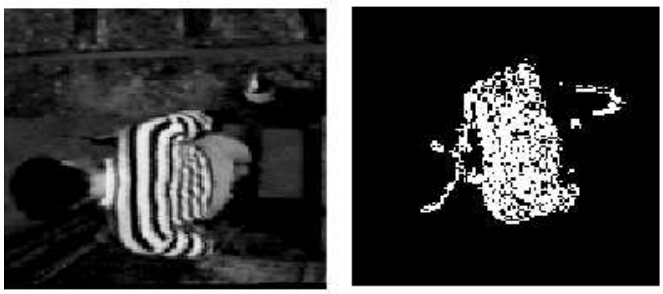


FIG. 1.5 – *Détection de chute sur une voie*

texturé. La figure 1.5 montre une image extraite de la séquence d'images originale et le résultat de la détection.

1.5 Système distribué de vidéosurveillance

L'architecture matérielle et logicielle d'un système distribué incluant le PCC et des caméras est ici décrite, à travers un exemple de détection d'intrusions en zones interdites.

Ces algorithmes de détection de situations potentiellement dangereuses ont été intégrés à un système développé à l'INRETS, appelé système de caméras intelligentes, et décrit dans [Kho03]. Ce terme désigne un cas particulier de ce que les anglo-saxons appellent *Intelligent Distributed Systems* [Vel05, VV05].

1.5.1 Systèmes distribués de caméras intelligentes

Un système est dit distribué lorsque les traitements ou calculs qui lui sont nécessaires ne sont pas effectués de manière centralisée (c'est-à-dire sur une seule unité de calculs) mais sont distribués (répartis) sur plusieurs unités de calculs reliées par un moyen de communication tel qu'un réseau ethernet.

Ce système est dit intelligent lorsque ces unités sont capables de coopérer grâce à un protocole de communication de haut niveau [MS05]. L'organisation qui résulte de la communication entre les différents programmes s'exécutant sur les différentes unités de calculs constitue alors un système multi-agents, paradigme décrit par Weiss [Wei99], dans lequel des entités (les agents) effectuent une tâche collective, basée sur la communication et la coopération, tout en étant capables de prendre des initiatives individuelles.

Ainsi, la vidéosurveillance s'est naturellement orientée vers les systèmes distribués [VV05, Vel05], et plus particulièrement la vidéosurveillance dans le domaine des transports publics [LSV03]. L'ouvrage [RJPR02] dresse un panorama complet des enjeux et des techniques relatives aux systèmes de surveillance distribués. Enfin, Desurmont et al. témoignent de l'intérêt de cette approche [DBC+05] puisqu'il s'appliquent à développer, en vue de le commercialiser, un tel système de vidéosurveillance qui se veut générique[4], donc aisément réutilisable.

[4] Facilement personnalisable selon les besoins exacts du client.

1.5.2 Description du système développé à l'INRETS

Le système distribué que j'ai réalisé au Laboratoire LEOST[5], dans le cadre du projet PRISMATICA est mentionné dans [LSV03, VLS03, VVS03, VK04]. Il intègre de nombreuses fonctionnalités qui ne seront pas développées ici. Ma contribution en tant qu'ingénieur d'études concerne une partie de ce système, appelé **système de caméras intelligentes**, qui a également été décrit en détail dans [Kho03]. Nous proposons d'en rappeler ici les principes.

Le terme **caméra intelligente**, désigne une caméra couplée à une unité de calcul de type PC munie d'une carte d'acquisition vidéo et d'une carte réseau ethernet. Il est ainsi possible de déporter des algorithmes d'analyse d'images sur la caméra intelligente, afin de satisfaire les contraintes de traitement à la cadence d'acquisition des images vidéo. Nous avons implanté les algorithmes de détection de situations potentiellement dangereuses précédemment cités.

Une fois qu'une situation potentiellement dangereuse a été détectée par une caméra intelligente, celle-ci prend l'initiative de transférer cette information vers le PCC à destination de l'agent de sécurité, par l'intermédiaire d'un programme superviseur, lui aussi connecté au réseau ethernet. Ce programme superviseur remplace le répartiteur que l'agent de surveillance doit actionner lorsqu'il désire visualiser le contenu d'une scène observée par une caméra donnée.

Cependant, le superviseur remplit toujours la fonction de répartiteur : il laisse à l'agent de surveillance la possibilité d'utiliser le système comme il le désire, c'est-à-dire de visionner, à sa demande, la scène observée par la caméra de son choix. La fonction supplémentaire consiste à transmettre à l'agent de sécurité les informations reçues de la part des caméras intelligentes réparties sur le site uniquement lorsqu'une situation potentiellement dangereuse a été détectée.

La figure 1.6 présente le système de détection automatique d'intrusion en zone interdite développé à l'INRETS. Chaque caméra observe une zone interdite au public. Comme dans le cas du système de vidéosurveillance classique, l'agent de surveillance peut choisir de surveiller la scène observée par la caméra de son choix, sur un des écrans de surveillance. Sur ce schéma, il utilise trois des cinq écrans à sa disposition.

[5] Laboratoire d'Electronique, Ondes et Signaux pour les Transports

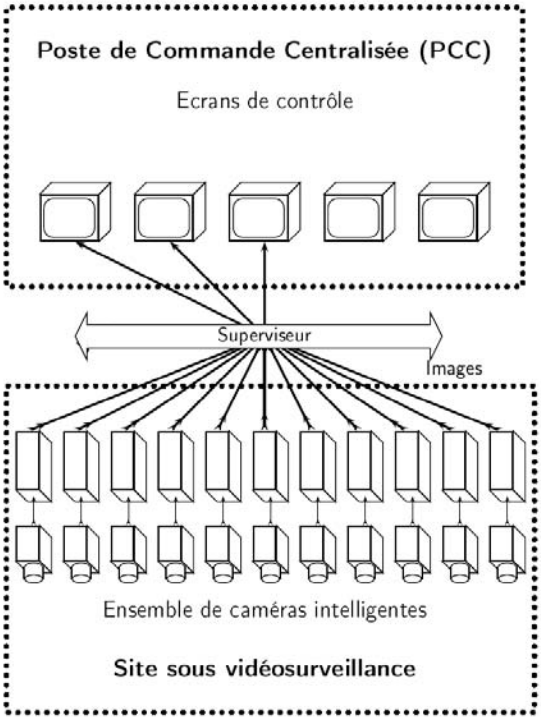


FIG. 1.6 – *Système de détection automatique de situations potentiellement dangereuses développé par l'INRETS.*

La figure 1.7 illustre le principe de fonctionnement du système de détection automatique : lorsqu'une caméra intelligente détecte une situation potentiellement dangereuse (ici une intrusion dans une zone interdite), elle envoie au programme superviseur situé dans le PCC les informations relatives à l'identification de la caméra dans le site. Le programme superviseur déclenche alors directement sur un écran de contrôle, dédié à cet effet, l'affichage de la séquence d'images correspondant à la situation potentiellement dangereuse détectée et indique l'emplacement de la scène observée. De plus, un signal sonore est émis pour avertir l'agent de surveillance.

1.6 Les deux nouvelles fonctionnalités souhaitées

A partir d'un système distribué tel que celui qui est développé à l'INRETS, les opérateurs souhaitent disposer de deux nouvelles fonctionnalités répondant à

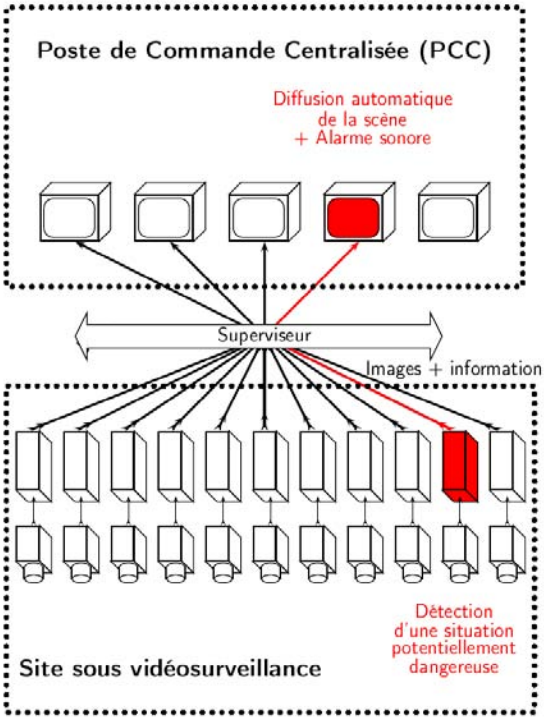


FIG. 1.7 – *Gestion automatique d'une situation potentiellement dangereuse détectée par une des caméras intelligentes.*

leurs besoins en termes d'aide à l'exploitation et en termes de sécurité.

1.6.1 Localisation automatique d'une personne

Quand l'une des caméras intelligentes du système de vidéosurveillance a détecté une situation potentiellement dangereuse provoquée par une personne, l'agent de surveillance désire localiser cette personne au sein du site à un instant donné. Pour ce faire, il est nécessaire de disposer d'un algorithme capable d'analyser les images représentant le passage d'une personne dans le champ d'observation d'une caméra intelligente et capable de déterminer s'il s'agit de la personne que l'agent de surveillance désire localiser.

1.6.2 Création automatique des matrices origine-destination

Par ailleurs, un tel système de localisation automatique de personnes peut également fournir un outil d'aide à l'exploitation en contribuant à la création

automatique des matrices origine-destination. En effet, pour une ou plusieurs personnes présentes dans le site, leur localisation automatique peut permettre d'associer l'endroit par lequel elles sont entrées dans le site avec l'endroit par lequel elles en sont sorties, tout en fournissant la date et l'heure de ces passages.

1.7 Conclusion

La sécurité des usagers se déplacant dans les sites de transports publics revêt une importance particulière. Les systèmes de surveillance, tels qu'ils ont été développés dans les projets de recherche CROMATICA et PRISMATICA, sont capables de détecter automatiquement des situations dangereuses ou potentiellement dangereuses.

Notre thèse a pour premier objectif de contribuer à la conception d'un système qui est capable, à la demande d'un agent de sécurité, de localiser automatiquement une personne au sein d'un site public suite à la détection automatique d'une situation potentiellement dangereuse.

Le second objectif est de contribuer à la conception d'un second système capable de créer automatiquement les matrices origine-destination dans le cadre de l'aide à l'exploitation des sites de transports publics.

Chapitre 2

Cahier des charges et acquisition des données

2.1 Introduction

Dans ce chapitre, nous proposons un système de localisation automatique d'une personne se déplacant dans un site de transports publics. Nous expliquons en quoi ce système peut être utilisé pour répondre aux besoins des exploitants, en termes de sécurité et d'aide à l'exploitation.

Dans le premier paragraphe, nous détaillons le cahier des charges de ce système, rédigé pour répondre aux besoins des exploitants. Nous y intégrons les deux scénarii envisagés : la localisation automatique d'une personne et la constitution des matrices origine-destination. Nous précisons également les différentes contraintes qui devront être prises en compte lors de la réalisation.

Dans le deuxième paragraphe, nous montrons que notre travail consiste à développer une méthode de comparaison de séquences d'images couleur qui sera intégrée dans le système formé d'un ensemble de caméras intelligentes et d'un superviseur. Nous montrons en quoi, dans le contexte de nos travaux, la reconnaissance d'objets déformables dans des séquences d'images couleur peut être effectuée en comparant les contenus de ces séquences par l'intermédiaire de signatures. Nous définissons cette notion de signature puis expliquons la façon dont ces signatures peuvent être utilisées par un système de localisation de personnes et lors de la constitution des matrices origine-destination.

Dans le troisième paragraphe, nous proposons de revenir plus en détails sur les séquences d'images considérées : nous décrivons les conditions d'acquisition

et les pré-traitements qui leur son appliqués, ainsi que leurs particularités. Nous terminons par la définition d'une séquence-personne qui constituera le point de départ de nos travaux.

2.2 Cahier des charges fonctionnel découlant de l'analyse des besoins

Le cahier des charges fonctionnel du système que nous proposons se décompose en quatre volets.

Le premier est constitué de deux scénarii : l'un relatif à la sécurité et l'autre à l'aide à l'exploitation. Le deuxième volet porte sur les contraintes liées au respect des libertés individuelles et celles liées à l'infrastructure matérielle. Le troisième volet concerne les performances que le système doit atteindre afin de pouvoir être exploité pour la localisation d'une personne ayant provoqué une situation potentiellement dangereuse ou pour constituer automatiquement des matrices origine-destination. Enfin, dans le quatrième volet de ce cahier des charges, nous proposons quelques hypothèses simplificatrices.

2.2.1 Scénario - Sécurité

Une fois qu'une situation potentiellement dangereuse a été détectée, le système doit localiser la personne qui est à l'origine de cette situation suspecte. Pour ce faire, lorsque cette personne traverse le champ d'observation de l'une des caméras surveillant le site considéré, l'agent de surveillance doit en être informé par l'intermédiaire du programme superviseur.

Nous supposons ici que l'agent de sécurité dispose de la séquence d'images correspondant à la situation potentiellement dangereuse qui vient d'être automatiquement détectée.

La localisation automatique s'effectue à la demande de l'agent de sécurité lorsque celui-ci décide de surveiller le déplacement d'une personne qui semble être à l'origine de la situation potentiellement dangereuse détectée.

Nous supposons également que, lors du fonctionnement en conditions réelles, les séquences d'images observées ont été pré-traitées : si une ou plusieurs personnes sont présentes dans la scène, chaque image est décomposée en autant de parties distinctes qu'il y a de personnes. La sélection dans la séquence d'images

de la personne à identifier ne fait pas l'objet de notre étude. Nous supposons enfin que l'on dispose de séquences d'images représentant uniquement la personne que l'agent de surveillance désire localiser. De telles séquences seront désignées par le terme **séquences requête**.

Si la personne surveillée repasse dans le champ d'observation d'une autre caméra que celle qui a détecté la situation potentiellement dangereuse, le système doit automatiquement déterminer à partir de l'analyse de la nouvelle séquence d'images, qu'il s'agit bien de cette personne et communiquer à l'agent de surveillance les informations relatives à la présence ou au déplacement de la personne dans la zone surveillée (emplacement de la caméra, date et heure de l'identification).

Le système doit enfin permettre à l'agent de surveillance de visualiser la séquence d'images représentant la personne se déplaçant dans le champ d'observation de la caméra identifiée. Ainsi, lorsque l'agent de surveillance effectue une demande de localisation d'après une séquence requête, chaque séquence représentant le passage d'une personne dans le champ de vision d'une des caméras du site est qualifiée de **séquence candidate**.

La localisation peut parfois échouer et confondre la personne recherchée avec une autre personne, si cette dernière est suffisamment ressemblante. Ainsi, lorsque le système a localisé la personne, l'agent de surveillance doit pouvoir vérifier sur l'écran qu'il s'agit bien de la personne qu'il désire surveiller. Dans le cas contraire, il ne valide pas cette localisation et le système fournit alors une autre localisation.

2.2.2 Scénario - Aide à l'exploitation

La demande faite au système de vidéosurveillance est, cette fois, différente de celle de la localisation d'une personne en particulier en cas de détection de situations potentiellement dangereuses.

Nous supposons ici que le système de localisation dispose de toutes les séquences d'images représentant chacune des personnes qui entrent dans le site, que ce soit par l'une des entrées destinées aux piétons ou par l'une des voies de sorties d'une des lignes de transports (trains, métros, bus, etc.). Il y a donc, cette fois, plusieurs séquences requêtes.

Le système dispose également de toutes les séquences d'images représentant

chacune des personnes qui quittent le site, soit par une des sorties, soit parce qu'elles empruntent une des lignes de transports. Ces séquences sont des séquences candidates.

Le but du système souhaité est de mettre en correspondance chaque séquence d'images requête avec la séquence candidate qui représente la même personne. Le résultat de cette mise en correspondance permet de mettre à jour automatiquement la matrice origine-destination.

2.2.3 Contraintes liées au respect des libertés individuelles

Dans le contexte de la surveillance des lieux publics, au sens large, la Commission Nationale de l'Informatique et des Libertés (CNIL)[1] veille au respect des libertés individuelles et interdit que les images des visages soient stockées sur un support matériel ou servent à des fins de reconnaissance.

Afin de respecter la loi n^o 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés[2], et par extension au respect de la confidentialité, les scènes surveillées seront observées par des caméras ne fournissant que des vues de dessus. Les images recueillies représentent donc essentiellement le dessus de la tête, les épaules et le haut du buste de chaque personne, ainsi que les mouvements alternatifs des bras et des jambes.

2.2.4 Contraintes liées à l'infrastructure matérielle

Certaines contraintes matérielles limitent également le choix de l'emplacement des caméras de surveillance. En effet, il n'est pas toujours possible de positionner une caméra de manière idéale dans un site, notamment pour des raisons de coûts,

[1] Créée par la loi n^o 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés, la CNIL est une autorité administrative indépendante chargée de veiller à la protection des données personnelles.

[2] Voir également les textes modifiant la loi : Loi n^o 88-227 du 11 mars 1988 (Journal officiel du 12 mars 1988) ; Loi n^o 92-1336 du 16 décembre 1992 (Journal officiel du 23 décembre 1992) ; Loi n^o 94-548 du ler juillet 1994 (Journal officiel du 2 juillet 1994) ; Loi n^o 99-641 du 27 juillet 1999, (Journal officiel du 28 juillet 1999) ; Loi n^o 2000-321 du 12 avril 2000, (Journal officiel du 13 avril 2000) ; Loi n^o 2002-303 du 4 mars 2002, (Journal officiel du 5 Mars 2002) ; Loi n^o 2003-239 du 18 mars 2003 (Journal officiel du 19 mars 2003) ; Loi n^o 2004-801 du 6 août 2004 (Journal officiel du 7 août 2004).

d'accessibilité ou de maintenance. Les caméras seront donc plutôt placées au "meilleur emplacement possible".

De plus, les exploitants souhaitent que le système de localisation exploite les caméras déjà installées sur le site. Cependant plusieurs types différents de caméras équipent le site, à savoir différents modèles de résolutions différentes et munis d'objectifs de différentes distances focales.

Puisque l'emplacement des caméras est tributaire de ces contraintes, l'analyse automatique de séquences d'images doit être indépendante de certaines conditions d'acquisition comme la distance focale de l'objectif de la caméra, la direction d'observation qui ne sera pas toujours parfaitement verticale, et la distance séparant la caméra du sol.

2.2.5 Performances requises par une utilisation en ligne

Dans le cas de la localisation d'une personne à travers un réseau de caméras, chaque personne passant sous chaque caméra du site est potentiellement une personne recherchée. Il est donc nécessaire d'obtenir du système une réponse dont le temps de latence n'excède pas ou peu la durée du passage de la personne dans le champ d'observation de la caméra.

Dans le cas de la constitution des matrices origine-destination, il s'agit de retrouver pour chaque personne entrant dans le site, la séquence d'images acquise ultérieurement représentant la même personne sortant du site. Le nombre de mises en correspondance à effectuer est alors égal au nombre de personnes ayant fréquenté le site, le nombre de cas à examiner peut s'avérer être très important.

2.2.6 Hypothèses simplificatrices

Dans le cadre de cette étude, nous nous plaçons dans le cas simple où une seule personne passe dans le champ d'observation de chaque caméra. Les cas plus délicats, tels que la présence de deux personnes pouvant éventuellement entrer en contact ou les flux de passagers plus denses, sortent du cadre de ce travail puisque ces situations, pour être exploitables, nécessitent un pré-traitement des images afin de séparer ces personnes, comme nous l'avons déjà indiqué précédemment.

De plus, le passage de la personne se fait ici dans le sens de la largeur du champ de vision, selon une trajectoire rectiligne, afin de simuler les conditions de dépla-

cement dans un couloir, qui représente l'exemple le plus simple de lieu de passage.

Grâce à l'élaboration du cahier des charges fonctionnel reflétant les besoins des organismes de transports, des scénarii et des contraintes ont pu être définis. Il s'agit maintenant de présenter le cahier des charges technique relatif à la conception du système de localisation automatique de personnes et de traduire les attentes en termes d'objectifs.

2.3 L'approche proposée

La plupart des sites de transports publics sont aujourd'hui équipés de réseaux de caméras de surveillance. Il est possible de modifier un ensemble de caméras en un réseau de caméra intelligentes, en couplant chaque caméra à une unité de calcul. Nous avons ainsi rédigé un cahier des charges technique traitant de la localisation automatique de personnes basée sur l'analyse de séquences d'images, énonçant d'un point de vue technique les performances requises par une utilisation en ligne.

Afin de rendre possible la localisation automatique d'une personne à travers ces réseaux de caméras intelligentes, qu'il s'agisse de surveillance ou d'aide à l'exploitation, nous avons défini une démarche commune dont la problématique consiste à **reconnaître des objets déformables dans des séquences d'images couleur**. Nous précisons ici les raisons pour lesquelles nous considérons que cette reconnaissance s'appuie sur la comparaison du contenu de ces séquences d'images.

Nous expliquons ensuite pourquoi la comparaison du contenu de deux séquences d'images ne se fait pas directement, mais par l'intermédiaire de la notion de signatures.

Nous détaillons enfin la manière dont seront utilisées ces signatures, de leur création à partir des séquences d'images acquises par les caméras jusqu'à la mesure de ressemblance entre deux signatures, pour aboutir, le cas échéant, à la transmission d'une information vers le PCC.

2.3.1 Comparaison du contenu de deux séquences d'images couleur

Chaque caméra acquiert les images destinées à l'unité de calcul correspondante à la fréquence de 25 images par seconde. Lorsqu'une personne passe dans le champ d'observation d'une telle caméra, nous supposons qu'il est possible d'isoler, à partir du flot des images successives, la séquence composée des seules images représentant cette personne.

Notre démarche consiste à comparer le contenu d'une séquence d'images dite **séquence requête**, représentant le passage de la personne à identifier, au contenu de séquences d'images dites **séquences candidates**, chacune représentant le passage d'une personne dans le champ d'observation d'une autre caméra. Si deux séquences représentent la même personne, elles seront dites **similaires**, dans le cas contraire, on parlera de séquences **différentes**.

Dans le cas de la localisation automatique, lorsque la personne à localiser traverse le champ d'observation d'une caméra intelligente autre que la caméra requête, le superviseur doit déterminer si la séquence candidate est similaire ou non à la séquence requête. Dans le cas de l'aide à l'exploitation, le système doit déterminer, pour chaque séquence requête, la séquence candidate qui lui est similaire.

Dans les deux cas, cette information est obtenue en appréciant la ressemblance entre la séquence requête et chacune des séquences candidates. Ces séquences candidates sont alors triées par ordre décroissant d'une mesure de ressemblance avec la séquence requête. Le superviseur considère que la séquence candidate qui ressemblance le plus à la séquence requête a de fortes chances d'être similaire à cette séquence requête.

2.3.2 Signatures de séquences d'images

Dans le cadre de la localisation de personnes comme dans celui de la constitution de matrices origine-destination, notre problématique consiste à reconnaître des personnes en déplacement qui peuvent être considérées comme des objets déformables présents dans des séquences d'images couleur acquises par différentes caméras et représentant chacune le passage d'une seule personne.

Pour ce faire, nous avons proposé de comparer deux-à-deux les séquences d'images afin de déterminer si leurs contenus se ressemblent ou non, autrement dit, s'il s'agit de la même personne représentée dans les deux séquences considérées.

Il s'agit maintenant de disposer d'outils qui permettent de quantifier cette ressemblance entre deux séquences d'images : la comparaison s'effectue alors en mesurant une grandeur scalaire positive à l'aide d'une *fonction de comparaison*.

Cependant, il n'existe pas de méthode efficace pour mesurer directement la ressemblance entre le contenu de deux images ou de deux séquences d'images. Cette mesure s'effectue par l'intermédiaire de ce que nous appelons des **signatures**.

Pour le moment, nous retenons simplement que la signature d'une séquence d'images contient des informations numériques qui caractérisent son contenu[3]. Nous reviendrons plus en détail sur la composition de différentes signatures dans les chapitres suivants, et nous nous contentons ici d'expliquer comment les utiliser.

Nous retrouvons très fréquemment dans la littérature la notion de signature, comme dans le système de recherche d'images PicToSeek [GS99c] de Gevers et Smeulders. Ce système indexe automatiquement et de manière hors-ligne les images d'une base de données, selon des critères visuels. Il recherche ensuite, en ligne, les images de la base qui sont similaires à une image requête, fournie par l'utilisateur, et les présente classées selon les valeurs d'un critère de ressemblance entre les signatures des images candidates et la signature de l'image requête.

Notre travail s'oriente ici vers l'étude de différentes signatures de séquences d'images et leur application à la comparaison de séquences. Pour ce faire, nous allons nous intéresser aux différentes méthodes de caractérisation du contenu d'une séquence d'images, et plus particulièrement aux différents critères visuels sur lesquels s'appuient la comparaison.

La seule information de luminance contenue dans les images en niveaux de gris ayant montré ses limites pour caractériser et discriminer des objets [SB91a], nous proposons d'exploiter l'information de couleur.

Afin d'apprécier la ressemblance entre deux séquences d'images, nous privilégions l'utilisation des informations de couleur et de texture apportées par les

[3] Certaines signatures qui peuvent contenir des informations sémantiques n'ont pas été exploitées

vêtements de la personne observée, plutôt que l'exploitation de celles provenant de la silhouette de cette personne. En effet, dans le cadre de la recherche d'une signature pour des séquences d'images, les critères peuvent être également ceux liés au mouvement. Or des personnes qui se déplaçent sont considérées, en termes de traitement d'images, comme des objets déformables. Dans ce contexte, une analyse des formes semble peu appropriée à ce genre de comparaison.

Nous nous intéresserons donc particulièrement aux signatures de séquences d'images qui permettent de caractériser leur contenu en termes de couleurs et de textures, tout en sachant que les objets observés sont déformables.

2.3.2.1 Comparaison de deux signatures de séquences d'images

Les signatures contiennent les informations nécessaires pour qu'une formule mathématique leur soit appliquée et fournisse une grandeur numérique qui représente leur ressemblance. Pour comparer les contenus de la séquence d'images requête S_{Req} à celui d'une séquence d'images candidate S_{Can}, une *fonction de comparaison* $Prox$ est appliquée aux signatures $\Phi(S_{Req})$ et $\Phi(S_{Can})$ des deux séquences d'images. Elle fournit un scalaire qui permet de quantifier la ressemblance entre le contenu de ces séquences d'images. Lorsque la nature de cette fonction de comparaison entre deux signatures n'est pas précisée, nous parlons de *mesure de proximité* :

$$Ressemblance(S_{Req}, S_{Can}) = Prox(\Phi(S_{Req}), \Phi(S_{Can}))$$

Ainsi une *forte mesure de proximité* entre deux signatures de séquences signifie simplement que les contenus des deux séquences se ressemblent, sans préciser la manière dont le résultat est obtenu.

La nature de la mesure de proximité varie selon les signatures employées. Nous considérons deux types de mesure de proximité : une mesure de similarité et une mesure de distance. Une *mesure de similarité* entre deux signatures est positive et bornée. Lorsqu'elle est proche de 1 nous considérons que les deux signatures se ressemblent fortement et lorsqu'elle est proche de 0, que les signatures ne se ressemblent pas du tout. Une *mesure de distance* est positive mais n'est pas bornée. Nous interprétons que les signatures sont proches lorsque la distance est proche de 0. Cette mesure s'évalue selon une métrique à définir.

Pour résumer la démarche nous pouvons dire que :

- La reconnaissance de personnes dans des séquences d'images revient à comparer les contenus de ces séquences deux-à-deux.
- Cette comparaison s'effectue en déterminant une mesure de proximité entre leurs deux signatures.
- Selon les signatures employées, la mesure de proximité appropriée est soit une mesure de similarité, soit une mesure de distance.

2.3.3 Respect des contraintes de performance liées à l'utilisation du système

Les temps de transmission des données vidéo, ainsi que le temps nécessaire à la comparaison d'une séquence requête avec toutes les séquences candidates acquises par toutes les caméras du site sont prohibitifs pour que ces opérations puissent être assumées par le superviseur. C'est pourquoi les traitements seront répartis sur les unités de calculs des caméras intelligentes.

Dans le cas de la localisation automatique, lorsque l'opérateur déclenche la procédure de localisation, l'unité de calcul associée à la caméra qui a acquis la séquence d'images requête est chargée d'extraire certaines données représentatives de cette séquence, qui sont désignées par le terme de **signature requête**.

A plus forte raison, dans le cas de la constitution des matrices origine-destination, la caméra intelligente qui a acquis une séquence est également chargée de constituer la signature qui lui correspond avant de la transmettre au superviseur.

2.3.3.1 Compacité des signatures

Dans le cas de la localisation automatique, la taille mémoire occupée par les signatures, appelée compacité des signatures, est un facteur prépondérant au respect des contraintes de performances.

Afin de limiter le temps de transmission de la signature requête aux autres unités de calculs, la taille de cette signature doit être la plus petite possible. En reçevant cette signature requête, chaque unité de calcul doit déterminer si l'une de ses séquences candidates est similaire à la séquence requête par l'évaluation d'une des mesures de proximité entre la signature requête et les signatures candidates.

La compacité des signatures permet également de réduire le temps nécessaire au calcul de la mesure de proximité, pour faire en sorte que ce calcul n'excède

pas la durée moyenne du passage d'une personne dans le champ de vision d'une caméra intelligente.

Dans le cas de la constitution des matrices origine-destination, il est, cette fois, indispensable de calculer toutes les signatures de toutes les séquences requêtes et candidates.

Le nombre nécessaire de comparaisons est, de plus, bien plus important que dans le cas de la localisation puisque chaque séquence-requête n'est plus comparée à une seule séquence-candidate, mais à toutes les séquences-candidates disponibles.

Toutefois, la faiblesse du temps de calcul nécessaire à une comparaison de signatures n'est plus primordiale puisque ces calculs peuvent être effectués hors-ligne, par exemple la nuit, une fois que le site est fermé au public. Dans ce cas, c'est la taille mémoire nécessaire au stockage des signatures qui devient un facteur crucial d'efficacité.

Cependant, un compromis sera toujours nécessaire entre la compacité d'une signature et son pouvoir de discrimination entre les cas où deux séquences-personnes sont similaires et les cas où elles sont différentes.

2.3.3.2 Temps de calcul des signatures candidates

Notre démarche présente de nombreux points communs avec la recherche d'images par l'exemple [SC96]. Une recherche commence par la constitution d'une signature requête, puis cette signature est comparée aux signatures candidates déjà disponibles, car préalablement calculées. Le temps nécessaire au calcul des signatures n'est pas, dans ce cas, un facteur crucial devant impérativement être minimisé. Seul le temps nécessaire à la comparaison des signatures doit être limité afin de garantir le confort d'utilisation d'un système de recherche d'images par l'exemple.

Dans le cas de la localisation automatique d'une personne à travers un réseau de caméras, aucune des signatures candidates n'est disponible au moment de la création de la signature requête. Il est donc primordial de minimiser également le temps nécessaire au calcul des signatures candidates, au même titre que le temps nécessaire à la mesure de proximité entre deux signatures.

2.3.4 Exploitation des signatures de séquences d'images dans notre application

Maintenant que nous avons défini les notions de signatures de séquences d'images et celle de mesure de proximité entre deux signatures, nous expliquons comment les exploiter dans les deux systèmes que nous avons décrits : le système de localisation de personnes et le système de constitution de matrices origine-destination.

2.3.4.1 Localisation de personnes

Dans le cas du système de localisation de personnes à travers un réseau de caméra intelligentes, le déroulement des actions est le suivant :

1. Dans un premier temps, une situation potentiellement dangereuse a été détectée (par exemple l'intrusion d'une personne dans une zone interdite aux usagers) et la séquence d'images qui lui correspond a été sauvegardée sur le disque dur embarqué dans la caméra intelligente. L'agent de surveillance a été automatiquement informé par le superviseur et visionne cette séquence. Le cas échéant, s'il désire suivre le déplacement de la personne qui est à l'origine de la situation, il déclenche la procédure de localisation de cette personne.

2. La caméra intelligente, appelée alors caméra-requête, extrait ensuite de cette séquence d'images la séquence-personne requête puis constitue la signature requête qu'elle transmet à toutes les autres caméra intelligentes, dites candidates, surveillant le site.

3. Chaque passage d'une personne dans le champ de vision de chacune des caméras intelligentes candidates est converti en une séquence-personne. La caméra intelligente candidate extrait la signature candidate qu'elle compare à la signature requête dont elle dispose, grâce à une mesure de proximité.

4. Le résultat de la mesure de proximité est transmis au PCC où le superviseur détermine, parmi toutes les caméras intelligentes candidates, celle qui fournit la signature candidate qui ressemble le plus à la signature requête. L'agent de surveillance est alors informé de la localisation probable de la personne dont il désire suivre le déplacement.

2.3.4.2 Constitution des matrices origine-destination

Dans le cas de la constitution des matrices origine-destination, il s'agit de faire correspondre chaque séquence requête représentant une personne entrant dans le site à la séquence candidate représentant la même personne quittant le site. Pour ce faire, le déroulement des opérations est le suivant :

1. La signature de chaque séquence requête est extraite par une caméra intelligente qui observe la scène correspondante (une personne entrant dans le site), puis la caméra intelligente transmet la signature requête au superviseur.
2. La signature de chaque séquence candidate est également extraite par une caméra intelligente qui observe la scène correspondante (une personne quittant le site). La caméra intelligente transmet ensuite la signature candidate au superviseur.
3. Pour chaque signature requête, le superviseur calcule hors-ligne les mesures de proximité avec les signatures candidates représentant des séquences acquises à des instants ultérieurs à la séquence requête. En triant par ordre croissant les mesures de proximité entre chaque signature requête et les signatures candidates, il détermine la séquence candidate qui correspond le plus vraisemblablement à chaque séquence requête, ce qui lui permet de constituer la matrice origine-destination.

2.4 Acquisition et pré-traitements des séquences d'images

Nous avons introduit la notion de signature de séquence d'images, mais il reste à présenter les données à partir desquelles nous allons constituer ces signatures. Ce paragraphe est consacré à la présentation des séquences d'images considérées dans le cadre de notre travail.

Nous décrivons d'abord la mise en place, au sein du laboratoire LAGIS, d'une plateforme d'acquisition de séquences d'images couleur qui est proche des conditions d'acquisition au sein d'une station de métro.

Nous détaillons ensuite les pré-traitements effectués sur les séquences d'images considérées afin de les rendre exploitables.

Enfin, nous présentons les séquences d'images examinées et mettons en évidence certaines caractéristiques de ces séquences au travers d'exemples.

2.4.1 Plateforme d'acquisition des séquences d'images

Conformément aux contraintes imposées par le cahier des charges, les images acquises représentent les passages de différentes personnes dans des conditions proches de celles d'un site de transports publics.

L'objectif de la mise en place de cette plate-forme expérimentale en laboratoire est de constituer une base de séquences d'images représentant le déplacement de différentes personnes dans un couloir de métro.

2.4.1.1 Dispositif d'acquisition

Lors de la conception du dispositif d'acquisition, plusieurs paramètres des conditions d'acquisition ont été simplifiés pour les prises de vue, en particulier les conditions d'éclairage et la trajectoire de la personne dans la scène. Celle-ci se déplace sur un tapis de couleur noire de telle sorte que l'arrière-plan de la scène est réduit à un fond homogène, ceci afin de faciliter l'extraction des pixels représentant la personne en déplacement. La figure 2.1 est une vue latérale de la plate-forme d'acquisition des séquences d'images.

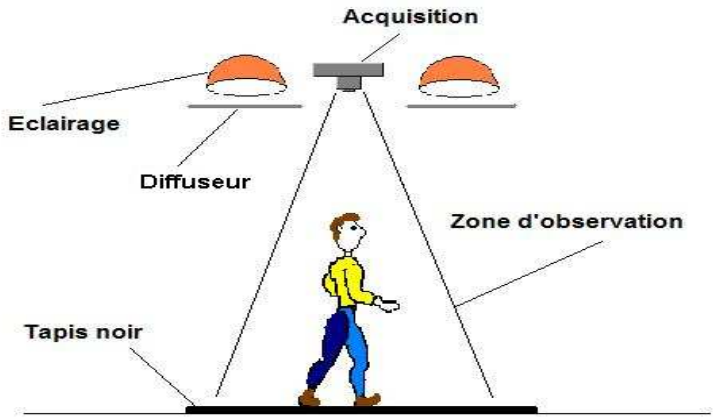


FIG. 2.1 – *Schéma du dispositif d'acquisition des séquences d'images couleur.*

Les prises de vue ont été réalisées sous des conditions d'éclairage contrôlées. Un seul illuminant de type halogène a été utilisé. La géométrie du dispositif

d'éclairage a été ajustée de telle sorte que l'intensité des rayons incidents soit relativement constante dans la scène observée. Pour cela, nous avons placé deux projecteurs quartz-halogène focalisables d'une puissance de 800 Watts chacun, de part et d'autre de la caméra. De plus, nous les avons recouverts de papier calque qui diffuse la lumière émise. Bien que nous ayons pris beaucoup de soin à ajuster la géométrie du dispositif d'éclairage, nous constaterons, dans le quatrième paragraphe, la présence d'ombres dans les images des séquences considérées.

La caméra utilisée est le modèle couleur XC003P de Sony. Il s'agit d'une caméra matricielle tri-CCD composée de trois capteurs CCD disposés de telle sorte qu'un élément de surface d'un objet se projette sur trois éléments photosensibles différents, sensibles aux trois couleurs rouge, vert et bleu. Ainsi ces trois éléments fournissent les trois composantes couleur rouge, verte et bleue d'un pixel.

La taille de chaque capteur est $\frac{1}{3}$ de pouce, la distance focale de l'objectif utilisé est 8 mm et la caméra est placée à une hauteur de 2m80. Le champ de vision couvre donc une surface rectangulaire au sol de 1m20 de largeur et de 1m70 de longueur.

La résolution du capteur est de 768×576 pixels en trames entrelacées, et la fréquence d'acquisition est de 25 images par seconde.

La caméra fournit des images couleur analogiques qui sont numérisées par la carte d'acquisition Matrox MeteorII, quantifiant les niveaux d'intensité de chaque canal Rouge, Vert, Bleu sur 256 niveaux.

2.4.1.2 Le passage d'une personne et la séquence d'images correspondante

Une fois le dispositif d'acquisition mis au point, il s'agit de reproduire le déplacement des usagers dans un couloir de métro. Notre étude se limite au cas où une seule personne à la fois est présente dans le champ d'observation.

Dans le but de simplifier le contenu des scènes à analyser, nous avons considéré le cas le plus simple, où la direction du déplacement est parallèle à celle du couloir. Les personnes observées selon une vue de dessus ont suivi une trajectoire rectiligne parallèle à la largeur du champ d'observation de la caméra. Lors de l'expérience, elles ont adopté une démarche relativement naturelle, sans gestes particuliers, et se sont déplacées à une vitesse normale.

Toutes les séquences d'images acquises durent exactement quatre secondes. A

raison de 25 images par seconde, nous disposons donc de séquences de 100 images. La première image et la dernière image d'une séquence représentent le fond noir uniforme.

Une séquence représentant le passage d'une personne dans le champ d'observation de la caméra peut être décomposée en trois parties : l'apparition progressive de la personne dans le champ, la traversée du champ durant laquelle la personne est vue entièrement, et enfin sa disparition progressive du champ d'observation. Nous proposons d'illustrer ces trois parties à l'aide d'un exemple de passage d'une personne.

La figure 2.2 fournit des images[4] extraites de la première partie du passage de la personne. Celle-ci entre progressivement dans le champ d'observation de la caméra : ses pieds apparaissent dans les premières images, viennent ensuite les jambes et le haut du buste.

Puis la deuxième partie de la séquence est constituée des images dans lesquelles la personne apparaît en entier. Celle-ci traverse le champ d'observation de la caméra. La figure 2.3 représente des images extraites de la deuxième phase de son passage.

Enfin, la troisième partie correspond à la disparition progressive de la personne du champ d'observation, représentée en figure 2.4.

[4] Ces images ont été pré-traitées, comme nous le verrons dans la prochaine section, de sorte que seuls les pixels représentant la personne sont représentés

t=600ms	t=800ms	t=1000ms
		

FIG. 2.2 – *Apparition progressive de la personne dans le champ d'observation (première partie de la séquence $\mathcal{A}^{(1)}$). La durée en ms correspond à celle qui sépare l'instant de l'acquisition de la première image de la séquence de l'instant d'acquisition de l'image affichée.*

t=1300ms	t=1400ms	t=1500ms
		
t=1600ms	t=1700ms	t=1800ms
		

FIG. 2.3 – *Traversée de la personne dans le champ d'observation (deuxième partie de la séquence $\mathcal{A}^{(1)}$).*

t=2200ms	t=2400ms	t=2600ms
		

FIG. 2.4 – *Disparition progressive de la personne du champ d'observation (troisième partie de la séquence $\mathcal{A}^{(1)}$).*

2.4.2 Pré-traitements et constitution des séquences-personnes

Afin de pouvoir comparer de manière efficace les images des séquences, un certain nombre de pré-traitements ont dû être effectués.

Il s'agit du désentrelacement des images, de la sélection des images exploitables et de l'extraction des pixels-personnes, à savoir des pixels représentant la

personne, par opposition aux pixels qui représentent l'arrière-plan de la scène matérialisé par le tapis de couleur noire.

Nous définissons ensuite le concept de *séquence-personne*, constituée d'images contenant uniquement des pixels-personnes, et représentant cette personne en entier.

2.4.2.1 Désentrelacement

La première étape des pré-traitements à effectuer sur les séquences d'images acquises est leur désentrelacement temporel. Cette opération, illustrée figure 2.5, consiste à séparer les trames paires et impaires qui composent les lignes de chaque image.

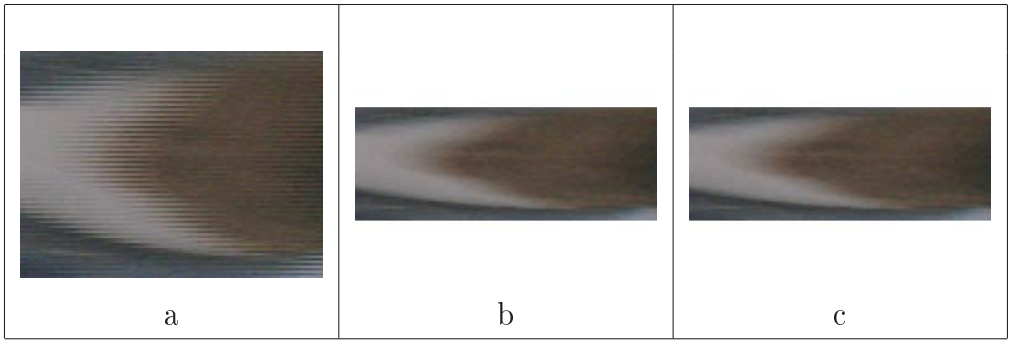


FIG. 2.5 – *Procédé de désentrelacement : a) Extrait d'une image entrelacée. b) Demi-image obtenue à partir des trames impaires. c) Demi-image obtenue à partir des trames paires.*

En effet, le procédé d'acquisition de la caméra est tel que deux demi-images sont acquises successivement lors de chaque cycle d'acquisition, puis sont entrelacées pour former une image par cycle.

Ce procédé permet d'obtenir une excellente qualité du rendu visuel de la séquence lorsque celle-ci est affichée sur un écran. Cependant, lorsque les images sont destinées à subir des traitements numériques, il convient de les présenter sous forme désentrelacée.

Ainsi, à partir d'une séquence de 100 images de 768×576 pixels, acquises à la fréquence de 25 images par seconde, nous obtenons une séquence de 200 images de 768×288 pixels acquises à la fréquence de 50 images par seconde.

2.4.2.2 Sélection des images exploitables

Tout d'abord, seules les images dans lesquelles la personne est représentée en entier seront conservées. En effet, les images dans lesquelles n'apparaissent que les jambes et le bas du buste de l'individu contiennent trop peu d'informations exploitables pour caractériser la personne observée.

Le choix du critère des images "complètes" est donc arbitraire, mais présente l'avantage d'une mise en œuvre assez simple. Nous avons sélectionné ces images de manière interactive, en les examinant à l'aide d'un simple programme de visualisation d'images.

Nous obtenons en moyenne une centaine d'images complètes par séquence, la grande majorité des séquences contenant entre 80 et 120 images complètes.

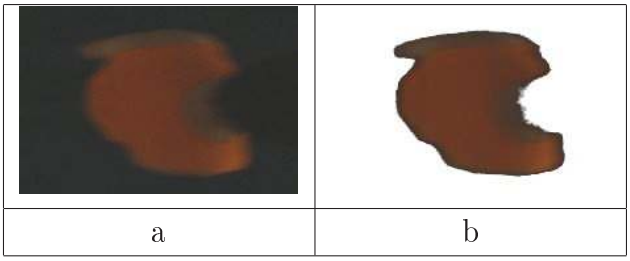
2.4.2.3 Extraction des pixels-personnes

Les pixels représentant la personne (appelés *pixels-personnes*) sont ensuite extraits de chacune des images complètes de la séquence. La figure 2.6 illustre le procédé d'extraction des pixels-personnes.

Ce travail est facilité par la présence au sol du tapis noir : l'analyse de l'image de fond représentant uniquement le tapis nous a permis de relever une caractéristique commune à tous les pixels formant cette image.

En effet, nous constatons que la somme de leurs trois composantes colorimétriques Rouge, Vert et Bleu ne dépasse pas une valeur de seuil fixée à 30 pour ces expériences. Pour chaque image représentant la personne en entier, nous "éliminons" donc tous les pixels dont la somme des composantes colorimétriques est inférieure à la valeur 30 en leur attribuant la couleur noire (R=0,G=0,B=0). Nous précisons que, dans ce document, ces pixels seront représentés en blanc pour éviter l'impression de surfaces noires importantes.

Il est à noter cependant que cette méthode présente un inconvénient majeur, remarquable dans l'exemple de la figure 2.6. Lorsque les cheveux de la personne sont bruns, les pixels représentant la tête sont, eux aussi, supprimés car la somme de leurs composantes trichromatiques est inférieure au seuil de valeur 30. Cependant, ce traitement très simple permet d'obtenir des images exploitables pour la comparaison.



a b

FIG. 2.6 – *Image représentant un individu (a) et le résultat de l'extraction des pixels-personnes (b). Les pixels représentant le fond sont étiquettés en blanc.*

2.4.2.4 Constitution des séquences-personnes

On appelle image-personne l'image constituée des pixels-personnes, et dont les pixels représentant le fond ont été étiquetés en blanc.

Définition 1 (Séquence-personne) *L'ensemble des images-personnes conservées constituent une* ***séquence-personne****.*

Les pré-traitements détaillés dans ce paragraphe sont simplistes, car nous nous sommes mis dans des conditions idéales pour faciliter l'extraction des pixels personnes. Ils devront être adaptés pour une implantation ultérieure du système de localisation de personnes dans des sites publics, en fonction des capteurs employés et des conditions d'éclairage.

2.4.3 La base de séquences-personnes

2.4.3.1 Les huit personnes de la base

Une première base de données de séquences-personnes a été constituée au sein des locaux du LAGIS.

Huit personnes se sont déplacées sous notre dispositif, chacune à quatre reprises et à une allure de marche normale. Chaque personne est désignée par une lettre de l'alphabet allant de $\mathcal{A}$ à $\mathcal{H}$.

La séquence-personne correspondant à l'un de ces quatre passages est désignée par la lettre correspondant à la personne, accompagnée d'un chiffre : 1, 2, 3 ou 4. Par exemple, à la personne $\mathcal{A}$ correspondent les quatre séquences-personnes $\mathcal{A}^{(1)}$, $\mathcal{A}^{(2)}$, $\mathcal{A}^{(3)}$ et $\mathcal{A}^{(4)}$.

La figure 2.7 montre huit images représentant les huit personnes observées selon une vue de dessus.

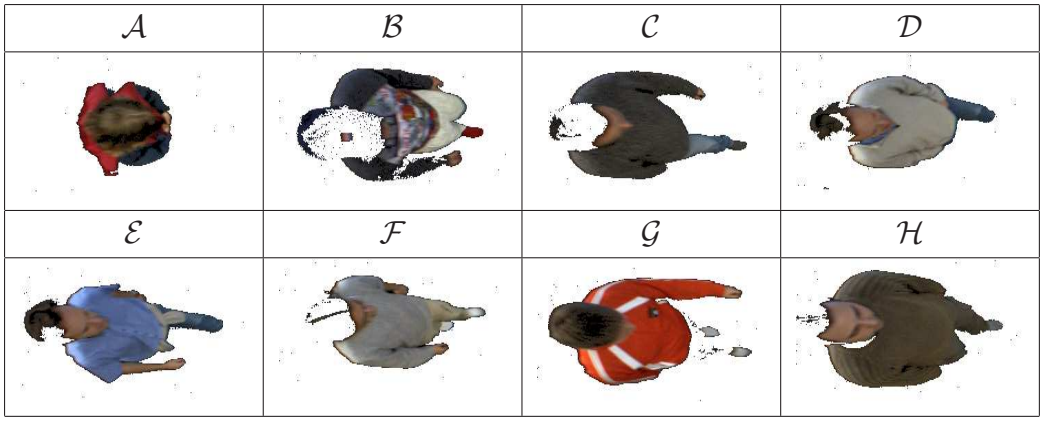


FIG. 2.7 – *Les huit individus de la base.*

Bien que cette base soit composée de huit personnes différente seulement, nous verrons par la suite que certains couples de séquences représentant deux personnes différentes sont malgré tout difficilement différenciables de couples représentant deux fois la même personne. Ainsi, afin de pouvoir évaluer la robustesse des algorithmes de comparaison de séquences développés, la base est constituée de telle sorte que deux personnes différentes se ressemblent fortement, comme les personnes $\mathcal{D}$ et $\mathcal{F}$. Seule la couleur de leur pantalon permet alors de les distinguer.

2.4.3.2 Description d'une séquence-personne

Un examen visuel permet de dégager certaines caractéristiques des séquences-personnes.

La caméra étant fixe, le seul mouvement perceptible est dû au déplacement de la personne. Un autre élément important concerne la continuité du mouvement tout au long de la séquence-personne, et donc l'absence de ruptures brusques entre deux images successives.

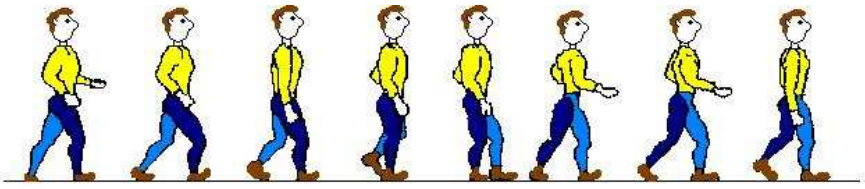


FIG. 2.8 – *Les différentes postures adoptées lors de la marche.*

La figure 2.8 illustre les différentes postures adoptées lors de la marche. En

raison du rythme de la marche, les jambes et les bras ont un mouvement de balancier, ce qui explique la diversité des images lors du déplacement de la personne à travers le champ d'observation. Dans certaines images d'une séquence-personne, les bras et les jambes sont parfaitement visibles, tandis que dans d'autres images ces membres sont occultés car ils sont parallèles au corps. Enfin, une majorité des images d'une même séquence représentent des positions intermédiaires de ces membres entre ces deux positions.

2.4.3.3 Deux exemples

La figure **??** présente des images extraites de trois séquences-personnes. Nous retrouvons les positions évoquées dans l'analyse de la marche. Une image sur huit est affichée à partir de la séquence acquise à 50 images par seconde, soit une image affichée toutes les 160 millisecondes.

2.5 Conclusion

A travers le cahier des charges fonctionnel, nous avons défini deux scénarii répondant aux besoins des exploitants en termes de sécurité et d'aide à l'exploitation. Nous avons également dressé la liste des contraintes qui nous sont imposées, ainsi que les performances requises relatives aux temps de réponse des systèmes envisagés.

Nous avons ensuite proposé une approche selon laquelle la localisation automatique de personnes ainsi que l'élaboration automatique de matrices origine-destination souhaitées par les opérateurs de transports reposent sur la comparaison de séquences d'images. Dans notre cas d'étude, cette comparaison nécessite la définition de signatures représentatives des séquences à comparer.

En raison des impératifs relatifs au temps de réponse du système, ces signatures doivent être suffisamment compactes pour, d'une part, pouvoir être rapidement transmises à travers le réseau et d'autre part, pour que la mesure de ressemblance puisse être calculée dans un délai compatible avec les contraintes de temps de réponse imposées par les exploitants de sites de transports publics. Les différentes contraintes ont été détaillées pour la localisation automatique de personnes et la constitution des matrices origine-destination.

Nous avons ensuite décrit, pour chacun de ces deux cas, le protocole d'utili-

	$\mathcal{C}^{(1)}$	$\mathcal{D}^{(1)}$
t		
t+160 ms		
t+320 ms		
t+480 ms		
t+640 ms		
t=800 ms		
t+960 ms		
t+1120 ms		
t+1380 ms		
t+1540 ms		
t+1700 ms		

sation des signatures, en réponse aux deux scénarii du cahier des charges.

Enfin, nous avons décrit les données expérimentales à partir desquelles nous travaillerons.

Grâce à une plate-forme expérimentale, nous avons dans un premier temps reconstitué les conditions d'acquisition conformes au cahier des charges : les personnes qui ont participé aux prises de vues ont traversé le champ d'observation de la caméra en simulant un passage dans un couloir observé à la verticale.

Nous avons ensuite décrit les différents pré-traitements qui ont été appli-

qués aux séquences d'images pour obtenir des séquences-personnes : séquences d'images représentant la personne en entier et dont on retient uniquement les pixels représentant la personne.

Les huit personnes constituant la base de données de travail ont été décrites, chacun étant passé quatre fois dans le champ d'observation de la caméra. Trois séquences-personnes ont également été présentées et leurs premières caractéristiques mises en évidence.

Chapitre 3

Comparaison d'images couleur : signatures et descripteurs

3.1 Introduction

Dans le deuxième chapitre, nous avons présenté les séquences d'images couleur acquises, conformément au cahier des charges, dans des conditions qui reproduisent celles d'un site de transports publics. Après avoir détaillé les prétraitements qui ont été appliqués à ces séquences, nous avons défini la notion de séquence-personne qui constitue l'objet de notre étude, puis nous avons fourni quelques exemples de séquences-personnes afin d'en dégager certaines caractéristiques.

Ce chapitre dresse un état de l'art non exhaustif de la comparaison d'images à partir de leurs signatures qui regroupent différents attributs caractéristiques extraits de ces images.

Dans le premier paragraphe, nous expliquons le processus physique de formation de la couleur et la physiologie de sa perception par un être humain. Nous présentons également les différents systèmes employés pour représenter la couleur, avant de définir chacune des images numériques couleur constituant les séquences considérées.

Le deuxième paragraphe traite de la reconnaissance d'objets dans des images couleur par comparaison de leur contenu, à partir de leurs signatures colorimétriques. Nous attachons une grande importance à cette technique car elle est à la

base de nombreuses méthodes de comparaison de séquences d'images.

Enfin, le troisième paragraphe présente une seconde famille de signatures qui permettent de caractériser les textures observées dans une image. Nous montrons en quoi elles contiennent une information plus riche que les signatures prenant uniquement en compte la distribution des couleurs des pixels.

3.2 Image numérique couleur

Nous présentons dans un premier temps les concepts de base de la perception humaine de la couleur. Nous montrons que la lumière d'un objet dépend de l'éclairage et expliquons brièvement comment l'être humain perçoit la couleur.

Ceci nous amène ensuite à expliquer de quelle manière la couleur est mesurée dans le cadre d'applications industrielles. Nous découvrons également les raisons pour lesquelles certaines conventions ont été établies en se basant sur le fonctionnement de la vision humaine.

Puis nous passons en revue les différents systèmes dans lesquels la couleur peut être représentée. Nous dégageons plusieurs familles de systèmes de représentation, selon les attributs qui sont employés pour carctériser la couleur.

Enfin, nous donnons la définition mathématique d'une image numérique couleur.

3.2.1 Lumière et perception humaine

La perception de la couleur d'un objet résulte d'une succession complexe de différents phénomènes physiques et physiologiques. L'interprétation de ces phénomènes, effectuée par le cerveau humain, fait de la couleur une notion subjective car elle obéit en partie à des lois de nature psychologique. Nous détaillons ici le processus de formation du stimulus de couleur d'un objet, à partir de la formation de la lumière et jusqu'à l'étape d'interprétation assurée par le cortex cérébral.

En premier lieu, indépendamment de l'objet observé, la lumière est le phénomène physique indispensable à la formation de la couleur. Elle est émise par une source naturelle ou artificielle (comme le soleil ou une ampoule). Chaque source lumineuse est caractérisée par une grandeur physique : sa *répartition spectrale d'énergie relative*, notée $S(\lambda)$.

Cette fonction s'exprime sans unité et correspond à la normalisation de la répartition spectrale d'énergie pour une longueur d'onde particulière. Certaines sources de lumière correspondant à des conditions courantes d'observation ont été normalisées par la CIE (Commission Internationale de l'Eclairage) sous le nom d'*illuminants*. Les fonctions correspondant à quelques illuminants courants sont représentées sur la figure 3.1.

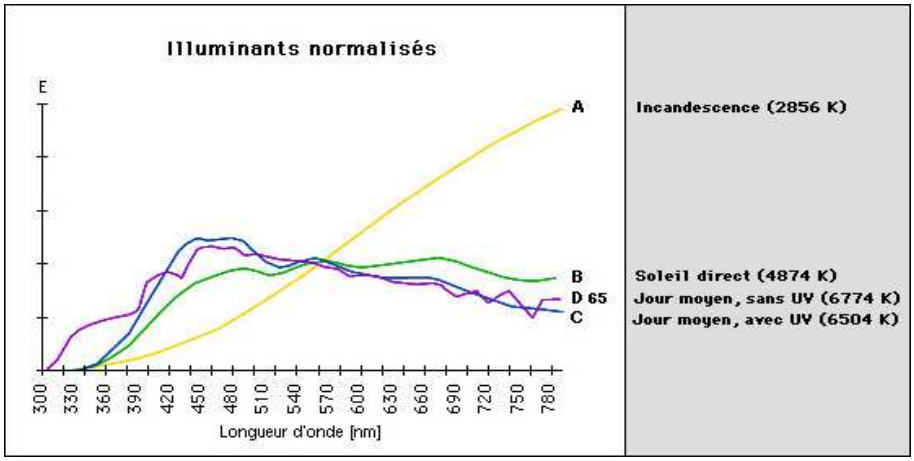


FIG. 3.1 – *Quelques illuminants définis par la CIE.*

Une partie des rayons lumineux incidents (le *spectre*) émis par la source est absorbée par les pigments de l'objet, tandis qu'une autre partie est réfléchie pour former le *stimulus de couleur*, noté $C(\lambda)$.

La figure 3.2 illustre le cheminement des rayons lumineux, de la source à l'oeil en passant par l'objet observé. Les caractéristiques d'un matériau sont estimées par sa *réflectance spectrale* notée $\beta(\lambda)$, qui correspond au rapport entre l'énergie de la lumière réfléchie par le matériau et l'énergie de la lumière réfléchie par un diffuseur parfait observé dans les mêmes conditions d'éclairage et d'observation. Le stimulus de couleur s'exprime par la relation $C(\lambda) = \beta(\lambda) \times S(\lambda)$.

Le stimulus de couleur atteint ensuite l'oeil dans lequel il traverse la cornée, le cristallin puis le corps vitré avant de se focaliser sur la rétine. Celle-ci comporte deux types de cellules photosensibles, ou photorécepteurs : les cônes et les bâtonnets [col04].

Les bâtonnets permettent la vision nocturne ou scotopique. On en dénombre

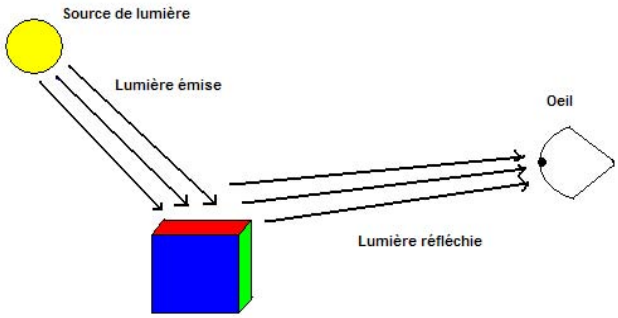


FIG. 3.2 – *Les rayons lumineux émis par la source sont partiellement réfléchis par l'objet jusqu'à atteindre l'oeil.*

environ 80 à 140 millions et leur répartition est plus dense à la périphérie de la rétine qu'au centre. Ils ne nous permettent pas, quand l'intensité de la lumière est faible, de percevoir la couleur des objets. Par ailleurs, ils sont sensibles au mouvement.

Les cônes permettent la vision diurne ou photopique. Ils sont beaucoup moins nombreux que les bâtonnets puisqu'on n'en dénombre que 4 à 7 millions. Leur densité est la plus importante au niveau de la fovéa, où notre vision est la plus sensible. En effet, on y dénombre environ 50000 cônes et aucun bâtonnet. Lorsqu'on fixe un objet, les rayons lumineux issus de cet objet se focalisent sur la fovéa, siège de l'acuité visuelle. Lorsque l'intensité de la lumière incidente est suffisante, ils nous permettent de percevoir la couleur des objets. En effet, les cônes contiennent des pigments qui réagissent à la réception du stimulus et qui déclenchent des impulsions électriques envoyées au cerveau [Kow90]. Les cônes se répartissent en trois types qui diffèrent par leur sensibilité spectrale, comme le montre la figure 3.3.

Les courbes de sensibilité relatives à ces trois pigments présentent des maxima aux longueurs d'onde des couleurs rouge, vert et bleu, à savoir respectivement 570 nm (notée L pour Long), 530 nm (notée M pour Medium) et 440 nm (notée S pour Small). Ce sont donc ces pigments qui effectuent la conversion de l'énergie lumineuse en un signal interprétable par le système nerveux. 4% des cônes sont sensibles à la couleur bleue, 32% au vert et 64% au rouge.

Enfin, les cônes et les bâtonnets convertissent l'énergie lumineuse en un signal véhiculé par le nerfs optique en direction du cerveau. C'est à ce moment que le

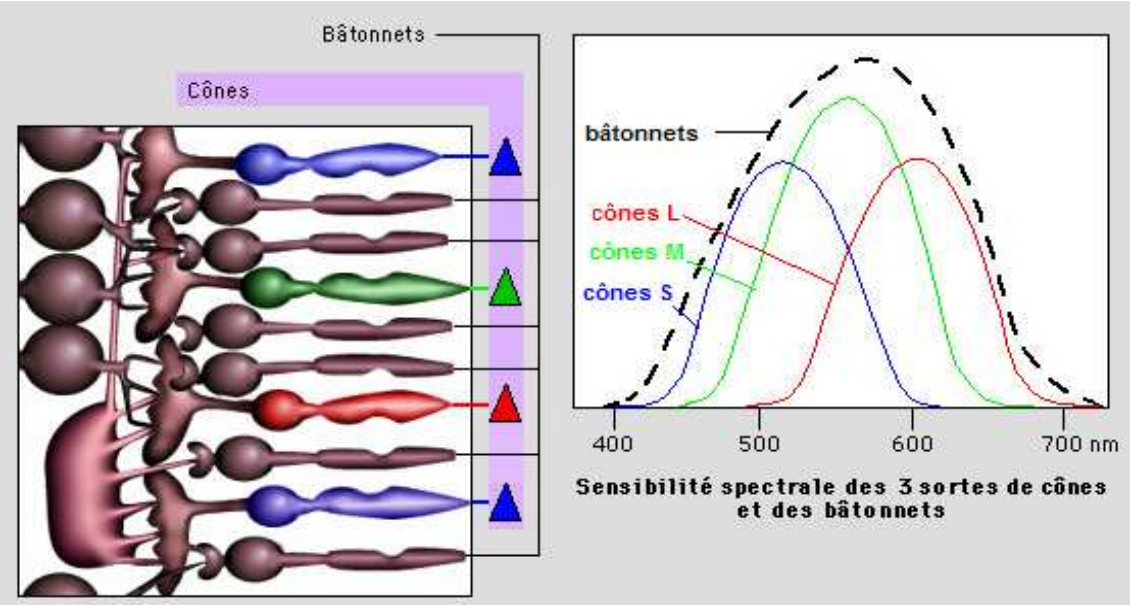


FIG. 3.3 – *Les cônes et les bâtonnets de la rétine humaine, et leurs sensibilités respectives dans le domaine spectral du visible.*

signal électro-chimique est interprété par le cortex visuel comme une "sensation" de couleur.

3.2.2 Représentation de la couleur

La compréhension du mécanisme de perception de la couleur, en particulier la distinction des trois types de cônes de sensibilités spectrales différentes a fourni la base physiologique d'une théorie fondamentale en colorimétrie : la trivariance visuelle ou théorie trichromatique présentée dans [Tro91].

Elle met en évidence que tout stimulus de couleur peut être reproduit par le mélange de trois autres stimuli : le rouge, le vert et le bleu appelés primaires de référence. Ces trois couleurs primaires sont nécessaires et suffisantes pour reproduire toutes les couleurs par synthèse additive, c'est-à-dire par le mélange de lumières colorées correspondant chacune à une des trois primaires.

Pour exprimer la mesure d'un stimulus de couleur à l'aide de ces couleurs primaires, il est nécessaire de définir un système de représentation de la couleur. A ces trois primaires, nous faisons correspondre respectivement trois vecteurs directeurs normés $\mathbf{R}$, $\mathbf{G}$ et $\mathbf{B}$ qui forment le repère d'un espace tridimensionnel

fini d'origine O, appelé cube des couleurs.

Dans cet espace, chaque stimulus de couleur est représenté par un point-couleur $\mathbf{c}$ dont les coordonnées $I^R(\mathbf{c})$, $I^G(\mathbf{c})$ et $I^B(\mathbf{c})$ sont les niveaux d'intensité normalisés des trois couleurs primaires. L'origine O du cube correspond à la couleur noire $(0,0,0)$ tandis que le blanc se trouve aux coordonnées $(1,1,1)$. L'axe des gris qui joint ces deux points est appelé axe achromatique. Il contient toutes les nuances de gris allant du noir au blanc.

3.2.3 Systèmes de représentation

Nous avons vu que la couleur peut être représentée dans un repère tridimensionnel appelé cube des couleurs. Ce premier système de représentation s'inspire de la perception humaine de la couleur.

Cependant, il existe dans la littérature d'autres systèmes de représentation qui permettent d'exprimer les composantes d'une couleur, chaque système étant conçu pour correspondre à des propriétés physiques ou physiologiques particulières, ou encore en fonction de l'utilisation qui en sera faite.

Nous présentons ici plusieurs familles de systèmes de représentation issues de [Sèv96] et [Van00], et détaillons certains d'entre eux qui seront utilisés afin de mesurer leur influence dans le cadre de la comparaison d'images. Nous donnons également les fonctions de passage des composantes trichromatiques R, G et B aux composantes exprimées dans les autres systèmes.

3.2.3.1 Systèmes de primaires

Dans le système (R,G,B), les composantes trichromatiques d'un stimulus de couleur sont liées à sa *luminance*, qui s'obtient par la somme de ces trois composantes. L'attribut de luminance correspond à la sensation visuelle selon laquelle une surface paraît émettre plus ou moins de lumière : on la qualifie alors de plus ou moins sombre ou claire.

Un des inconvénients du système de représentation (R,G,B) pour la reconnaissance d'objets est sa faible robustesse face aux changements des conditions d'éclairage. Considérons deux images représentant le même objet éclairé avec le même type d'illuminant mais dont l'intensité varie d'une image à l'autre. Muselet [MMKP02] a montré que les couleurs d'un même élément de surface présent dans les deux images sont différentes. En effet deux stimuli de couleur qui possèdent

le même caractère chromatique sont caractérisés par des composantes trichromatiques R, G et B différentes à cause de leur luminance.

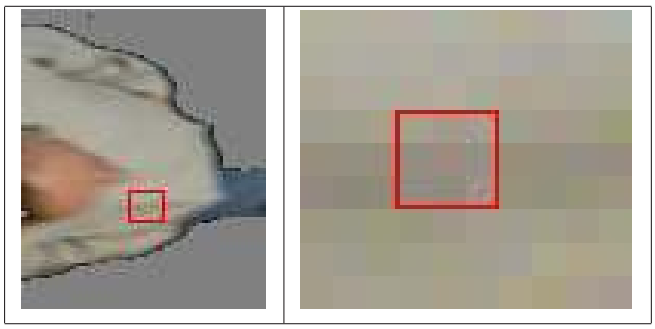


FIG. 3.4 – *Une partie d'une image couleur représentée dans le système* (R, G, B) *extraite de la base de données (a) et l'agrandissement des pixels de la zone en rouge (b).*

En analysant les séquences-personnes de la base de données réalisée en laboratoire, nous avons constaté la présence d'ombres. Ces ombres sont provoquées conjointement par desvariation locale d'intensité de la lumière incidente et par des variations locales des propriétés de réflectance des vêtements observés. Nous pouvons constater ce phénomène sur la figure 3.4 qui montre une personne portant des vêtements de couleurs unies. Les couleurs des pixels présentent pourtant une importante variation selon que les parties du corps sont plus ou moins éclairées. En effet, la formation de plis dans les vêtements entraîne inévitablement ce phénomène d'ombres remarquables sur presque toutes les images de toutes les séquences.

Pour les neufs pixels délimités par le carré rouge au centre de l'agrandissement, les valeurs des trois canaux R, G et B sont reportées dans les matrices suivantes :

$$I^R = \begin{pmatrix} 95 & 95 & 93 \\ 79 & 88 & 74 \\ 73 & 73 & 82 \end{pmatrix}, \; I^G = \begin{pmatrix} 91 & 84 & 89 \\ 69 & 66 & 75 \\ 68 & 66 & 75 \end{pmatrix}, \; I^B = \begin{pmatrix} 67 & 69 & 74 \\ 56 & 56 & 69 \\ 59 & 55 & 63 \end{pmatrix}$$

On constate dans cet exemple que, en raison des variations locales de la réflectance de la surface du tissus, les composantes couleur de deux pixels voisins peuvent être très différentes alors qu'ils représentent tous un pull-over de couleur unie. Ainsi, sur le canal correspondant à la couleur Rouge, on observe une dif-

férence de $95 - 73 = 22$ niveaux dans un seul voisinage 3×3 supposé être de couleur unie.

Le système (R, G, B) ne semble donc pas adapté à nos besoins puisque nous souhaitons caractériser les pixels représentant une surface homogène par des couleurs très proches. Il s'agit de trouver un système de représentation qui soit le moins sensible possible à de telles variations locales de réflectance spectrale.

Ceci nous amène à présenter un second système de représentation de primaires pour ne prendre en compte que les caractéristiques de chrominance. On normalise les valeurs des composantes trichromatiques par rapport à la luminance en normalisant chacune d'entre elles par la somme des trois. Les composantes ainsi obtenues sont appelées coordonnées trichromatiques, coordonnées réduites ou encore composantes normalisées.

Le système (r, g, b) est donc une représentation plane des couleurs inscrite dans l'espace (R, G, B). Les composantes trichromatiques normalisées sont obtenues à partir des composantes trichromatiques (R, G, B) de la manière suivante :

$$\begin{cases} r = \frac{R}{R+G+B} \\ g = \frac{G}{R+G+B} \\ b = \frac{B}{R+G+B} = 1 - r - g \end{cases} \tag{3.1}$$

Dans le système de représentation (R, G, B), les trois sommets du cube correspondant aux trois primaires R, G et B forment le triangle de Maxwell qui est perpendiculaire à l'axe achromatique. La figure 3.5, représente le cube des couleurs : elle fournit une représentation des axes du système (R, G, B) ainsi que de l'axe achromatique, et met en évidence le triangle de Maxwell.

Notons que, dans ce système, seules deux composantes sont nécessaires et suffisantes pour représenter un point sur le triangle de Maxwell puisque la troisième composante est obtenue à partir des deux premières. Ces composantes couleur renseignent sur les proportions de rouge, de vert et de bleu du pixel, quelle que soit la luminance. Il faut donc prendre deux composantes parmi les trois pour être indépendant de l'éclairage.

Les composantes normalisées sont moins sensibles aux variations locales des propriétés de réflectance de la surface des objets que les composantes (R, G, B). Dans la cas de la reconnaissance d'objets, l'utilisation du système de représentation (r, g, b) peut être préférable à celle du système (R, G, B). Dans le cadre de notre étude, bien que l'illuminant utilisé soit le même pour toutes les prises de vues, nous avons remarqué que la présence d'ombres est inévitable. Or celles-ci

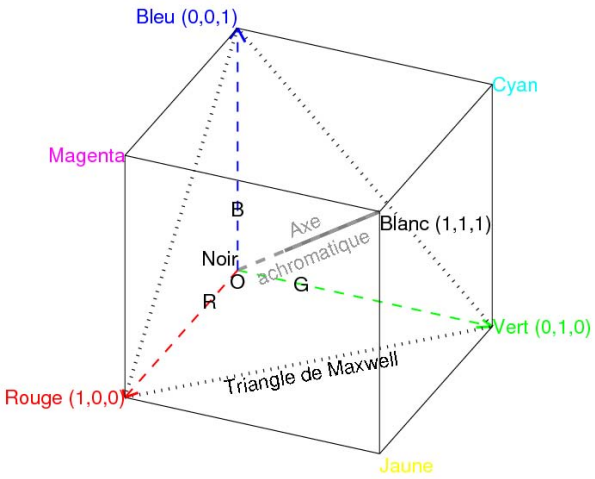


FIG. 3.5 – *Normalisation du système (R, G, B) : les axes principaux du système (R, G, B), l'axe achromatique en gris et le triangle de Maxwell.*

se caractérisent par une variation de luminance et non de chrominance des pixels représentant les zones ombrées dans les vêtements.

Il existe d'autres systèmes de primaires, comme le système (X, Y, Z) de la CIE et les systèmes $(R*, G*, B*)$, proposés par différents standards de définition de primaires. Nous ne les détaillerons pas dans notre étude et reportons le lecteur à la thèse de Vandenbroucke [Van00].

3.2.3.2 Systèmes luminance-chrominance

Après avoir présenté les systèmes de primaires issus du modèle physiologique de la vision humaine, nous nous intéressons à une autre famille de systèmes appelés systèmes luminance-chrominance. Nous avons déjà évoqué ces deux notions correspondant respectivement à l'intensité du stimulus de couleur et à sa chrominance proprement dite. Les composantes d'un système de luminance-chrominance sont évaluées à partir des composantes trichromatiques (R, G, B) par une transformation qui dépend de la nature du système. Cette transformation peut être linéaire (une matrice de passage est alors employée) ou non linéaire. On distingue ainsi :

- Les systèmes perceptuellement uniformes [Van00] comme le système (U, V, W) : ceux-ci possèdent une métrique qui permet d'établir une correspondance

entre la perception humaine de la différence entre deux couleurs et la mesure d'une distance entre deux points-couleur représentés dans ce système.
- Les systèmes de télévision [Van00] : ceux-ci sont utilisés pour la transmission des signaux de télévision. Ils séparent l'information de chrominance et celle de luminance. Il existe deux principaux systèmes de télévision : le système (Y', I', Q') correspondant à la norme NTSC et le système (Y', U', V') correspondant à la norme PAL.

D'autres systèmes appartiennent à cette famille, comme les systèmes antagonistes qui tentent de reproduire le modèle de la théorie des couleurs opposées de Hering, ou encore le système de Carron [Car95].

Dans le cadre de notre problématique, l'intérêt de ces systèmes luminance-chrominance est qu'ils isolent sur un canal l'information de luminance. Ceci permet d'exploiter uniquement les canaux de chrominance en vue de comparer le contenu de deux images.

3.2.3.3 Systèmes perceptuels

Après avoir décrit les systèmes luminance-chrominance qui permettent de caractériser une couleur indépendamment de la luminosité, nous présentons ici les système dits perceptuels car ils sont basés sur trois notions qui décrivent notre perception subjective de la couleur [Van00] :
- la luminance,
- la teinte,
- la saturation.

La luminance, comme nous l'avons déjà vu, correspond à la notion de clair ou de foncé. Elle correspond à l'intensité fournie par le stimulus couleur perçu. La teinte définit une couleur comme rouge, vert, bleu mais aussi jaune, orange ou violet. Elle correspond à la longueur d'onde dominante d'un stimulus de couleur, c'est-à-dire la longueur d'onde pour laquelle l'énergie correpondante est la plus élevée. Le blanc, le noir et les nuances de gris de l'axe achromatique sont des couleurs qui n'ont pas de teinte. La saturation, enfin, est une grandeur qui permet d'estimer le niveau de coloration d'une teinte, indépendamment de la luminance. Elle se décrit par les adjectifs vif, pâle ou terne.

Les systèmes perceptuels se différencient d'abord par le système de primaires donc ils sont déduits. Il existe également deux sous-familles de systèmes de re-

présentation qui se distinguent par la manière dont les coordonnées d'un point-couleur s'expriment :

- Les systèmes de coordonnées polaires ou cylindriques.
- Les systèmes humains de perception de la couleur. Ceux-ci sont évalués directement à partir des composantes trichromatiques d'un système de primaires.

La figure 3.6 représente le système (L, T, S) exprimé en coordonnées cylindriques.

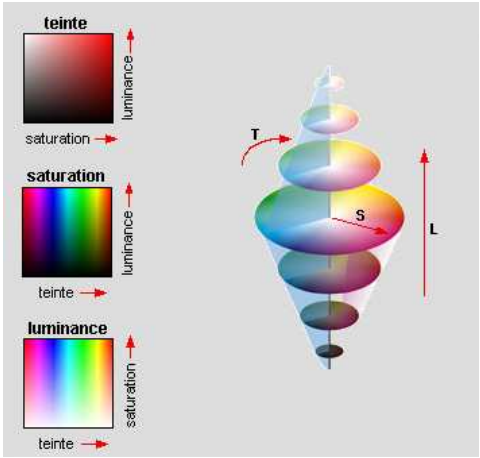


FIG. 3.6 – *Système de représentation* (L, T, S)

La transformation du système (R, G, B) vers le système (L, T, S) isole la luminance dans un canal, de la même manière que le fait un système luminance-chrominance. Par ailleurs, les composantes teinte et saturation sont calculées par les équations suivantes :

$$\begin{cases} L = \frac{R+G+B}{3} \\ S = 1 - \frac{\min(R,G,B)}{L} \\ si\ G > B : T = \arccos\left(\frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{(R-G)^2+(R-G)(G-B)}}\right) \\ si\ G < B : T = 2\pi - \arccos\left(\frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{(R-G)^2+(R-G)(G-B)}}\right) \end{cases} \tag{3.2}$$

Toutefois la saturation est instable lorsque les niveaux des trois composantes R, G et B sont faibles. Cependant, ceci ne représente pas un inconvénient dans le cadre de notre travail puisque les pixels caractérisés par des couleurs sombres correspondent à l'arrière-plan de la scène et ont été préalablement supprimés lors des pré-traitements.

Un autre système basé sur les composantes teinte et saturation est le système (H, S, V) (pour *hue,saturation,value*).

$$\begin{cases} V = max(R, G, B) \\ S = \frac{V - \min(R,G,B)}{V} \\ si\ V = R,\ H = \frac{G-B}{V-min(R,G,B)} \\ si\ V = G,\ H = 2 + \frac{B-R}{V-min(R,G,B)} \\ si\ V = B,\ H = 4 + \frac{R-G}{V-min(R,G,B)} \end{cases} \tag{3.3}$$

Ici encore, l'intérêt de ces systèmes de représentation est qu'ils isolent l'information de luminance dans un canal. Les niveaux de teinte et de saturation, représentés sur deux autres canaux, sont peu sensibles aux variations d'intensité. Il est ainsi judicieux de les exploiter dans le cadre d'une comparaison d'images à partir des composantes couleur.

3.2.3.4 Système de Gevers et Smeulders

Gevers et Smeulers [GS99a] proposent un système de représentation $(l1, l2, l3)$ calculé à partir du système (R, G, B), et détaillé dans [Mus05]. Gevers et Smeulders ont montré que ce système est invariant par rapport à la position de l'objet dans la scène, à son orientation, à la direction incidente de l'éclairage et à l'intensité de l'illuminant utilisé [GS99b]. Les composantes d'un point-couleur sont calculées à partir des équations suivantes :

$$\begin{cases} l1 = \frac{(R-G)^2}{(R-G)^2+(R-B)^2+(G-B)^2} \\ l2 = \frac{(R-B)^2}{(R-G)^2+(R-B)^2+(G-B)^2} \\ l3 = \frac{(G-B)^2}{(R-G)^2+(R-B)^2+(G-B)^2} \end{cases} \tag{3.4}$$

3.2.3.5 Systèmes retenus

Nous avons décidé de coder les couleurs des pixels dans les trois espaces de représentation suivants : les systèmes (r, g, b), (L, S, T) et (l_1, l_2, l_3) de Gevers et Smeulders. En effet, dans ces espaces de représentation, les variations locales d'intensité sont supposés être moins importantes.

Dans le système de primaires (r, g, b) normalisé ainsi que dans le système (l_1, l_2, l_3), la troisième composante s'exprime en fonction des deux autres : elle représente donc une information redondante. Enfin, le système perceptuel (H, S, V)

isole dans le canal de luminance les informations que nous ne désirons pas exploiter.

Dans les trois cas de figure, nous pouvons donc nous limiter à l'exploitation de deux canaux parmi les trois : (r, g), (H, S), et (l_1, l_2).

3.2.4 Définition mathématique d'une image numérique couleur

Maintenant que nous avons dégagé une vue d'ensemble non exhaustive des différents systèmes de représentation de la couleur qui peuvent être utilisés dans le cadre de notre travail, nous pouvons définir précisément l'image numérique couleur. Nous basons cette définition sur le système de représentation (R, G, B) car c'est dans cet espace que sont acquises les images par la caméra.

Toutefois, la manière de représenter une image numérique sur plusieurs canaux reste valable quel que soit le système de représentation de la couleur. Comme nous l'avons précisé, nous utilisons d'autres espaces pour nos travaux.

Une image numérique couleur est acquise par une caméra vidéo couleur puis numérisée par un ordinateur via une carte d'acquisition. Une hypothèse simplificatrice consiste à considérer que l'image est spatialement bi-dimensionnelle alors qu'elle résulte en réalité de la projection d'une scène observée tri-dimensionnelle. Nous avons vu que la couleur de chaque pixel est représentée par superposition des primaires rouge, vert et bleu.

Une image numérique couleur $\mathbf{I}$ est représentée par plusieurs signaux échantillonnés bidimensionnels à support et à valeurs bornées que nous notons $\mathbf{I}(x, y)$. Dans le plan image de taille $X \times Y$, x et y représentent les coordonnées en colonnes et en lignes du pixel P, avec $[x, y] \in \mathbb{N}^2, 0 \leq x \leq X - 1$ et $0 \leq y \leq Y - 1$.

$\mathbf{I}(x, y)$ représente la mesure de la couleur de l'élément de surface projeté sur le pixel de coordonnées (x, y) dans l'image. Ainsi dans le système (R, G, B) les trois niveaux des composantes trichromatiques R, G et B du pixel de coordonnées (x, y) sont représentés par $I^R(x, y), I^G(x, y)$ et $I^B(x, y)$.

La couleur d'un pixel est représentée par les trois composantes trichromatiques rouge, vert et bleu. Ces composantes sont quantifiées sur N niveaux, avec généralement $N = 256$. Les niveaux possibles s'étalent donc sur l'intervalle $[0..255]$,

le blanc de référence correpondant à la valeur $(255, 255, 255)$.

3.3 Comparaison d'images couleur par comparaison de signatures

Dans ce paragraphe, nous donnons une définition plus formelle d'une signature d'image calculée à partir de la notion d'image numérique présentée dans le paragraphe précédent.

Nous décrivons d'abord le type de signatures que nous avons retenu, basé sur la notion d'attributs. Ces signatures sont formées en regroupant des attributs et permettent ainsi de caractériser le contenu d'une image couleur. La comparaison des contenus de deux images passe alors par la comparaison des signatures des deux images. Nous présentons différents types d'attributs selon qu'ils caractérisent les formes, les couleurs ou les textures et argumentons le choix de ceux que nous avons retenus.

Nous présentons ensuite les attributs qui caractérisent les couleurs d'une image et sont regroupés dans une structure appelée histogramme couleur, puis différentes variantes de la notion d'histogramme qui prennent en compte la répartition des couleurs dans l'image, qu'elle soit globale ou locale.

Puis nous terminons par les attributs qui permettent de caractériser les textures présentes dans une image et qui sont regroupés dans une structure appelée matrice des co-occurrences.

3.3.1 Critère de choix des attributs

3.3.1.1 Définition

Les attributs d'une image correspondent à des statistiques calculées à partir des couleurs des pixels formant cette image, et qui caractérisent son contenu. Ces attributs servent ainsi à décrire les formes ou les couleurs présentes dans l'image. Ils sont regroupés sous la forme d'une signature qui décrit le contenu de l'image selon différentes caractéristiques morphométriques ou colorimétriques. Dans le cas d'une signature de séquence d'images, ces attributs peuvent être également ceux associés au mouvement.

3.3.1.2 Différents types d'attributs

Il existe plusieurs types d'attributs selon les caractéristiques qu'ils mettent en évidence. Ainsi, on peut distinguer :

- les attributs *morphométriques* qui sont déterminés à partir des régions extraites d'une image,
- les attributs de couleur qui caractérisent la distribution des couleurs des pixels formant l'image,
- les attributs de texture qui caractérisent l'interaction spatiale entre les pixels de l'image.

Une signature peut combiner plusieurs types d'attributs dont l'interprétation est différente. Il est ainsi possible de caractériser une image en fonction des formes qu'elle contient mais aussi des couleurs et des textures.

Nous avons choisi d'étudier uniquement les attributs de couleur et de textures car, comme nous l'avons déjà évoqué, l'analyse des formes des régions n'est pas appropriée à la comparaison de personnes en mouvement.

3.3.2 Histogrammes couleur

Les attributs de couleur qui sont le plus souvent utilisés pour caractériser le contenu d'une image sont regroupés dans l'histogramme couleur de cette image. Cette signature contient peu d'informations pertinentes car elle ne décrit que la *distribution* des couleurs dans l'image : chaque attribut représente la probabilité de trouver, dans l'image, un pixel d'une couleur donnée. C'est pourquoi de nombreux auteurs ont proposé d'enrichir cette signature avec la *répartition* spatiale des couleurs dans l'image. Nous présentons donc également quelques exemples de ces variantes.

3.3.2.1 Histogrammes couleur dans le système (R, G, B)

A partir d'une image couleur, on peut caractériser la distribution colorimétrique de ses pixels par l'*histogramme couleur normalisé* $H[I]$ d'une image I [SB91b].

Définition 2 (Histogramme couleur normalisé) *L'histogramme couleur d'une image I est une structure tridimensionnelle $H[I]$ composée de $N \times N \times N$ cellules, quand chaque composante couleur est quantifiée sur N niveaux.*

La cellule $H[I](x, y, z)$ contient la probabilité de trouver dans l'image un pixel caractérisé par la composante rouge égale à x, la composante verte égale à y, et

la composante bleue égale à z. Elle est obtenue en divisant le nombre de pixels dont le point-couleur est égal à (x, y, z) par le nombre total de pixels.

L'histogramme est insensible au changement de résolution spatial, ainsi qu'à la rotation et à la translation des objets représentés dans l'image.

L'utilisation de l'histogramme normalisé est intéressante en raison de son lien avec les probabilités : en effet, le contenu de chaque cellule d'un histogramme représente une probabilité d'apparition dans l'image de la couleur correspondante. La somme des cellules d'un histogramme normalisé vaut donc 1.

La signature n'est ici composée que de l'histogramme couleur normalisé. Pour comparer deux histogrammes normalisés $H[I]$ et $H[J]$ caractérisant deux images contenant chacune un seul objet, Swain et Ballard [SB91b] calculent leur intersection selon la formule suivante :

Définition 3 (Intersection d'histogrammes normalisés)

$$Inter(H[I], H[J]) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} \min(H[I](x, y, z), H[J](x, y, z)) \qquad (3.5)$$

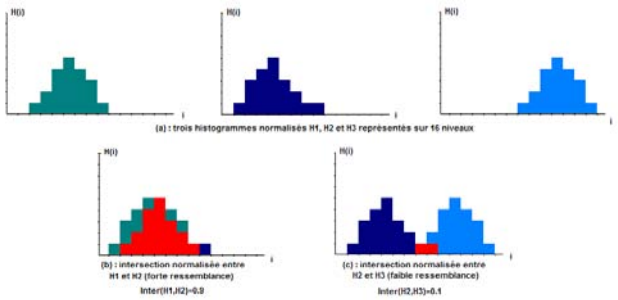


FIG. 3.7 – *L'intersection entre deux histogrammes est représentée par la partie en rouge : deux histogrammes similaires (b) et deux histogrammes très différents (c). Pour simplifier l'illustration, les histogrammes présentés sont monodimensionnels.*

La comparaison d'images par intersection d'histogrammes normalisés fournit une mesure de similarité : une mesure proche de 1 sera interprétée comme le cas où les contenus des images sont similaires en termes de couleurs tandis qu'une valeur proche de 0 indique que les contenus des images sont différents (voir figure 3.7).

Stricker et Swain évaluent les performances atteintes par la comparaison d'histogrammes colorimétriques pour l'indexation et la recherche dans des bases d'images hétérogènes [SS94]. Ils ont montré que les résultats obtenus par comparaison d'histogrammes couleur sont meilleurs que les résultats obtenus par comparaison d'histogrammes de luminance.

Cependant, la mesure fournie par le calcul d'intersection d'histogrammes entre deux images est à manier avec précaution car l'interprétation est une probabilité que les contenus soient identiques. Ainsi, si l'on mélange aléatoirement les pixels d'une image, l'histogramme n'est pas modifié, bien que l'image ne représente plus le même objet.

3.3.2.2 Histogrammes couleur dans d'autres systèmes de représentation

Nous avons vu dans le paragraphe précédent que le système (R, G, B) de représentation de la couleur est très sensible aux variations locales de luminance. Nous nous intéressons donc maintenant à l'utilisation d'histogrammes couleur dans d'autres espaces couleur.

Drew, Wei et Li[DWL98b], [DWL98a] proposent d'exploiter l'intersection d'histogrammes invariants aux changements d'éclairage. En normalisant les canaux colorimétriques, il est possible de caractériser chaque image par un histogramme bidimensionnel de chromacités, calculé à partir de deux[1] canaux du système (r, g, b) normalisé.

C'est par une approche similaire que nous travaillerons d'après les canaux de teinte et de saturation du système (H, S, V), pour ne pas tenir compte de la luminance.

Gevers et Smeulers[GS99b],[GS99a], calculent l'histogramme couleur dans le système de représentation $(l1, l2, l3)$. Ce dernier est invariant par rapport à la position de l'objet dans la scène, à son orientation, à la direction de l'illuminant et à son intensité, comme nous l'avons évoqué dans le paragraphe précédent. Pour une étude comparative plus approfondie des histogrammes de couleurs invariants à d'autres paramètres d'acquisition qui n'entrent pas dans le cadre de nos travaux,

[1] Rappelons que dans le système (r, g, b) normalisé, seuls deux canaux sont nécessaires puisque le troisième s'exprime en fonction des deux premiers.

nous renvoyons le lecteur à [GS996].

3.3.2.3 Autres attributs basés sur la distribution couleur

D'autres attributs qui mesurent les distributions de couleur sous-jacente à une image sont intéressantes en raison de leur compacité, comme celles proposées par Deng [DMK+01], ou Krishnamachari [KAM00].

Deng introduit la notion d'*histogramme de couleurs dominantes* en considérant d'abord que seul un faible nombre de couleurs sont présentes dans une image [DMK+01]. Cet histogramme de couleurs dominantes ne caractérise que les couleurs majoritaires, c'est-à-dire celles qui représentent un certain pourcentage de pixels dans l'image. La taille mémoire occupée par la signature formée de l'histogramme de couleurs dominantes est ainsi considérablement réduite, comparée à celle d'un histogramme couleur "classique".

Krishnamachari décrit un histogramme couleur de manière qualitative plutôt que quantitative à l'aide de *histogramme binaire compact* [KAM00]. Tandis qu'une cellule d'un histogramme fournit la probabilité d'apparition d'une couleur dans l'image à l'aide d'une valeur réelle comprise entre 0 et 1, la cellule d'un histogramme binaire ne donne qu'une valeur binaire qui indique si la couleur correspondante est présente ou non dans l'image. Puisqu'un seul bit est nécessaire pour caractériser la présence ou l'absence de chaque couleur, la taille mémoire occupée par l'histogramme est réduite de manière considérable.

Toutefois le principal inconvénient des histogrammes basés sur la distribution des couleurs est qu'ils ne reflètent pas la répartition des couleurs dans l'image. Les attributs que nous venons de décrire ne représentent que les probabilités d'apparition des couleurs des pixels dans l'image. C'est pourquoi d'autres attributs ont été développés afin de prendre en compte la répartition spatiale des couleurs et de former des signatures plus pertinentes. Grâce à ces attributs, il est possible d'analyser les textures présentes dans une image.

3.3.3 Interactions spatiales entre pixels

3.3.3.1 Histogrammes des pixels cohérents / non cohérents

Pass [PZM96] propose un attribut basé sur une classification des pixels en deux classes : les pixels dits *cohérents* et les pixels dits *non cohérents*. Remarquons que

les proportions de pixels cohérents et de pixels incohérents dépendent de l'espace de représentation de la couleur.

Les pixels cohérents sont ceux pour qui les points-couleur d'un nombre minimal de pixels voisins sont proches au sens de la norme L_2 (distance euclidienne). Le nombre minimum de voisins et le seuil sur la distance sont deux paramètres ajustés par l'utilisateur. Ainsi, les pixels cohérents sont ceux qui sont centrés dans une zone homogène en couleur et les pixels non cohérents dans une zone de couleurs hétérogènes. Deux histogrammes forment la signature de l'image : l'histogramme des pixels cohérents et celui des pixels incohérents. Deux intersections d'histogrammes sont alors nécessaires pour comparer les images.

3.3.3.2 Joint histogram

Pass introduit également le concept de *joint-histogram* [PZ99] dont chaque cellule indique le nombre de pixels ayant des caractéristiques spatio-colorimétriques identiques : la couleur d'un pixel, la norme du gradient couleur calculé en ce pixel, la densité des contours calculés en ce pixel dans un voisinage choisi, le nombre de pixels voisins dont l'intensité dépasse un certain seuil fixé par l'utilisateur, et enfin le rang du pixel, c'est-à-dire le nombre de pixels voisins dont l'intensité est inférieure à sa propre intensité. La fonction de comparaison employée est la distance au sens de la norme L_1 entre les joint-histograms.

3.3.3.3 Longueurs de plages

Par ailleurs, Chan [CC01] propose une variante de l'histogramme couleur comportant des informations sur la répartition globale des couleurs grâce aux *run-length features* ou *longueurs de plages*. Ceux-ci dénombrent les pixels de couleurs proches qui sont connexes selon une direction donnée. Ces attributs sont toutefois sensibles au nombre de niveaux utilisés pour quantifier les composantes couleur.

3.3.3.4 Ruptures de couleurs

Chan analyse les interactions locales entre les pixels grâce à un attribut qui mesure les ruptures couleur [CC04]. Chaque ligne de l'image est parcourue de gauche à droite selon la direction horizontale. Lorsque le point-couleur d'un pixel est différent de celui du pixel précédent, un compteur correspondant à ce point-couleur est incrémenté. Le procédé est répété selon les directions verticales et diagonales, un ensemble de compteurs étant associé à chaque direction.

3.3.3.5 Graphes de Park

Enfin, Park [PYL99] propose d'employer des graphes pour caractériser le contenu des images : le *modified color adjacency graph* et le *spatial variance graph*. Ces graphes non orientés représentent les relations d'adjacence entre pixels en fonction de leurs couleurs.

Chaque noeud du modified color adjacency graph correspond à un point-couleur présent dans l'image et contient le nombre de pixels caractérisés par cette couleur. Les arêtes entre deux noeuds indiquent le nombre de paires de pixels voisins dont les couleurs sont représentées par ces noeuds.

Le spatial variance graph indique, pour chaque noeud, la variance des coordonnées spatiales des pixels associés aux points-couleurs des noeuds. Une arête reliant deux noeuds indique la variance des coordonnées spatiales des pixels associés aux points-couleurs de l'un de ces noeuds.

Park étend l'intersection d'histogrammes de Swain [SB91b] pour comparer ces graphes deux-à-deux. Bien que les résultats obtenus par cette méthode en reconnaissance d'objets statiques soient satisfaisants, l'implantation de ces graphes ne peut pas être retenue en raison des temps de réponse trop importants qu'elle engendre pour une quantité importante d'images.

Puisque notre travail consiste à exploiter l'information apportée par les vêtements des personnes observées, il semble judicieux de tenir compte simultanément de la distribution des couleurs et de l'interaction spatiale entre les pixels. Toutefois, les calculs et comparaisons des extensions de l'histogramme couleur que nous venons de voir sont coûteuses en temps de calcul et en espace mémoire. Notre attention s'est donc portée vers des méthodes plus simples à implanter et qui permettent toutefois de prendre en compte l'interaction spatiale entre les pixels à travers la notion de texture.

3.3.4 Attributs de texture

3.3.4.1 Notion de texture

Il n'existe pas de définition précise de la texture[2]. La définition du "Petit Robert"[RDR04] est : *Arrangement, disposition des éléments d'une matière. Agencement des parties, des éléments (d'une oeuvre, d'un tout).*

[2]Le terme vient du *latin* textura qui signifie tisser, et désigne en vieux Français l' "arrangement" des fils d'un tissage.

En traitement d'images, on parle de primitives arrangées selon des règles particulières. On peut alors distinguer les notions de microtexture et de macrotexture. Prenons l'exemple d'un mur de briques : d'une part, la texture d'une brique de construction est formée par l'arrangement des pixels représentant cette brique. Les pixels sont les éléments structurants et on parlera de microtexture. D'autre part, la texture du mur est formée par l'arrangement bien spécifique de ces briques qui, dans ce cas, sont les éléments structurants. On parle alors de macrotexture.

Dans le cadre de notre travail, nous avons privilégié l'analyse des microtextures : nous analysons les interactions entre pixels sans chercher à en dégager des motif précis. En effet, nous n'avons aucune information a priori sur les éventuels motifs des vêtements des personnes observées par les caméras.

3.3.4.2 Méthodes invariantes aux transformations géométriques

Zhang et Tan [ZT02] classifient en trois catégories les méthodes d'analyse de texture invariantes, c'est-à-dire dont les résultats ne changent pas si l'on applique à l'image une translation, une rotation, une transformation de la perspective ou une transformation affine. Ces trois catégories sont :

- les méthodes statistiques,
- les méthodes basées sur un modèle,
- les méthodes structurelles.

Nous écartons les méthodes structurelles car, de nouveau, nous ne cherchons pas à dégager des motifs précis. Pour la même raison, nous écartons les méthodes basées sur un modèle particulier et nous nous focalisons sur les méthodes statistiques.

3.3.4.3 Approches statistiques : matrices de co-occurrences

L'une des approches statistiques de l'analyse de la texture est basée sur les matrices de co-occurrences [Har79]. Bien que les images que nous traitons sont en couleur, nous commençons par expliquer le principe de création des matrices de co-occurrences lorsqu'elles sont obtenues à partir d'images monochromatiques[3].

Définition 4 (matrice de co-occurrences) *La matrice de co-occurrences d'une image monochromatique I est notée* $\mathbf{Q}[I]$. *La cellule*

[3] Une image monochromatique, dite *en niveaux de gris* ne possède qu'un seul canal : celui de luminance

$Q[I](x, y)$ contient le nombre de fois qu'un pixel dont le niveau est égal à x, est voisin d'un pixel dont le niveau est égal à y.

3.3.4.4 Généralisations

Les *corrélogrammes* [RHC97] sont calculés par la même procédure que celle qui est utilisée pour construire la matrice de co-occurrences, mais cette fois en faisant varier la distance qui sépare le pixel considéré du pixel voisin de la valeur 1 à une valeur fixée. Quand la valeur de la distance est égale à 1, les valeurs des cellules du corrélogramme sont celles des cellules de la matrice de co-occurrences.

Bien que la signature composée des corrélogrammes semble intéressante, sa structure est volumineuse. Puisque nous travaillons sur un grand nombre d'images, nous n'avons pas exploité ces attributs.

3.3.4.5 Matrice de co-occurrences chromatiques

Nous souhaitons prendre en compte les informations apportées d'une part par les couleurs et d'autre part par les textures. Nous nous sommes donc intéressés à une approche statistique globale des caractéristiques de textures couleur telle que celle proposée par Skrzypniak [SMP00] et améliorée par Muselet [MMKP02]. Nous présentons ici les attributs de texture couleur que nous avons employés, qui combinent l'exploitation des composantes colorimétriques d'une image et l'analyse de la texture initiée par Haralick et basée sur les co-occurrences.

Définition 5 (matrice de co-occurrences chromatiques) *La matrice de co-occurrences chromatiques d'une image I, et relative aux canaux r et g est notée $M_{rg}[I]$. La cellule $M_{rg}[I](x, y)$ contient le nombre de fois qu'un pixel dont de niveau de la composante r égal à x est dans le voisinage 3×3 d'un pixel dont le niveau de la composante g est égal à y.*

Définition 6 (matrice de co-occurrences chromatiques normalisée) *La matrice de co-occurrences chromatiques normalisée d'une image est notée $\mathbf{C}_{rg}[I]$. La cellule $C_{rg}[I](x, y)$ contient le nombre de fois qu'un pixel dont le niveau de la composante r égal à x est voisin d'un pixel dont le niveau de la composante est égal à y, **divisé par le nombre total de co-occurrences**. Chaque cellule représente donc une probabilité d'apparition de niveau d'un pixel, connaissant celui de ses voisins.*

Pour mesurer la ressemblance entre les contenus de deux images couleur I et J, on applique la méthode proposée par Muselet [Mus05]. La similarité entre les signatures, composées de matrices de co-occurrences chromatiques normalisées, est obtenue en calculant l'intersection des matrices de co-occurrences normalisées en tenant compte cette fois des textures présentes dans les images.

L'intersection entre deux matrices de co-occurrences normalisées $C_{rg}[I]$ et $C_{rg}[J]$ s'obtient de la manière suivante :

$$Inter(C_{rg}[I], C_{rg}[J]) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} min(C_{rg}[I](x,y), C_{rg}[J](x,y))$$

.

3.4 Signature retenue

Dans le premier paragraphe de ce chapitre, nous avons retenu les plans couleur (r,g), (H,S) et (l_1,l_2) car ils s'avèrent peu sensibles aux faibles variations locales d'intensité.

Nous proposons d'exploiter la notion de co-occurrences chromatiques dans chacun de ces plans couleur.

Par exemple dans le plan (r,g), la signature de texture couleur de l'image I est formée des trois matrices de co-occurrences chromatiques normalisées $C_{rr}[I]$, $C_{rg}[I]$, $C_{gg}[I]$. La matrices de co-occurrences $C_{gr}[I]$ n'est pas utilisée car elle n'apporte aucune information supplémentaire. En effet elle peut être obtenue par une simple transposition de la matrice $C_{rg}[I]$.

Ainsi, pour mesurer la ressemblance entre deux images I et J, nous calculons les signatures ϕ suivantes, chacune composée des trois matrices de co-occurrences chromatiques normalisées :

- $\phi[I] = \{C_{rr}[I], C_{rg}[I], C_{gg}[I]\}$
- $\phi[J] = \{C_{rr}[J], C_{rg}[J], C_{gg}[J]\}$

puis nous mesurons les trois intersections normalisées suivantes :

- $Inter_{rr}(C_{rr}[I], C_{rr}[J])$,
- $Inter_{rg}(C_{rg}[I], C_{rg}[J])$,
- $Inter_{gg}(C_{gg}[I], C_{gg}[J])$.

Enfin, on calcule la moyenne des trois intersections pour comparer les contenus des deux images I et J :

$$Sim(I,J) = Sim(\phi[I], \phi[J]) = \frac{Inter_{rr} + Inter_{rg} + Inter_{gg}}{3} \tag{3.6}$$

Cette mesure de similarité peut aussi être évaluée en codant les couleurs les plans chromatiques (H,S) et (l_1,l_2).

Afin de mesurer la ressemblance entre les contenus de deux images, nous mesurons la similarité entre leurs signatures.

D'autres mesures de comparaison comme les mesures de distance sont employées dans [PRT99], dans ce cas on utilise la norme $L1$ (distance de Manhattan) ou la norme $L2$ (distance euclidienne).

3.5 Conclusion

Après avoir présenté le processus physique de formation des couleurs, nous avons vu comment l'être humain perçoit celles-ci à partir de la rétine. La physiologie de la perception de la couleur a fourni les bases de la théorie trichromatique qui permet de représenter la couleur.

Nous avons vu que le système de primaires (R,G,B) est très sensible aux variations d'intensité de l'illuminant ainsi qu'aux ombres causées par le déplacement d'une personne et les plis de ses vêtements : en effet nous observons d'importantes variations de la luminance sur des zones pourtant homogènes.

Ces variations sont un phénomène inévitable lorsque les images représentent des personnes en mouvement. Nous avons donc présenté différents systèmes plus robustes face à ces variations. Dans le cadre de nos travaux, nous retenons les plans couleur (r,g), (H,T) et (l_1,l_2).

Nous avons ensuite passé en revue un certain nombre de signatures d'images utilisées pour caractériser leur contenu. Nous nous sommes intéressés aux signatures qui permettent de décrire ce contenu en termes de distribution des couleurs et nous avons vu en quoi ceci constituait une information insuffisante pour discriminer convenablement deux images.

Nous avons enfin motivé le choix des matrices de co-occurrences chromatiques qui caractérisent les textures couleur représentées dans une image.

Nous avons également présenté une fonction de comparaison de deux images à l'aide de leurs signatures composées des matrices de co-occurrences normalisées.

Nous proposons maintenant de faire état des travaux d'ores et déjà accomplis dans le domaine de la comparaison de séquences d'images.

Chapitre 4

Comparaison de séquences d'images couleur

4.1 Introduction

Nous avons vu, d'après le cahier des charges détaillé dans le premier chapitre, que notre problématique nous amène à comparer les contenus de séquences-personnes par l'intermédiaire de leurs signatures.

Afin de respecter les contraites de temps de réponse, le temps de calcul nécessaire au calcul de deux signatures et à leur comparaison doit être du même ordre de grandeur que la durée d'acquisition d'une séquence-personne, c'est-à-dire de quelques secondes seulement.

Il est donc souhaitable de réduire, d'une part la taille mémoire nécessaire pour représenter ces signatures et, d'autre part, le temps de calcul nécessaire à leur comparaison.

Dans le chapitre précédent, nous avons présenté plusieurs méthodes de comparaison de deux images en calculant une distance entre leurs signatures colorimétriques. Les images-personnes sont des matrices de pixels et le caractère spatial de ces données mérite d'être pris en considération : il est donc possible d'enrichir l'information apportée par la probabilité d'apparition des couleurs caractérisant ces pixels-personnes par une probabilité de connexité entre pixels-personnes, en employant les matrices de co-occurrences chromatiques.

Ce chapitre concerne maintenant la comparaison du contenu de séquences d'images par l'utilisation d'une mesure de ressemblance entre leurs signatures.

Dans le premier paragraphe, nous introduisons les notations que nous emploierons par la suite car celles-ci sont particulièrement adaptées à l'analyse de séquences d'images.

Dans le deuxième paragraphe, nous abordons les problèmes posés par le caractère temporel de ces séquences. La comparaison de toutes les images d'une séquence requête à toutes les images d'une séquence candidate est très coûteuse en ressources, autant en termes de quantité de mémoire que de temps de calcul.

C'est pourquoi nous présentons par la suite trois familles de signatures de séquences d'images. Deux d'entre elles résument de différentes manières l'information apportée par les signatures de toutes les images d'une séquence, tandis que la troisième tient compte l'aspect dynamique des données. Pour chacune de ces familles de signatures, leurs points forts et leurs inconvénients sont analysés.

La première approche fait l'objet du troisième paragraphe : elle consiste à extraire des images-clés d'une séquence, c'est-à-dire un nombre restreint d'images représentatives. La signature de la séquence est alors constituée des signatures de ces images.

Les signatures de séquences dites *globales* sont présentées dans le quatrième paragraphe. Elles caractérisent les couleurs de l'ensemble des images de la séquence.

Enfin, les signatures dites *spatio-temporelles* sont abordées dans le cinquième paragraphe. Celles-ci tiennent compte des propriétés d'interaction entre les pixels dans le domaine spatial et également dans le domaine temporel.

4.2 Notations spécifiques à l'analyse de séquences d'images

Avant de commencer cet état de l'art consacré à la comparaison de séquences d'images, nous introduisons la notation qui sera employée par la suite dans ce document. En effet, cette notation est particulièrement adaptée à l'analyse de séquences d'images. Une notation similaire est d'ailleurs également employée par Ferman [FTM02] dont les travaux sont présentés dans ce chapitre.

4.2.1 Vecteurs d'attributs

Dans le chapitre précédent nous avons défini plusieurs objets mathématiques bidimensionnels (les matrices de co-occurrences) ou tridimensionnels (l'histogramme couleur) et qui servent à regrouper des attributs caractérisant une image.

Cependant ils peuvent tous être représentés sous la forme de vecteurs d'attributs monodimensionnels.

4.2.1.1 Méthode de construction

A partir d'une matrice de co-occurrences, on peut construire un vecteur d'attributs en parcourant toutes les cellules de la matrice selon un ordre pré-établi. Dans la mesure où cet ordre n'a aucune influence sur les traitements ultérieurs, du moment que celui-ci est toujours le même, nous avons choisi de parcourir la matrice ligne par ligne comme indiqué sur la figure 4.1.

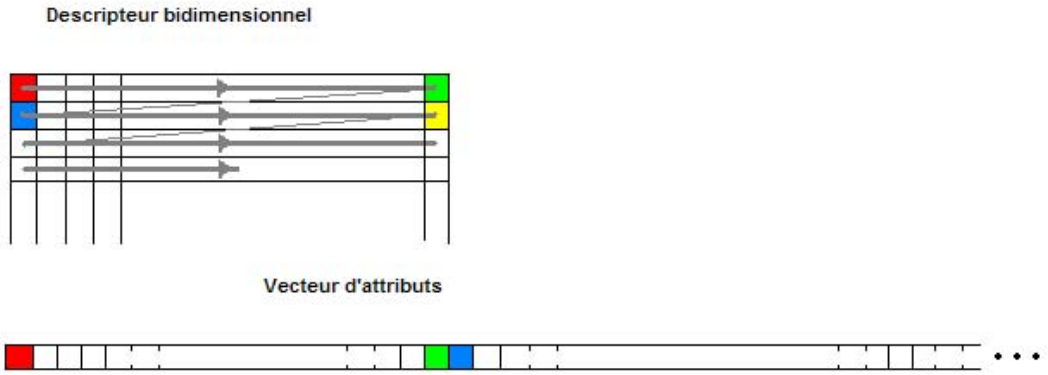


FIG. 4.1 – Représentation d'une matrice sous la forme d'un vecteur d'attributs

Il en va de même pour un histogramme couleur à trois dimensions, ainsi que pour toute structure multidimensionnelle.

4.2.1.2 Formules de passage

Un histogramme couleur calculé à partir de l'espace de représentation (R, G, B) est tridimensionnel. On note donc $H[I](x, y, z)$ une cellule de l'histogramme d'une image I.

Pour simplifier les notations nous pouvons considérer que l'histogramme couleur $H[I]$ d'une image I est composé de cellules $H[I](\mathbf{c})$ repérées par le point-couleur $\mathbf{c}$ dont les composantes sont $(R = x, G = y, B = z)$.

Si par exemple, cet histogramme couleur tridimensionnel contient $v = 256 \times 256 \times 256$ cellules, on peut donc remplacer la notation $H[I](x, y, z)$ par $H[I](w)$ avec $w = x + y \times 256 + z \times 256^2$.

4.2.2 Notations adoptées

Par la suite, nous considérons qu'une signature d'image est composée de un ou plusieurs vecteurs d'attributs. Nous adoptons la représentation monodimensionnelle, que ces signatures soient composées de l'histogramme couleur ou des matrices de co-occurrences chromatiques.

Le premier avantage de cette notation est d'être plus claire puisqu'un seul index est utilisé pour désigner une coordonnée de chaque vecteur.

Le second avantage de cette notation, corollaire au premier, est que celle-ci est uniformisée quel que soit le type d'objet mathématique employé (une matrice, ou une structure tridimensionnelle). En effet, les méthodes que nous emploierons sont appliquables quel que soit le type de vecteur d'attributs. Cependant, ceci ne prétend pas prouver qu'elles sont toujours pertinentes.

4.2.2.1 Signatures d'images

Lorsqu'il s'agit d'un vecteur d'attributs quelconque, celui-ci est noté $\mathbf{x}$ et v représente le nombre de composantes de ce vecteur.

Le vecteur d'attributs particulier qui correspond à l'histogramme couleur tridimensionnel H_{RGB} est noté $\mathbf{x}_{H_{RGB}}$ et le nombre de ses composantes $v_{H_{RGB}}$ vaut N^3, avec N le nombre de niveaux pour quantifier chaque canal de la couleur de l'image.

Le vecteur d'attributs qui correspond à la matrice de co-occurrences chromatiques C_{rr} est noté $\mathbf{x}_{C_{rr}}$ avec $v_{C_{rr}} = N^2$.

La signature ϕ d'une image I composée de trois vecteurs d'attributs, à savoir des trois matrices chromatiques calculées à partir des canaux r et g est notée :

$$\phi_{C_{(r,g)}}[I] = \{\mathbf{x}_{C_{rr}}[I], \mathbf{x}_{C_{rg}}[I], \mathbf{x}_{C_{gg}}[I]\}$$

4.2.2.2 Séquences de signatures d'images

Pour une séquence-personne $\mathcal{A}^1$ de $n_{A^{(1)}}$ images successives, la séquence des $n_{A^{(1)}}$ signatures des images est notée :

$$\Phi_{C_{(r,g)}}[\mathcal{A}^{(1)}] = \{\phi_{C_{(r,g)}}[\mathcal{A}_1^{(1)}], \phi_{C_{(r,g)}}[\mathcal{A}_2^{(1)}], \ldots, \phi_{C_{(r,g)}}[\mathcal{A}_{n_{A^{(1)}}}^{(1)}]\}$$

4.2.2.3 Indexation

Nous terminons par l'introduction d'une dernière précision concernant les index employés pour désigner une image parmi celles d'une séquence-personne de n images, et ceux qui le sont pour désigner la cellule d'un vecteur d'attributs de v composantes.

La lettre i indique l'index de l'image considérée de la séquence-personne, on a donc $i \in [1, n]$. La lettre j indique l'index de la cellule du vecteur d'attribut considéré, on a donc $j \in [1, v]$.

4.3 Difficultés rencontrées lors de la comparaison de séquences d'images

Il peut être envisageable d'exploiter les comparaisons de signatures d'images pour comparer des séquences d'images. Cependant, il est nécessaire que la signature d'une séquence occupe le moins de place mémoire possible. De plus, lorsqu'il s'agit de comparer deux séquences-personnes, les données présentent un caractère temporel qui est la cause de difficultés que nous exposons dans ce paragraphe.

4.3.1 Variations au sein d'une séquence de matrices de co-occurrences

Nous avons vu, dans le deuxième chapitre, que les séquences-personnes représentent le passage d'une personne qui se déplace, observée selon une vue de dessus. Le mouvement de balancier des bras et celui des jambes est pseudo-périodique. Une séquence de vecteurs d'attributs, comme celle qui correspond à la séquence des matrices de co-occurrences relative à deux canaux (r et g par exemple), présente donc des variations. La figure 4.2 représente cinq images extraites d'une séquence-personne, et qui ont été acquises à des instants bien séparés. On constate

que, selon la posture adoptée par la personne dans l'image, les jambes ne sont pas toujours visibles.

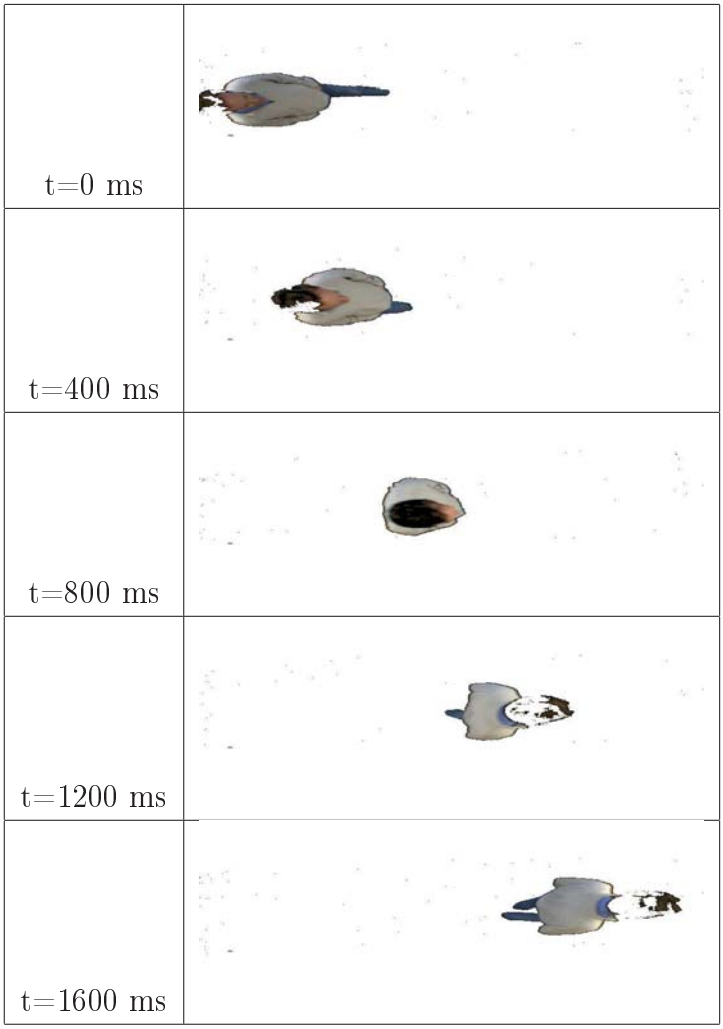


FIG. 4.2 – *Cinq images de la séquence $\mathcal{D}^{(1)}$, acquises à 400 ms d'intervalle.*

Puisque la matrice de co-occurrences est calculée à partir des informations présentes dans l'image, on peut supposer que les contenus des matrices évoluent au cours de la séquence. Afin de vérifier cette hypothèse, nous avons calculé les matrices de co-occurrences C_{rg} des cinq images de la figure 4.2, puis mesuré leurs similarités qui sont regroupées dans le tableau 4.1.

Nous constatons que les valeurs de ces mesures sont relativement faibles lorsque l'on compare une image dans laquelle les jambes de la personne sont visibles à une image dans laquelle on ne distingue que le buste et la tête de la

personne. Ceci met en évidence que les matrices de co-occurrences subissent de fortes variations au fil des images d'une même séquence.

4.3.2 De la comparaison d'images à la comparaison de séquences d'images

Sachant que l'on peut mesurer la ressemblance entre les contenus de deux images-personnes, la présence de variations observées dans une séquence de vecteurs d'attributs induit la question suivante : si l'on désire comparer une séquence d'images S_{Req} avec une séquence d'images S_{Can}, comment déterminer les couples formés d'une image de S_{Req} et d'une image de S_{Can}, dont on calculera la similarité, pour associer les images deux à deux et déterminer ainsi les similarités pertinentes ?

A ce problème d'ordre méthodologique s'ajoute un problème pratique : supposons que pour chaque image de S_{Req}, on décide de ne considérer l'image de S_{Can} qui lui ressemble le plus. Comment dans ce cas déterminer cette image la plus ressemblante, autrement qu'en calculant malgré tout les similarités entre chaque image de S_{Req} et toutes les images de S_{Can} ? Nous pouvons conclure qu'il n'est pas possible de déterminer ces couples les plus pertinents autrement qu'en examinant toutes les combinaisons possibles. Le nombre de comparaisons d'images à effectuer s'élève alors à $n_{S_{Req}} \times n_{S_{Can}}$.

TAB. 4.1 – Similarités entre les cinq matrices de co-occurrences C_{rg} des images de la figure 4.2

	t=0 ms	t=400 ms	t=800 ms	t=1200 ms	t=1600 ms
t=0 ms	1	0.779	0.693	0.734	0.686
t=400 ms		1	0.792	0.706	0.578
t=800 ms			1	0.740	0.623
t=1200 ms				1	0.771
t=1600 ms					1

4.4 Signatures d'images-clés

Une première méthode pour comparer deux séquences d'images consiste à résumer l'information contenue dans la séquence de signatures. Pour une séquence S de n images, ce résumé se fait simplement en retenant un nombre n' d'images-clés pour caractériser la séquence, avec $n' << n$ afin de réduire, d'une part la taille de la signature et d'autre part le temps de calcul nécessaire à la comparaison de deux signatures.

4.4.1 Principe

La signature d'une séquence d'images est alors composée des signatures des images-clés. Ces images sont considérées comme étant les plus représentatives de la séquence.

Le nombre d'images-clés est relativement faible, il est donc possible de déterminer, pour chaque image-clé requête, l'image-clé candidate qui lui est la plus proche en calculant les mesures de proximité entre toutes les images-clés de la séquence-personne requête et toutes celles de la séquence-personne candidate. Le nombre de comparaisons d'images nécessaires n'est pas aussi pénalisant que lorsque l'on désire comparer chaque image de la séquence requête à toutes les images de la séquence candidate.

Pour obtenir une mesure scalaire de proximité entre deux séquences d'images S_{Req} et S_{Can} caractérisées par des images-clés, et sachant qu'on dispose de mesures de proximité $prox$ entre chaque image-clé de S_{Req} et chaque image-clé de S_{Can}, Cheung propose de mesurer la proximité entre les deux séquences de la manière suivante [CZ00] :

Soit $p_{Can}(i'_{Req})$ le numéro de l'image-clé de S_{Can} la plus ressemblante à l'image-clé i'_{Req} de S_{Req}, et n'_{Req}, n'_{Can} le nombre d'images clés de S_{Req} et de S_{Can}. $p_{Can}(i'_{Req})$ s'exprime par :

$$p_{Can}(i'_{Req}) = \arg \max_{i'_{Can} \in S_{Can}} prox(i'_{Req}, i'_{Can}).$$

La proximité entre les séquences S_{Req} et S_{Can} s'exprime alors par :

$$Prox(S_{Req}, S_{Can}) = \frac{\left[\sum_{i'_{Req}=1}^{n'_{Req}} prox(i'_{Req}, p_{Can}(i'_{Req})) + \sum_{i'_{Can}=1}^{n'_{Can}} prox(p_{Req}(i'_{Can}), i'_{Can})\right]}{n'_{Req} + n'_{Can}} \quad (4.1)$$

Cela revient à calculer la moyenne des plus fortes proximités entre, d'une part, chaque image-clé de S_{Req} et l'image-clé de S_{Can} qui lui ressemble le plus, et d'autre part chaque image-clé de S_{Can} et l'image de S_{Req} qui lui ressemble le plus.

Cette procédure nécessite malgré tout une phase préliminaire de calcul de proximité entre toutes les images-clés de S_{Req} et toutes celles de S_{Can}, ce qui est possible compte tenu du faible nombre d'images-clés retenues pour les deux séquences.

4.4.2 Détermination des images-clés

Il existe, dans la littérature, de nombreuses techniques de détermination des images-clés d'une séquence [HJ99, GB00, UF99, DDAK99, CN00, CSL99].

On distingue tout d'abord les méthodes interactives des méthodes automatiques. Nous ne nous intéresserons pas aux méthodes interactives dans lesquelles un opérateur sélectionne les images qu'il juge représentatives, assisté ou non par la machine.

Parmi les méthodes automatiques, on peut distinguer plusieurs approches selon qu'elles prennent en compte ou non le caractère temporel des séquences.

4.4.2.1 Extraction d'images-clés selon un critère temporel

Une première approche est généralement employée dans le cas où les images-clés séparent ce qu'on appelle des *shots*[1] [ZQZ01]. Dans ce cas, la séquence d'images peut être divisée en sous-séquences, chacune étant bien distincte de l'autre du point de vue leurs contenus respectifs et la transition d'une sous-séquence à une autre étant significative.

Pour ce faire, on considère l'évolution d'un indicateur dans le temps [CI02] : chaque variation significative correspond à une transition entre deux sous-séquences qui sont, elles, représentées par une absence de variations significatives de cet indicateur pour plusieurs images consécutives. Une transition correspond à deux images successives dont les contenus sont significativement différents. Les images-clés sont extraites de chacune des sous-séquences obtenues, en conservant par exemple l'image qui est au centre de chaque sous-séquence.

[1] Shot : séquence d'images qui ne présente aucune rupture ou transition brusque, et qui est donc difficilement divisible en sous-séquences [LZFS02].

L'avantage de cette approche est que l'aspect temporel est conservé, mais un premier inconvénient apparaît si la séquence comporte des sous-séquences qui se ressemblent sans toutefois être consécutives : elles sont alors considérées comme deux sous-séquences distinctes, bien que les images représentatives de ces dernières sont vraisemblablement proches et constituent alors une redondance d'information.

Un autre inconvénient de cette approche concerne directement nos séquences-personnes qui ne sont pas divisibles en shots car elles sont déjà des shots : elles ne comportent aucune variation suffisamment significative pour que l'on puisse distinguer véritablement des transitions. On peut donc difficilement dégager les images-clés "les plus représentatives" par cette méthode.

Enfin, certains auteurs choisissent les images-clés d'une séquence selon un critère pré-établi : la première image [TAOS93], la dernière image, l'image de rang médian [GFT97], ou encore la première et la dernière image [LPE97]. Ceci ne semble pas être adapté aux séquences-personnes car il n'y a pas de lien direct entre le rang des images acquises et le contenu de ces images.

4.4.2.2 Extraction des images-clés par simplification de trajectoire

Une deuxième approche pour déterminer les images-clés d'une séquence consiste à représenter le vecteur d'attributs de chaque image I_i par un point dans l'espace de représentation dont la dimension est le nombre de ses attributs. Une séquence de vecteurs d'attributs est alors représentée par un nuage de points où chaque point correspond au vecteur d'attributs d'une image de la séquence. On peut relier les points correspondant aux images successives pour former une courbe qui décrit la trajectoire de ces vecteurs d'attributs.

DeMenthon propose de simplifier cette courbe pour former un polygone dont les sommets représentent les images-clés [DKD98]. La figure 4.4.2.2 montre un exemple de trajectoire dans un plan, formée par les points correspondant aux vecteurs d'attributs à deux composantes de chaque image d'une séquence (en bleu). Le polygone simplifié mettant en évidence les images-clés est représenté en rouge.

Cette méthode est relativement coûteuse en ressources de calcul. En effet,

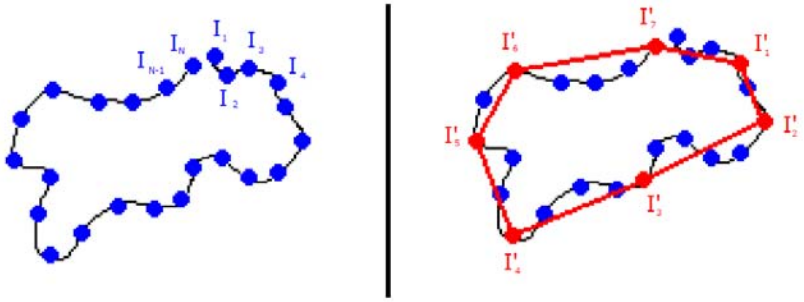


FIG. 4.3 – *Représentation des vecteurs d'attributs à deux composantes d'une séquence d'images sous forme de trajectoire dans le temps : (a) la courbe obtenue en reliant les n points représentés en bleu et correspondant aux images I_i ; (b) la simplification par un polygone met en évidence des images-clés I'_i correspondant aux points représentés en rouge.*

l'analyse de la trajectoire dans un espace multidimensionnel demande d'autant plus d'opérations que la dimension de l'espace est élevée. Il est également possible de travailler dans un espace de dimensions réduites [Ple03], mais dans ce cas, il est nécessaire de calculer la projection à appliquer à ces points. C'est cette opération qui est alors coûteuse en ressources.

4.4.2.3 Extraction d'images-clés par classification des images

Une troisième approche, enfin, consiste à considérer la matrice de proximités intra-séquence, regroupant les proximités entre, d'une part, toutes les images d'une séquence, et d'autre part ces mêmes images. Les mesures de proximité sont calculées à partir d'histogrammes couleur des images dans [DA00, HM00c, HM00a, HM00b].

Une matrice de proximités intra-séquence est nécessairement carrée, symétrique et définie positive. A partir de ce type particulier de matrices, il est possible de dégager les n' images les plus éloignées les unes des autres [Yah03].

Girgenson propose une méthode qui, à partir d'une matrice de proximités intra-séquence, permet de représenter les images dans un espace métrique et de regrouper en classes les images aux contenus similaires en estimant le centre de gravité de chaque classe [GB00]. La sélection des images-clés se fait de la manière

suivante : pour chaque classe, l'image la plus proche du centre de gravité constitue une image-clé. Dans ce cas, le nombre d'images-clés dépend du nombre de classes qui est prédéterminé. Ferman [FT97] adopte une démarche comparable en ne retenant que les n' classes les plus importantes, c'est-à-dire celles qui contiennent le plus grand nombre d'images.

Des approches similaires s'appuient sur la matrice de proximités intra-séquence mais sans fixer le nombre d'images-clés à extraire : en effectuant une classification non supervisée, l'algorithme détermine le nombre de classes donc le nombre d'images-clés représentatives [HM00d]. De même, Zhang et al. effectuent une classification non supervisée de toutes les images et sélectionnent, pour chaque classe, l'image dont le point est le plus proche de son centre de gravité [ZRHM98].

Ces approches sont d'autant plus efficaces que les images sont suffisamment différentes pour être représentées par des nuages de points bien séparés dans un espace de représentation. Ceci n'est malheureusement pas le cas des images d'une séquence-personne.

Par ailleurs, ces méthodes s'avèrent également coûteuses en termes de ressources : il est non seulement nécessaire de calculer la matrice de proximités intra-séquences, mais aussi de réaliser une opération de classification avant de pouvoir sélectionner les images-clés.

4.4.3 Critiques

La méthode de comparaison par images-clés permet de constituer une signature compacte d'une séquence d'images : en effet l'information apportée par toutes les images est réduite à celle que contiennent ces images-clés. L'économie de ressources est significative, autant du point de vue du stockage et de la transmission d'une signature que du temps de calcul nécessaire à une comparaison entre deux signatures. Toutefois, il faut prendre en compte le temps nécessaire à la sélection des images-clés.

Nous avons vu, à travers ces quelques exemples, que l'extraction d'images-clés, qu'elle soit réalisée à partir d'un critère temporel, de l'analyse d'une trajectoire ou à partir de l'exploitation d'une matrice de proximités intra-séquence, risque de ne pas fournir des résultats satisfaisants dans le cas de séquences-personnes. En effet, il est difficile d'extraire des images-clés qui sont véritablement représen-

tatives de la personne observée.

Nous allons donc nous intéresser à une autre approche qui a toujours pour objectif l'économie de ressources tout en maintenant un pouvoir élevé de discrimination entre le cas où les séquences sont similaires et celui où elles sont différentes.

4.5 Signatures par descripteurs de séquences

Face aux limites de l'approche qui consiste à caractériser une séquence d'images à l'aide de signatures d'images-clés, il est préférable de considérer l'ensemble des images d'une séquence et de caractériser directement cet ensemble par une seule signature composée de *descripteurs* de séquences d'images.

Au lieu de limiter le nombre d'images à considérer, cette approche consiste à employer plusieurs descripteurs de séquences d'images, chacun d'eux caractérisant l'intégralité de la séquence des signatures de ces images.

4.5.1 Descripteurs de *group-of-frames*

Ferman et al. [FTMK00] appellent *group-of-frames* une séquence d'images présentant une certaine continuité, c'est-à-dire qu'elles ne contiennent pas de rupture brusque entre deux images consécutives. Nous pouvons donc aisément assimiler une séquence-personne à un group-of-frames. Ils proposent une signature de group-of-frames composée de différents descripteurs calculés à partir d'histogrammes couleur. Ces descripteurs sont calculés à partir de l'ensemble des histogrammes de toutes les images d'une séquence. Ils ont été intégrés dans le standard MPEG-7[2] en tant que descripteurs de couleur [MPE99].

Les descripteurs de Ferman caractérisant une séquence d'images S de n images, notées I_1 à I_n, sont constitués des n histogrammes couleur correspondants. Nous utilisons la notation basée sur les vecteurs d'attributs, à savoir que le contenu des histogrammes couleurs tridimensionnels est représenté sous la forme d'un vecteur : un seul index est nécessaire pour désigner la cellule d'un vecteur d'attributs.

[2]Le *Motion Picture Expert Group* établit les normes et les standards en matière de vidéo numérique

4.5.1.1 Histogramme Couleur Moyen (*Average Color Histogram*)

L'histogramme moyen [FTMK00] de la séquence d'images S est obtenu en accumulant les histogrammes $\mathbf{x}_H[I_1]$ à $\mathbf{x}_H[I_n]$ de toutes les images puis en divisant chaque cellule par le nombre n d'images de la séquence. La j^{eme} cellule de l'histogramme moyen $\mathbf{avg}_H[S]$ est donc définie par :

$$\mathbf{avg}_H[S](j) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_H[I_i](j) \tag{4.2}$$

Un défaut éventuel de ce descripteur, déjà évoqué par Bednar [BW84], est la sensibilité de l'opérateur statistique de moyenne face aux valeurs extrêmes ou aberrantes. En effet, puisque les données sont équipondérées, une valeur trop éloignée des autres peut biaiser le résultat qui risque de ne plus être représentatif de l'ensemble des données.

4.5.1.2 Histogramme Couleur Median (*Median Color Histogram*)

Une manière de pallier ce défaut consiste à remplacer l'opérateur statistique de moyenne par celui de médiane, comme cela est proposé par Arce [AGN86]. L'influence des valeurs aberrantes est alors réduite. La cellule j de l'histogramme médian [FTMK00] correspondant est définie par :

$$\mathbf{med}_H[S](j) = median\{\mathbf{x}_H[I_1](j), \mathbf{x}_H[I_2](j), \ldots, \mathbf{x}_H[I_n](j)\} \tag{4.3}$$

Pour obtenir la valeur de chaque cellule (j) de l'histogramme médian, on constitue la liste des valeurs contenues dans la cellule (j) de l'histogramme de chaque image de la séquence. La liste est ensuite triée par ordre croissant, et la valeur médiane de cette liste correspond à la valeur de la cellule (j) de l'histogramme médian. Ceci est répété autant de fois qu'il y a de cellules dans un histogramme.

Comparé au calcul de l'histogramme moyen, celui de l'histogramme médian est plus coûteux en temps de calcul car il nécessite d'effectuer le tri de chaque liste de valeurs correspondant aux cellules des histogrammes. Ferman utilise l'algorithme *quicksort* : l'ordre de grandeur du nombre de comparaisons nécessaire à l'ordonnancement des listes est de $v \times O(n\log(n))$ avec n le nombre d'images et v le nombre de cellules de l'histogramme.

4.5.1.3 Histogramme-Minimum

Ferman propose également l'histogramme minimum [FTMK00] qui est obtenu en retenant, la plus petite valeur des cellules analogues des histogrammes de toutes les images de la séquence. La cellule (j) de l'histogramme-mninimum est définie par :

$$\mathbf{min}_H[S](j) = min\{\mathbf{x}_H[I_1](j), \mathbf{x}_H[I_2](j), \ldots, \mathbf{x}_H[I_n](j)\} \tag{4.4}$$

Chaque cellule de l'histogramme-minimum représente le plus petit nombre de pixels d'une couleur donnée qui sont présents dans toutes les images. Ce descripteur caractérise la séquence d'images de manière différente de celle des histogrammes moyen et médian, en ce sens qu'il ne décrit pas la distribution des couleurs dans l'ensemble des images mais plutôt la distribution de chaque couleur "dans le pire des cas".

4.5.1.4 *Alpha-trimmed Histograms*

L'histogramme moyen et l'histogramme médian sont des cas particuliers d'une famille de descripteurs appelés *alpha-trimmed average histograms* [FTM02] et qui sont créés à l'aide de l'opérateur *trimmed mean* (moyenne élaguée[3]). L'histogramme moyen d'une séquence de n images est obtenu en triant les v listes des cellules des histogrammes et en calculant leur moyenne en ne prenant en compte que les valeurs centrales. Chaque cellule j de l'*alpha-trimmed average histogram* est définie par :

$$\alpha\mathbf{Trim}_H[S](j,\alpha) = \frac{1}{n - 2 \times [\alpha n]} \sum_{m=[\alpha n]+1}^{n-[\alpha n]} \hat{h_{(j)}}(m) \tag{4.5}$$

où $[\alpha n]$ représente le plus grand entier inférieur à αn et $\hat{h_j}$ la liste ordonnée des valeurs de la cellule (j) de tous les histogrammes.

Le paramètre α est compris entre 0 et 0.5 et détermine la quantité de données qui ne sont pas prises en compte. Notons que lorsque ce paramètre α vaut 0, on retrouve l'histogramme moyen et lorsqu'il vaut 0.5 on retrouve l'histogramme médian.

[3] Une moyenne élaguée d'un échantillon est la moyenne empirique de l'échantillon, privé d'un certain pourcentage de ses valeurs extrêmes.

4.5.1.5 Comparaison de signatures

En retenant plusieurs valeurs du paramètre α, on obtient différents *alpha-trimmed average histograms*. Ces descripteurs forment alors la signature d'une séquence d'images.

Les fonctions de comparaison employées par Ferman pour comparer les descripteurs deux-à-deux sont soit l'intersection d'histogrammes, soit les normes $L1$ et $L2$, soit la mesure du χ^2 [Sap90].

Enfin, la moyenne des mesures de proximités entre descripteurs homologues constitue la proximité entre les deux signatures de séquences.

Les descripteurs de Ferman sont calculés sur l'ensemble des histogrammes, et l'ordre temporel n'intervient pas : ces descripteurs sont indépendants du rang d'acquisition des images. En effet, chaque cellule j d'un descripteur est calculée indépendamment des autres cellules après un réordonnancement de la liste des cellules j des histogrammes de chaque image d'une séquence.

Nous allons maintenant décrire des descripteurs qui prennent en considération le rang d'acquisition des images au sein d'une séquence.

4.5.2 Histogramme Couleur Dominant (*Dominant Color Histogram*)

Lin et al. proposent l'*Histogramme Couleur Dominant* [LZ00] et le comparent aux descripteurs de Ferman dans [LNZS01]. Ils s'appuient sur le principe d'un descripteur, mais exploitent également l'information apportée par l'aspect temporel des données.

Leur descripteur est basé sur l'histogramme dominant [MZ98] de chacune des images qui ne prend en compte que les cellules dont les valeurs sont les plus importantes, parmi toutes les cellules de l'histogramme, selon un critère non détaillé dans [MZ98].

Ceci a pour effet, d'une part de rendre la représentation de l'histogramme plus compacte, d'autre part de réduire la dispersion introduite par les cellules dont les valeurs sont faibles. Lin et al. étendent ce principe de valeur dominante à la dimension temporelle [LZ00]. La création de l'histogramme couleur dominant d'une séquence d'images se fait de la manière suivante :

– D'abord, l'histogramme dominant de chaque image est calculé.

- Les cellules de cet histogramme dominant qui représentent un maximum local sont ensuite identifiées.
- Le voisinage 3×3 des cellules, centré sur chacun de ces maxima locaux désigne un *objet-couleur*.
- La somme des valeurs contenues dans les cellules d'un objet-couleur représente le nombre de pixels que comporte cet objet-couleur.
- Seuls les objets-couleur qui comportent le plus grand nombre de pixels (les 20 premiers) sont conservés et appelés *objets couleur dominants*.
- Ensuite, les objets-couleur dominants qui sont présents dans plusieurs images consécutives sont identifiés.
- Enfin, seuls les objets-couleur dominants présents dans le plus grand nombre d'histogrammes dominants sont retenus pour constituer le descripteur de la séquence et un poids leur est affecté. Ce poids est défini par le rapport entre d'une part le nombre d'histogrammes dominants consécutifs dans lesquels l'objet-couleur est présent et d'autre part le nombre total d'histogrammes de la séquence.

Lin et Al. proposent ainsi un descripteur qui met en évidence les caractéristiques ou objets-couleur qui se prolongent dans le temps.

4.5.3 Critiques

Une approche par descripteurs semble être mieux adaptée qu'une approche par images-clés, car elle permet de calculer une signature plus compacte qu'une signature composée de plusieurs signatures d'images-clés. De plus, cette approche permet d'éviter les problème soulevés par la sélection des images-clés.

Ferman propose des solutions simples et économiques en termes de coût de représentation d'une signature de séquence. Le prix de cette simplicité est la perte de l'information temporelle : il est ainsi impossible d'exploiter les variations observées au long de la séquence. Dans le cas des séquences-personnes, il serait souhaitable, non seulement de prendre en compte, mais aussi d'exploiter ces variations induites par le déplacement des personnes, notamment le mouvement des jambes. Lin et al. réalisent un progrès dans ce sens puisqu'ils exploitent l'information dominante dans le temps.

Nous pouvons regretter que ce soit cette information que l'on cherche à privilégier : en effet, lorsque l'on observe les différentes images d'une séquence-personne,

on s'aperçoit que l'information dominante (au sens de Lin) est apportée par la tête et le buste de cette personne puisqu'ils sont visibles dans toutes les images. L'information apportée par les bras et les jambes de la personne, en revanche, risque d'être moins importante lors de la création d'un descripteur dominant. Pourtant, cette information est également intéressante lorsqu'il s'agit de caractériser une personne qui se déplace : un pantalon permet de caractériser la tenue d'une personne au même titre qu'une veste.

4.6 Signatures "spatio-temporelles"

Nous présentons ici la dernière famille de signatures de séquences d'images étudiées. Contrairement aux signatures par images-clé et aux signatures par descripteurs présentées précédemment, les signatures spatio-temporelles tiennent compte de l'évolution dynamique des textures présentes dans les images.

4.6.1 Textures temporelles

Nelson [PN93] et Polana [Pol94] ont introduit la notion de texture temporelle [NP92] dans le cadre de l'analyse du mouvement non paramétrique. Par opposition aux mouvements d'objets, déformables ou non mais définis par des contours identifiables comme ceux d'un véhicule ou d'une personne se déplaçant dans la scène observée, on peut observer des mouvements locaux comme ceux provoqués par des surfaces texturées en mouvement. Fablet [Fab01] caractérise les textures temporelles présentes dans des séquences d'images et formées par le feu ou la fumée, l'eau de la mer, d'un torrent ou d'une rivière plus calme, ou encore le feuillage d'un arbre agité par le vent.

De même que l'on peut caractériser une texture d'après son contenu spatial, il est possible de caractériser le contenu spatio-temporel d'une texture en mouvement à l'aide d'outils statistiques qui sont basés sur les outils employés pour caractériser les textures présentes dans une image : les co-occurrences temporelles et les corrélogrammes temporels. Nous ne perdons pas de vue la distinction entre les vêtements des usagers des moyens de transports et une véritable texture temporelle, mais nous nous intéressons ici au fait que l'aspect dynamique est pleinement pris en compte dans la définition des descripteurs.

4.6.2 Corrélogrammes temporels

Rautiainen et Doermann [RD02] étendent la notion de corrélogrammes en calculant les corrélogrammes temporels couleur. Ce descripteur statistique tient compte du voisinage spatio-temporel de chaque pixel.

Les corrélogrammes temporels d'une séquence S de n images I_1 à I_n sont regroupés dans une structure $\beta_d(S)$ à deux entrées. Chaque cellule $\beta_d(S)[\mathbf{c}_i, \mathbf{c}_j]$ représente la probabilité que, durant la séquence d'images, un pixel p_1 de couleur $\mathbf{c}_i$ soit éloigné d'une distance d d'un pixel p_2 de couleur $\mathbf{c}_j$. Cette distance est mesurée dans le plan-image et également dans le sens de défilement des images.

$$\beta_d[\mathbf{c}_i, \mathbf{c}_j] = Pr(p_2 \in (I_1(\mathbf{c}_j) \dots I_n(\mathbf{c}_j)), |p_1 - p_2| = d \tag{4.6}$$

Les inconvénients des corrélogrammes temporels sont malheureusement les mêmes que ceux des corrélogrammes : ils sont coûteux en termes de ressources, qu'il s'agisse du temps de calcul nécessaire à leur constitution, de celui nécessaire à leur comparaison, ou encore de la quantité de mémoire nécessaire à leur stockage.

4.6.3 Co-occurrences temporelles

La démarche de Fablet [Fab01] est moins coûteuse que celle que nous venons de présenter puisqu'elle consiste à calculer les matrices de co-occurrences spatio-temporelles au lieu des corrélogrammes.

Nous notons $T_{r,g}[S]$ la matrice de co-occurrences chromatiques spatio-temporelles d'une séquence S, relative aux canaux r et g. La cellule $T_{rg}[S](x, y)$ contient le nombre de fois qu'un pixel dont le niveau de la composante r égal à x se trouve dans le voisinage spatio-temporel $3 \times 3 \times 3$ (voir figure 4.4) d'un pixel dont le niveau de la composante g est égal à y.

Lorsque la couleur des pixels est représentée dans le plan chromatique (r, g), la signature de la séquence S est composée de trois matrices de co-occurrences spatio-temporelles : $T_{r,r}[S]$ $T_{r,g}[S]$ et $T_{g,g}[S]$.

La comparaison entre les matrices de co-occurrences spatio-temporelles est alors identique à celle entre matrices de co-occurrences chromatiques de deux images (voir paragraphe 3.4). Elle est basée sur la moyenne des intersections entre matrices de co-occurrences spatio-temporelles analogues.

Le temps de calcul nécessaire à la constitution des matrices de co-occurrences spatio-temporelles à partir d'une séquence d'images est cependant supérieur au

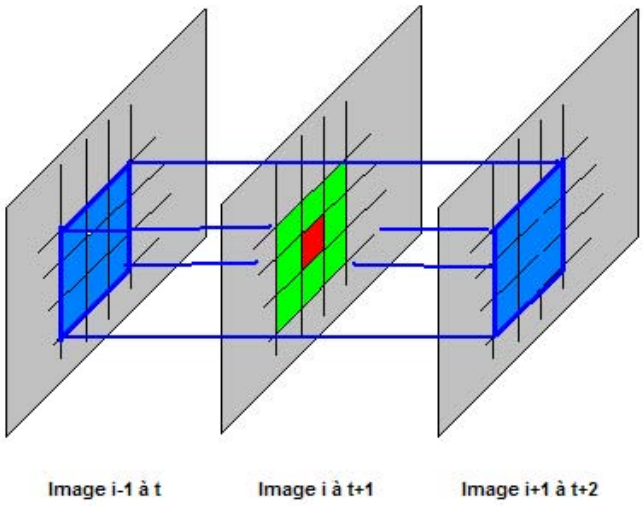


FIG. 4.4 – *Voisinage spatio-temporel* $3 \times 3 \times 3$ *d'un pixel P (en bleu).*

temps nécessaire à la constitution des matrices de co-occurrences chromatiques de toutes les images de cette même séquence. En effet le voisinage spatio-temporel de chaque pixel contient 27 pixels : on peut donc estimer que le temps nécessaire à la constitution d'une matrice de co-occurrences spatio-temporelles d'une séquence d'images est au moins trois fois plus élevé que celui nécessaire à la constitution des matrices de co-occurrences de toutes les images de cette séquence.

4.7 Conclusion

Dans ce chapitre, nous avons présenté les problèmes et les difficultés inhérentes à la comparaison de séquences d'images. L'aspect temporel apporte une dimension supplémentaire aux données. C'est pourquoi il n'est pas envisageable, pour le moment, de comparer toutes les images d'une séquence à toutes les images d'une autre séquence en utilisant les signatures d'images dont nous disposons. Il faut donc définir des signatures spécifiques aux séquences d'images.

Les signatures par images-clés permettent d'exploiter les variations observées dans une séquence d'images car elles sont composées de plusieurs signatures d'images. Cependant il est difficile de les déterminer de manière à ce qu'elles soient véritablement représentatives de la séquence.

Les signatures par descripteurs calculés pour l'ensemble des images d'une séquence sont plus robustes et plus compactes que les signatures d'images-clés. Elles prennent en compte les variations observées dans une séquence de signatures pour la résumer, mais ne permettent pas d'exploiter ces variations.

Enfin, les signatures dites *spatio-temporelles* caractérisent la déformation des textures apparaissant dans des images consécutives. Les matrices de co-occurrences chromatiques spatio-temporelles constituent une signature de séquence intéressante, quoique plus couteuses en termes de ressources que les signatures par descripteurs.

Nous allons maintenant proposer une signature par descripteurs qui tient compte des variations colorimétriques présentes au sein des images d'une séquence-personne mais aussi des interactions spatiales entre les pixels.

Chapitre 5

Comparaison de deux séquences de vecteurs d'attributs

5.1 Introduction

Dans la seconde partie de cette thèse, nous avons dressé un état de l'art non exhaustif de la comparaison de deux séquences d'images. Nous avons retenu du chapitre 3 que la signature d'une image couleur composée des matrices de co-occurrences chromatiques permet de décrire les textures couleur présentes dans cette image. Cette signature est ainsi plus pertinente que celle qui est composée de l'histogramme couleur d'une image.

Par ailleurs, dans le chapitre 4, nous avons présenté une signature de séquences d'images par descripteurs qui permet de résumer, à l'aide d'indicateurs statistiques, l'information contenue dans l'ensemble des signatures des images de cette séquence.

Dans ce chapitre, nous proposons de constituer une signature de séquence d'images en employant les descripteurs du chapitre 4 pour résumer l'information contenue dans une séquence de matrices de co-occurrences chromatiques.

Dans le premier paragraphe, nous définissons le concept de séquence de matrices de co-occurrences. Nous illustrons son contenu à l'aide d'un exemple et mettons en évidence quelques-unes de ses propriétés. Ces données constituent le point de départ de la construction des signatures de séquences-personnes.

Dans le deuxième paragraphe, nous proposons d'adopter l'approche de Ferman pour caractériser une séquence-personne, c'est-à dire de créeer une signature par

descripteurs estimés à partir de la séquence de matrices de co-occurrences. Nous décrivons la procédure que nous avons employée pour comparer deux séquences-personnes et analysons la pertinence des résultats obtenus sur la base de séquences que nous avons constituée.

Enfin, dans le troisième paragraphe, nous proposons de caractériser chaque séquence-personne par des séquences d'indices de textures, calculés à partir de séquences de matrices de co-occurrences. Nous comparons les résultats obtenus par ces deux approches. Par ailleurs, nous évaluons avec soin les ressources nécessaires à la mise en oeuvre de ces deux approches afin d'estimer leur faisabilité.

5.2 Les séquences-personnes considérées

5.2.1 Exemples de séquences-personnes

Dans ce chapitre et les suivants, les méthodes de comparaison de séquences-personnes sont toujours illustrées à partir des mêmes séquences issues d' exemples tirés de la base de séquences-personnes. Dans un but didactique, nous considérons deux personnes dont les vêtements sont très différents, comme le montrent la figure 5.1 représentant une image-personne extraite de la séquence représentant le déplacement de la personne $\mathcal{F}$ et la figure 5.2 associée à la personne $\mathcal{B}$. Nous considérons également la séquence-personne représentant le déplacement de la personne $\mathcal{D}$ (cf. figure 5.3) dont les vêtements ressemblent fortement à ceux de la personne $\mathcal{F}$. Nous considérons le cas où la séquence-personne requête $\mathcal{F}^{(1)}$ est celle qui représente le déplacement de la personne $\mathcal{F}$. Nous examinons les trois séquences candidates suivantes :

- la séquence candidate $\mathcal{F}^{(2)}$ représente un autre déplacement de la personne $\mathcal{F}$,
- la séquence candidate notée $\mathcal{B}^{(1)}$ représente le déplacement de la personne $\mathcal{B}$,
- la séquence candidate notée $\mathcal{D}^{(1)}$ représente le déplacement de la personne $\mathcal{D}$.

5.2.2 Séquences de signatures d'images par matrices de co-occurrences chromatiques

Nous avons vu dans l'état de l'art qu'une signature composée des trois matrices de co-occurrences chromatiques permet de caractériser les textures présentes dans

FIG. 5.1 – *Image-personne extraite de la séquence-personne représentant le déplacement de la personne $\mathcal{F}$.*

FIG. 5.2 – *Image-personne extraite de la séquence-personne représentant le déplacement de la personne $\mathcal{B}$.*

FIG. 5.3 – *Image-personne extraite de la séquence-personne représentant le déplacement de la personne $\mathcal{D}$.*

une image dont les couleurs sont codées dans un plan chromatique. La signature $\phi_{C_{(r,g)}}[I]$ d'une image I représentée dans le plan (r, g) est formée du triplet de matrices de co-occurrences normalisées $\{C_{rr}[I], C_{rg}[I], C_{gg}[I]\}$.

Lorsque les valeurs prises par les pixels de l'image sont réparties sur 16 niveaux par canal, chaque matrice de co-occurrences est de taille $v_M = 16 \times 16$, soit 256 cellules.

Pour caractériser une séquence de n images, nous disposons d'une séquence de n signatures d'images, chacune formée de trois vecteurs d'attributs pour ca-

ractériser la séquence S :

$$\Psi[S] = \{\phi_{C_{(r,g)}}[I_i]\} = \{\mathbf{x}_{C_{rr}}[I_i], \mathbf{x}_{C_{rg}}[I_i], \mathbf{x}_{C_{gg}}[I_i]\} i = 1, n$$

5.2.3 Examen d'une séquence de matrices de co-occurrences

Nous illustrons ici l'aspect temporel des données à l'aide de l'exemple présenté dans le premier paragraphe. Pour cela, nous fournissons la représentation graphique de la séquence de n matrices de co-occurrences $\mathbf{x}_{C_{rg}}[I_i]$ caractérisant la séquence $\mathcal{F}^{(1)}$, chaque matrice étant de taille $v_C = 16 \times 16$, soit 256 cellules.

La figure 5.4 comporte 256 courbes : elle représente l'évolution des v_C cellules au fil de l'acquisition des n images de la séquence. L'axe des abscisses représente le numéro de chaque image selon son ordre d'acquisition, et chaque courbe j est obtenue en parcourant la succession dans le temps des n valeurs de la j^{eme} cellule du vecteur d'attributs. La j^{eme} courbe correspond à la suite :

$$\{\mathbf{x}_{C_{rg}}[I_i](j)\}, i = 1 \ldots n$$

Ces valeurs sont comprises entre 0 et 1 mais dépassent rarement la valeur 0.1.

En examinant la superposition de ces courbes, nous remarquons que certaines atteignent une valeur significative, et que les variations d'une même cellule peuvent être importantes au fil de l'acquisition des images.

5.2.4 Objectif : résumer une séquence de signatures d'images

A partir de chaque séquence-personne, nous proposons de calculer trois séquences de matrices de co-occurrences puisque la signature d'une image dont les couleurs sont représentées dans un plan chromatique est formée de trois matrices de co-occurrences.

Une signature de séquence d'images composée de descripteurs permet de résumer chaque séquence de vecteurs d'attributs à l'aide d'un nombre réduit de vecteurs d'attributs.

L'information apportée par l'ensemble des signatures des images d'une séquence est ainsi résumée en réduisant le nombre de vecteurs d'attributs employés et non en réduisant leur taille.

Dans le paragraphe suivant, nous allons présenter la méthode que nous avons employée pour constituer une signature de séquence d'images par descripteurs à

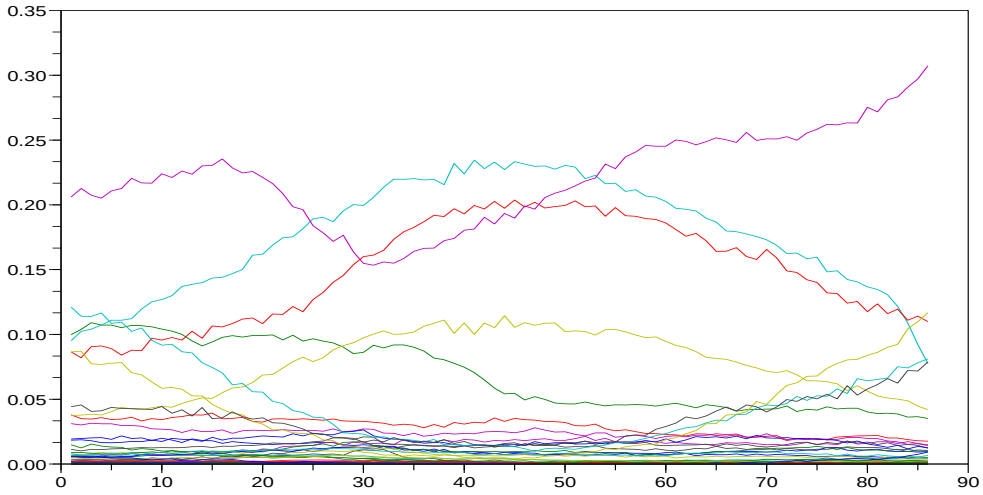


FIG. 5.4 – *Représentation des matrices de co-occurrences r-g des images de la séquence-personne $\mathcal{F}^{(1)}$. En abscisses, les images dans l'ordre d'acquisition. Chaque courbe représente la succession des valeurs prises par une composante du vecteur d'attribut pour les 86 images de la séquence.*

partir des matrices de co-occurrences normalisées.

5.3 Signatures par descripteurs de séquences d'images

Dans ce paragraphe, nous définissons la signature de séquence-personne basée sur les descripteurs de Ferman [FTMK00] comme le min et la médiane. Il s'agit avant tout de préciser le point de départ de nos travaux, qui commence par la vérification de l'hypothèse de Ferman selon laquelle une signature de séquence d'images par descripteurs permet de distinguer deux séquences-personnes.

Pour constituer cette signature par descripteurs, nous adoptons la même approche en considérant la séquence de signatures caractérisant les images de la séquence. La seule différence est que nous ne nous basons pas sur un seul vecteur d'attributs pour caractériser une image (Ferman utilise l'histogramme couleur) mais sur trois matrices de co-occurrences chromatiques, comme nous l'avons précisé dans l'état de l'art.

Nous nous plaçons dans l'espace couleur (r, g) mais la démarche est identique pour les autres espaces employés (H, S) et (l_1, l_2).

5.3.1 Principe

Nous considérons trois séquences "indépendantes" de vecteurs d'attributs correspondant aux trois séquences de matrices de co-occurrences normalisées :
$\{\mathbf{x}_{C_{rr}}[I_i]\}, \{\mathbf{x}_{C_{rg}}[I_i]\}$ et $\{\mathbf{x}_{C_{gg}}[I_i]\}, i = 1, \ldots, n$.

5.3.1.1 Calcul des descripteurs statistiques

Pour chacune de ces trois séquences de vecteurs d'attributs, nous calculons les descripteurs statistiques suivants : le *min*, la *médiane* et le *max*. Nous avons remarqué que l'utilisation des autres descripteurs (les descripteurs alpha-trimmed) n'apportent aucune information supplémentaire utile pour discriminer les séquences-personnes considérées.

Pour une séquence $\{\mathbf{x}_C[I_i]\}$ de n matrices de co-occurrences, chacune de taille v_C on calcule les trois indicateurs suivants :

$$\mathbf{min}_C(j) = min\{\mathbf{x}_C[I_1](j), \mathbf{x}_C[I_2](j), \ldots, \mathbf{x}_C[I_n](j)\},\ j = 1 \ldots v_C \tag{5.1}$$

$$\mathbf{med}_C(j) = median\{\mathbf{x}_C[I_1](j), \mathbf{x}_C[I_2](j), \ldots, \mathbf{x}_C[I_n](j)\},\ j = 1 \ldots v_C \tag{5.2}$$

$$\mathbf{max}_C(j) = max\{\mathbf{x}_C[I_1](j), \mathbf{x}_C[I_2](j), \ldots, \mathbf{x}_C[I_n](j)\},\ j = 1 \ldots v_C \tag{5.3}$$

Chacun de ces trois descripteurs est calculé pour chacune des trois séquences de vecteurs d'attributs $\{\mathbf{x}_{C_{rr}}[I_i]\}, \{\mathbf{x}_{C_{rg}}[I_i]\}, \{\mathbf{x}_{C_{gg}}[I_i]\}$ avec $i = 1, \ldots, n$.

Nous proposons donc d'employer la signature de séquence-personne suivante, obtenue par les descripteurs calculés sur l'ensemble des matrices de co-occurrences chromatiques. Pour une séquence-personne S dont les couleurs sont représentées dans le plan chromatique (r, g), la signature $\Phi_{C_{(r,g)}}[S]$ est alors exprimée par :

$$\Psi_{C_{(r,g)}}[S] = \begin{Bmatrix} \mathbf{min}_{C_{rr}}[S] & \mathbf{med}_{C_{rr}}[S] & \mathbf{max}_{C_{rr}}[S] \\ \mathbf{min}_{C_{rg}}[S] & \mathbf{med}_{C_{rg}}[S] & \mathbf{max}_{C_{rg}}[S] \\ \mathbf{min}_{C_{gg}}[S] & \mathbf{med}_{C_{gg}}[S] & \mathbf{max}_{C_{gg}}[S] \end{Bmatrix} \tag{5.4}$$

5.3.1.2 Distance entre signatures par descripteurs

La distance entre deux séquences-personnes S_{Req} et S_{Can} s'effectue en calculant la moyenne des neuf mesures de proximité entre descripteurs analogues

de $\Psi_{C_{(r,g)}}[S_{Req}]$ et $\Psi_{C_{(r,g)}}[S_{Can}]$. Pour ce faire, la distance entre deux descripteurs qui est employée est la norme $L1$. Par exemple pour comparer le descripteur $\mathbf{min}_{C_{rg}}[S_{Req}]$ au descripteur $\mathbf{min}_{C_{rg}}[S_{Can}]$, la distance s'exprime par $\sum_{j=1}^{v_C} \mid \mathbf{min}_{C_{rg}}[S_{Req}](j) - \mathbf{min}_{C_{rg}}[S_{Can}](j) \mid$.

5.3.2 Résultats expérimentaux

5.3.2.1 Exemples

Le tableau 5.1 fournit les mesures de distance obtenues en comparant la séquence requête $\mathcal{F}^{(1)}$ aux trois séquences candidates $\mathcal{F}^{(2)}$, $\mathcal{D}^{(1)}$ et $\mathcal{B}^{(1)}$ lorsque la couleur est codée dans chacun des plans chromatiques :

candidate	(r,g)	(H,S)	(l_1,l_2)
$\mathcal{F}^{(2)}$	0.111	0.038	0.057
$\mathcal{B}^{(1)}$	0.310	0.101	0.160
$\mathcal{D}^{(1)}$	0.106	0.024	0.077

TAB. 5.1 – *Distances mesurées par comparaison de descripteurs de séquences de vecteurs d'attributs entre la séquence-requête $\mathcal{F}^{(1)}$ et les trois séquences candidates $\mathcal{F}^{(2)}$, $\mathcal{D}^{(1)}$ et $\mathcal{B}^{(1)}$, en fonction du plan chromatique dans lequel sont représentées les séquences d'images.*

Ce tableau montre que, pour chacun des trois plans chromatiques, la distance qui sépare les signatures de deux séquence-personnes similaires (ligne 1) est bien inférieure à celle qui sépare les signatures de deux séquences-personnes très différentes (ligne 2).

Par contre, la troisième ligne du tableau montre que lorsque les vêtements de la personne représentée dans la séquence-candidate ressemblent à ceux de la personne représentée dans la séquence-requête, les distances mesurées sont beaucoup moins importantes : pour les plans couleur (r,g) et (H,S), les mesures sont même inférieures à celles qui sont obtenues en comparant les signatures de deux séquences représentant deux personnes similaires.

Enfin, dans l'espace (l_1,l_2), la distance entre les signatures des séquences $\mathcal{F}^{(1)}$ et $\mathcal{D}^{(1)}$ est à peine plus importante que celle entre les signatures des séquences

$\mathcal{F}^{(1)}$ et $\mathcal{F}^{(2)}$.

Toutefois, les personnes $\mathcal{F}$ et $\mathcal{D}$ constituent un des seuls cas de la base de données pour lequel les signatures de deux séquences différentes sont plus proches que celles de deux séquences similaires.

5.3.2.2 Score de pertinence

Nous avons appliqué la méthode que nous venons de présenter à la base de données de séquences-personnes décrite au chapitre 2. Le détail de la procédure d'évaluation de la pertience d'une méthode de comparaison appliquée à toutes les séquences de la base est fourni en annexe 2.

Nous pouvons simplement retenir que la pertinence d'une méthode de comparaison est évaluée à l'aide d'une valeur scalaire positive ou nulle. Si ce score de pertinence vaut 0, cela signifie que, pour chaque séquence-requête représentant une personne de la base, les séquences candidates les plus proches correspondent bien à cette personne. Enfin, la valeur maximale du score de pertinence, correspondant au "pire des cas", est fonction du nombre de séquences, et vaut 14 dans le cas de notre base de données. Il s'agit du cas où, pour chaque séquence-requête représentant une personne de la base, les séquences candidates qui correspondent bien à cette personne sont les plus éloignées.

Le tableau 5.2 fournit, pour chacun des plans chromatiques, les scores de pertinence atteints par notre méthode de comparaison.

(r, g)	(H, S)	(l_1, l_2)
0.053	0.139	0.075

TAB. 5.2 – *Scores de pertinences de la comparaison par descripteurs de séquences de vecteurs d'attributs appliquée à la base de données de séquences-personnes, en fonction du plan chromatique dans lequel sont représentées les séquences d'images.*

En dehors du plan chromatique (H, S) dans lequel les résultats sont moins pertinents, nous constatons que les scores atteints par cette méthode sont très satisfaisants : en effet, dans les plans chromatiques (r, g) et (l_1, l_2), seuls les personnes $\mathcal{F}^{(1)}$ et $\mathcal{D}^{(1)}$, dont les vêtements se ressemblent fortement, ne sont pas toujours bien discriminées.

Nous avons la confirmation qu'une signature par descripteurs peut être utilisée à partir des matrices de co-occurrences chromatiques normalisées pour caractériser une séquence-personne de manière satisfaisante dans le cadre de notre base de séquences.

5.3.3 Analyse des ressources nécessaires

Nous décrivons maintenant les ressources nécessaires à l'application de cette méthode.

La quantité de mémoire nécessaire à la représentation des signatures en vue du stockage ou de leur transmission par le réseau, ainsi que le temps nécessaire à la comparaison de deux signatures, sont des ressources qu'il faut prendre en compte.

Cependant nous avons vu que le temps nécessaire à la création des signatures doit également être considéré dans le cas de figure de la localisation automatique de personnes.

5.3.3.1 Ressources pour le calcul d'une séquence des signatures d'images

L'évaluation des ressources de temps de calcul et de mémoire nécessaires pour constituer la séquence de signatures d'images est à considérer séparément. En effet, il n'est pas question d'attendre que la dernière image de la séquence soit acquise pour entamer le calcul de la signature de la première image.

Nous considérons donc que, tant que faire se peut, les signatures d'images sont calculées l'une après l'autre, au fur-et-à-mesure de leur acquisition si la puissance de calcul le permet, du moins de manière progressive durant l'acquisition des images.

La quantité de cellules mémoire nécessaire au stockage de trois séquences de n vecteurs d'attributs s'élève à $3 \times n \times v_C$ cellules lorsque ces vecteurs d'attributs sont de taille v_C chacun. Toutefois cette quantité ne doit pas être prise en compte pour l'évaluation de la taille de la signature par descripteurs Ψ : une fois que les descripteurs sont constitués, la mémoire nécessaire au stockage de la séquence de signatures d'images peut être libérée.

Pour la même raison que celle que nous avons évoquée, le temps de calcul nécessaire à la constitution de la séquence de signatures d'images est considéré séparément. Le calcul d'une matrice de co-occurrences d'une image comportant

P pixels-personnes nécessite $P \times 16$ opérations élémentaires, 8 étant le nombre de couples de pixels voisins considérés et chaque couple faisant appel à deux opérations d'adressage. La construction des trois matrices $C_{rr}[I], C_{rg}[I], C_{gg}[I]$ met en oeuvre $3 \times 16 \times P$ pixels-personnes, et ce pour chacune des n images.

Les images-personnes de notre base de données sont représentées par une faible quantité de pixels-personnes, en comparaison au nombre total de pixels d'une image. Nous avons mesuré que le temps moyen nécessaire à la construction des trois matrices de co-occurrences chromatiques formant la signature d'une image, à l'aide d'un logiciel développé en C++, nécessite moins de 20 ms sur une machine équipée d'un processeur cadencé à 1,4 GHz. Ce temps de calcul est donc proche de la cadence d'acquisition des images-personnes, si l'on exclut le temps de calcul nécessaire aux pré-traitements.

Enfin, le temps de calcul nécessaire à la normalisation des trois matrices de co-occurrences dépend, lui, du nombre de cellules non nulles des matrices de co-occurrences. Cependant celui-ci est peu important puisque la normalisation de trois séquences de matrices de co-occurrences ne dépasse jamais 10 ms. Nous précisons que cette opération a été réalisée à l'aide du logiciel *Scilab*.

5.3.3.2 Ressources mémoire pour le stockage des signatures par descripteurs

Sachant que nous considérons trois séquences de matrices de co-occurrences, chacune étant résumée par trois descripteurs, la taille totale W occupée par cette signature et exprimée en nombre de valeurs scalaires est de :

$$W(\Psi_{C_{(r,g)}}[S_{Req}]) = W(\Psi_{C_{(r,g)}}[S_{Can}])) = 3 \times 3 \times v_C = 2304.$$

Ceci représente, cette fois, la taille de la signature qui sera transmise via le réseau ou stockée sur un support physique. La signature $\Psi_{C_{(r,g)}}$ que nous proposons a l'avantage d'être assez compacte, comparée à l'information qu'elle résume : en effet, pour chacune des trois séquences de n matrices de taille $v_C = 256$ il suffit de trois descripteurs de taille v_C (min, med et max).

De plus, la taille de cette signature est constante quel que soit le nombre d'images de la séquence-personne qu'elle caractérise. Si l'on code une valeur scalaire sur 4 octets, la taille mémoire d'une telle signature s'élève à exactement 9 Ko.

5.3.3.3 Temps de calcul nécessaire à la création des signatures

Ce coût se répercute d'une part sur la création de la signature requête, d'autre part sur la création de la signature candidate. Dans le cas de la localisation automatique, ce surcroit de calcul occasionné par la création de la signature candidate doit être pris en compte dans le temps total que nécessite une comparaison.

Le calcul des descripteurs min, max et médian, effectué à partir d'une séquence de matrices de co-occurrences normalisées, fait appel à un algorithme de tri, c'est pourquoi nous avons distingué , d'une part la création des signatures de chaque image, d'autre part la création des descripteurs qui nécessite de disposer de toutes les signatures d'images. Ce tri est nécessaire pour déterminer la valeur médiane de chaque composante, qui est préférable à la moyenne comme nous l'avons évoqué au chapitre 4 dans la présentation des descripteurs de Ferman.

Ce tri est effectué pour chaque composante d'un vecteur d'attributs. L'algorithme employé est celui du *quicksort*, le nombre d'opérations nécessaires au calcul de la signature candidate $\Psi_{C_{(r,g)}}[S_{Can}]$ est alors de :

$$T(\Psi^F_{C_{(r,g)}}[S_{Can}]) = 3 \times v_C \times n_{Can} \times log(n_{Can}).$$

Nous avons mesuré le temps nécessaire à la création d'une signature par descripteurs à partir d'une séquence de signatures d'images, toujours à l'aide du logiciel *Scilab*. Les $3 \times v_C$ tris de valeurs consomment la majeure partie des ressources en temps de calcul, toutefois ce temps est de l'ordre de 60 ms pour une séquence de 100 images (environ 20 ms pour chaque séquence de vecteurs d'attributs), ce qui reste très largement inférieur à la durée d'acquisition d'une séquence-personne.

5.3.3.4 Temps de calcul pour la comparaison des signatures

L'opération de comparaison par la norme $L1$ entre deux descripteurs nécessite v_C opérations élémentaires[1]. La comparaison se faisant entre descripteurs homologues, le nombre T d'opérations élémentaires est de l'ordre de :

$$T(Prox(\Psi^F_{C_{(r,g)}}[S_{Req}], \Psi^F_{C_{(r,g)}}[S_{Can}])) = 3 \times 3 \times v_C = 2304.$$

Ceci correspond à un simple parcours des v_C cellules pour chacun des neuf descripteurs.

[1] Une différence suivie d'une somme

Plus concrètement, la mesure expérimentale du temps nécessaire à la comparaison de signatures a toujours fourni un résultat inférieur à 1 ms lorsque celle-ci est réalisée à l'aide du logiciel *Scilab*.

5.3.4 Généralisation

Ferman utilise la séquence formée par les histogrammes couleur des images, tandis que nous utilisons les trois séquences formées des matrices de co-occurrences chromatiques. Cependant, la démarche est la même puisqu'il s'agit dans les deux cas de comparer des séquences de vecteurs d'attributs en mesurant la proximité entre leurs signatures composées de descripteurs.

Nous avons montré expérimentalement que cette approche s'avère particulièrement efficace pour la discrimination des séquences-personnes de notre base de données expérimentale lorsque nous l'adoptons avec des matrices de co-occurrences chromatiques.

Par la suite, nous continuerons donc à adopter cette approche pour comparer deux séquences de vecteurs d'attributs, seule la nature des attributs sera différente.

Nous avons proposé une méthode permettant de créer une signature qui résume une séquence de signatures d'images. L'avantage de cette signature est que le nombre de descripteurs est fixe, quel que soit le nombre d'images de la séquence.

Cependant la taille de cette signature de séquence-personne est proportionnelle à la taille de la signature de chaque image de la séquence, à savoir de la taille des matrices de co-occurrences.

Nous allons proposer dans le paragraphe suivant une signature de séquences-personnes caractérisant les couleurs et les textures observées dans les images, et dont la taille ne dépend pas de celle des matrices de co-occurrences.

5.4 Descripteurs de séquences de vecteurs d'indices de texture

Nous venons de présenter une méthode de comparaison de séquences-personnes. Les données sur lesquelles s'applique cette méthode sont les séquences de signa-

tures des images des séquences-personnes, chaque signature étant composée de un ou plusieurs vecteurs d'attributs.

Les descripteurs permettent de réduire la taille de la signature qui passe de $n \times v_C$ pour la séquence de n matrices de co-occurrences de taille v_C, à $3 \times v_C$, lorsque les descripteurs sont le min, la médiane et le max. Cependant la détermination de ces descripteurs nécessite un temps de calcul également proportionnel à la taille des vecteurs d'attributs.

Nous avons donc mis en évidence que le paramètre v intervient dans l'expression du calcul des ressources nécessaires : il s'agit du nombre de composantes du vecteur d'attributs utilisé pour constituer la signature d'une image.

5.4.1 Principe

Nous proposons de calculer des indices de texture à partir de chaque matrice de co-occurrences pour chaque image de la séquence considérée.

5.4.1.1 Indices de Texture

Les indices de texture sont des valeurs scalaires qui caractérisent les textures présentes dans une image en quatifiant certaines notions du langage "courant". Ils sont présentés par Cocquerez et Al. [CP95] de la manière suivante :

> *Les matrices de co-occurrences contiennent une masse d'information trop importante et difficilement manipulable dans son intégralité (256 valeurs lorsque les niveaux sont quantifiés sur 16 valeurs, et de nombreuses valeurs nulles en général). C'est pourquoi quatorze indices prenant en compte l'ensemble des valeurs d'une matrice de co-occurrences ont été définis par Haralick[Har79]. Ils correspondent à des caractères descriptifs des textures comme le contraste ou l'homogénéité.*

Bien que ces indices soient corrélés, Conners et Harlow [CH80] ont prouvé que cinq indices (homogénéité, contraste, entropie, corrélation et homogénéité locale) permettent une meilleure discrimination des textures que d'autres attributs tels que les longueurs de plage (voir chapitre 3, paragraphe 3.3).

Les matrices de co-occurrences caractérisent les interactions spatiales entre les pixels d'une image. Il est possible d'analyser ces matrices et d'extraire des attributs qui reflètent les propriétés de l'organisation spatiale d'une texture [CH80]. Chacun de ces attributs est une grandeur scalaire comprise entre 0 et 1.

Ainsi, une image homogène a tendance à présenter moins de transitions qu'une image fortement texturée. On peut donc calculer l'homogénéité de la texture qui s'exprime sous la forme d'un scalaire : soit M une matrice de co-occurrences non normalisée de taille $q \times q$, et s la somme des cellules de la matrice :

Le premier indice qui sera employé est celui d'**homogénéité** :

$$hom(M) = \frac{1}{s^2} \sum_{a=0}^{q-1} \sum_{b=0}^{q-1} M(a,b)^2$$

L'homogénéité d'une texture est d'autant plus élevée qu'on retrouve souvent dans l'image les mêmes couples de pixels voisins caractérisés par des couples de niveaux spécifiques.

Le second indice est celui de **contraste** :

$$con(M) = \frac{1}{s(q-1)^2} \sum_{k=0}^{q-1} k^2 \sum_{|a-b|=k} M(a,b)$$

Chaque terme est pondéré par la distance à la diagonale : le contraste de la texture est élevé lorsque les termes éloignés de la diagonale de la matrice de co-occurrences sont élevés.

Le troisième indice est l'**entropie** :

$$ent(M) = 1 - \frac{1}{s \times ln(s)} \sum_{a=0}^{q-1} \sum_{b=0}^{q-1} M(a,b) \times ln(M(a,b)),\ si M(a,b) > 0$$

Cette valeur est faible si on retrouve souvent le même couple de pixels dans l'image (contrairement à l'homogénéité). L'entropie caractérise le taux de désordre d'une texture.

Le quatrième indice que nous emploierons est celui d'uniformité :

$$uni(M) = \frac{1}{s^2} \sum_{a=0}^{q-1} M(a,a)^2$$

5.4.1.2 Caractérisation d'images par vecteurs d'indices statistiques

A toute matrice de co-occurrences non normalisée $M[I]$, il est possible de faire correspondre un vecteur d'indices $\mathbf{x}_L[I]$ composé des quatre indices calculés à partir de cette matrice : homogénéité, contraste, entropie, uniformité. Si l'on considère qu'une image I est caractérisée par trois matrices de co-occurrences chromatiques, la signature $\phi_{L_{(r,g)}}[I]$ de cette image est composée de trois vecteurs de quatre attributs chacun :

$$\phi_{L_{(r,g)}}[I] = (\mathbf{x}_{L_{rr}}[I], \mathbf{x}_{L_{rg}}[I], \mathbf{x}_{L_{gg}}[I])$$

avec :

$$\begin{aligned}
\mathbf{x}_{L_{rr}}[I] &= (hom(M_{rr}[I]), con(M_{rr}[I]), ent(M_{rr}[I]), uni(M_{rr}[I])), \\
\mathbf{x}_{L_{rg}}[I] &= (hom(M_{rg}[I]), con(M_{rg}[I]), ent(M_{rg}[I]), uni(M_{rg}[I])), \\
\mathbf{x}_{L_{gg}}[I] &= (hom(M_{gg}[I]), con(M_{gg}[I]), ent(M_{gg}[I]), uni(M_{gg}[I]))
\end{aligned} \tag{5.5}$$

5.4.1.3 Séquences de vecteurs d'indices de texture

Nous pouvons donc appliquer la démarche appliquée à la comparaison de descripteurs de deux séquences de vecteurs d'attributs, cette fois en utilisant les indices de texture calculés à partir des matrices de co-occurrences chromatiques non normalisées.

Ceci représente une étape supplémentaire et préalable au calcul des descripteurs que nous allons appliquer maintenant. Notons que la normalisation des matrices de co-occurrences ne doit plus être effectuée car les indices sont normalisés. On peut estimer que l'ordre de complexité du calcul de chaque indice est proportionnel à v_C.

5.4.1.4 Descripteurs de séquences de vecteurs d'indices de textures

Pour une séquence-personne S de n images, nous considérons les trois séquences de vecteurs d'indices de texture :

$$\begin{aligned}
&\{\mathbf{x}_{L_{rr}}[I_i]\},\ i = 1 \ldots n \\
&\{\mathbf{x}_{L_{rg}}[I_i]\},\ i = 1 \ldots n \\
&\{\mathbf{x}_{L_{gg}}[I_i]\},\ i = 1 \ldots n
\end{aligned} \tag{5.6}$$

Pour chacune des trois séquences de vecteurs d'indices de texture, on calcule, de la même manière que pour les matrices de co-occurrences, les trois descripteurs minimum $\mathbf{min}_L$, médian $\mathbf{med}_L$ et maximum $\mathbf{max}_L$.

candidate	(r, g)	(H, S)	(l_1, l_2)
$\mathcal{F}^{(2)}$	0.088	0.159	0.035
$\mathcal{B}^{(1)}$	0.373	0.189	0.538
$\mathcal{D}^{(1)}$	0.149	0.244	0.095

TAB. 5.3 – *Distances mesurées par comparaison de descripteurs de séquences de vecteurs d'indices de texture entre la séquence-requête $\mathcal{F}^{(1)}$ et les trois séquences candidates $\mathcal{F}^{(2)}$, $\mathcal{D}^{(1)}$ et $\mathcal{B}^{(1)}$, en fonction du plan chromatique dans lequel sont représentées les séquences d'images.*

Par exemple, le descripteur median $\mathbf{med}_{L_{rr}}$ est un vecteur à quatre composantes, qui contient les valeurs médianes des quatre indices (homogénéité, contraste, entropie, uniformité) qui ont été calculés à partir de la séquence de matrices de co-occurrences $M_{rr}[i], i = 1 \dots n$.

La signature d'une séquence-personne S est alors composée des 9 vecteurs d'attributs suivants :

$$\Psi_{L_{(r,g)}}[S] = \left\{ \begin{array}{ccc} \mathbf{min}_{L_{rr}}[S] & \mathbf{med}_{L_{rr}}[S] & \mathbf{max}_{L_{rr}}[S] \\ \mathbf{min}_{L_{rg}}[S] & \mathbf{med}_{L_{rg}}[S] & \mathbf{max}_{L_{rg}}[S] \\ \mathbf{min}_{L_{gg}}[S] & \mathbf{med}_{L_{gg}}[S] & \mathbf{max}_{L_{gg}}[S] \end{array} \right\} \tag{5.7}$$

Le nombre de ces vecteurs d'attributs est le même que le nombre de vecteurs d'attributs qui composent la signature $\Psi_{C_{(r,g)}}[S]$. Par contre la taille de ces vecteurs d'attributs est différente et vaut ici $v_L = 4$.

Enfin, pour comparer les signatures requête et candidate, nous calculons la moyenne des proximités entre les descripteurs de vecteurs d'attributs homologues. Ici encore, nous avons employé pour cela la norme L_1.

5.4.2 Exemple et pertinence des résultats

5.4.2.1 Exemples

Le tableau 5.3 fournit les mesures de distances obtenues en comparant la séquence requête $\mathcal{F}^{(1)}$ aux trois séquences candidates $\mathcal{F}^{(2)}$, $\mathcal{D}^{(1)}$ et $\mathcal{B}^{(1)}$ dans chacun des plans chromatiques :

Bien que la taille des descripteurs de séquences de vecteurs d'indices de textures est beaucoup plus petite que celle des descripteurs de séquences de matrices

de co-occurrences normalisées, la distance mesurée entre la séquence $\mathcal{F}^{(1)}$ et la séquence $\mathcal{F}^{(2)}$ est toujours inférieure à la distance mesurée entre la signature de la séquence $\mathcal{F}^{(1)}$ et celle de la séquence $\mathcal{D}^{(1)}$.

Nous constatons que, grâce à cette signature, les deux séquences-personnes $\mathcal{F}^{(1)}$ et $\mathcal{D}^{(1)}$ sont bien discriminées, pour les trois plans chromatiques, alors qu'ils ne l'étaient pas lors de leur comparaison à l'aide de la signature formée à partir des matrices de co-occurrences normalisées.

5.4.2.2 Scores de pertinence

Le tableau 5.4.2.2 fournit, pour chacun des plans chromatiques, les scores de pertinence atteints par la méthode de comparaison par descripteurs de séquences de vecteurs d'indices de textures, appliquée sur la base complète de séquences-personnes.

(r, g)	(H, S)	(l_1, l_2)
0.043	0.451	0

TAB. 5.4 – *Scores de pertinence de la comparaison par descripteurs de séquences de vecteurs d'indices de textures appliquée à la base de données de séquences-personnes, en fonction du plan chromatique dans lequel sont représentées les séquences d'images.*

Ce tableau montre que les taux de discrimination entre les séquences-personnes fournis par cette approche sont très astisfaisants.

5.4.3 Analyse des ressources nécessaires

Nous avons déjà analysé les ressources nécessaires à la création des séquences de matrices de co-occurrences dans la section précédente. Le nombre d'opérations élémentaires nécessaire est de l'ordre de $3 \times 16 \times P$ par image. Le temps de calcul moyen nécessaire à la création des trois matrices de co-occurrences est de 20 ms, et ce temps de calcul ne peut pas être réduit : nous ne nous sommes donc pas appliqués à optimiser cette partie de la procédure.

Nous nous intéressons ici à la création des séquences de vecteurs d'indices de texture à partir des séquences de matrices de co-occurrences.

5.4.3.1 Ressources nécessaires au calcul des séquences de vecteurs d'indices de texture

La quantité de cellules mémoire nécessaire au stockage de trois séquences de n vecteurs d'indices s'élève à $3 \times n \times v_L$ cellules lorsque ces vecteurs d'indices sont de taille v_L chacun. Cependant, ici encore, cette quantité ne doit pas être prise en compte pour l'évaluation de la taille de la signature par descripteurs Ψ : une fois que les descripteurs sont constitués, la mémoire nécessaire au stockage des séquences de vecteurs d'indices peut être libérée.

Le temps de calcul T_L prend en compte la constitution des vecteurs d'indices de textures. On estime que l'ordre de complexité du calcul de chaque indice est proportionnel à v_C, la constitution des trois séquences de n vecteurs d'indices est alors :

$$T_L = 3 \times n \times v_L \times v_C$$

.

Nous avons mesuré une valeur moyenne de $T_L = 1500$ ms pour une séquence de 100 images, toujours à l'aide du logiciel *Scilab*. Cependant, les trois vecteurs d'indices caractérisant une image sont calculés à partir des trois matrices de co-occurrences de cette image. Même si nous avons procédé de cette manière, il n'est pas nécessaire, en pratique, de débuter le calcul des indices de textures de la première image de la séquence après la création des matrices de co-occurrences de la dernière image de la séquence.

L'évaluation du temps de calcul nécessaire à la création des séquences de vecteurs d'indices est donc également séparée de l'évaluation du temps de calcul nécessaire au tri des listes de valeurs. En effet, ce tri intervient lors de la création des descripteurs qui n'est effectuée qu'à partir du moment où les séquences de vecteurs d'indices sont entièrement constituées.

5.4.3.2 Ressources de mémoire pour le stockage des signatures

Le nombre de cellules mémoires nécessaires pour stocker une signature est fixe, comme c'est toujours le cas pour une signature par descripteurs de séquences.

Tandis que la signature d'une séquence d'images par descripteurs résumant les trois séquences de matrices de co-occurrences chromatiques nécessite $3 \times 3 \times v_C = 9 \times 256 = 2304$ cellules mémoires pour être stockée, la signature d'une séquence

d'images formée à partir des indices de texture n'en nécessite que $3 \times 3 \times v_L = 9 \times 4 = 36$, qui représente 64 fois moins de cellules mémoire.

$$W(\Psi_L[S_{Req}]) = W(\Psi_L[S_{Can}]) = 3 \times 3 \times v_L$$

avec cette fois $v_L = 4$ tandis que $v_M = 256$.

Ainsi, 36 valeurs numériques suffisent à composer cette signature. Si l'on considère qu'une valeur est représentée sur 4 octets, la taille totale de la signature Ψ_L d'une séquence-personne s'élève à 144 octets, quel que soit le nombre d'images de cette séquence.

5.4.3.3 Temps de calcul nécessaire à la création des signatures

Nous avons vu que le temps de calcul nécessaire à la détermination des indices de texture est non négligeable et que pour chaque indice à calculer à partir d'une matrice de co-occurrences, il est au moins de l'ordre de v_C opérations.

Le dernier terme à prendre en compte dans l'évaluation du nombre d'opérations élémentaires est le tri des listes de valeurs lors de la constitution des descripteurs, que nous avons exprimé de manière générale, par :

$$T(\Psi_L) = 3 \times v \times n \times log(n)$$

avec v le nombre de composantes du vecteur d'attributs employé et n le nombre d'images de la séquence considérée.

Lorsque nous passons d'une matrice de co-occurrences de taille $v_M = 256$, à un vecteur d'indices de taille $v_L = 4$, nous réduisons le nombre de tris nécessaires à la création d'une signature par descripteurs.

Pour constituer les descripteurs d'une séquence de vecteur d'attributs basée sur les indices de textures, le nombre de tris ne vaut que $3 \times 4 = 12$ contre $3 \times 256 = 768$ si l'on utilise les matrices de co-occurrences normalisées. Ceci se traduit par une mesure de temps de calcul de l'ordre de 1 à 2 ms pour constituer les trois descripteurs de séquences de vecteurs d'indices de texture.

5.4.3.4 Temps de calcul pour la comparaison des signatures

L'opération de comparaison par la norme $L1$ entre deux descripteurs de séquences de vecteurs d'indices de textures nécessite v_L opérations élémentaires. La comparaison se faisant entre descripteurs homologues, le nombre T d'opérations élémentaires est de l'ordre de :

$$T(Prox(\Psi^F_{L_{(r,g)}}[S_{Req}], \Psi^F_{L_{(r,g)}}[S_{Can}])) = 3 \times 3 \times v_L = 36.$$

Ceci correspond à un simple parcours des $v_L = 4$ cellules pour chacun des neuf descripteurs.

La mesure expérimentale du temps nécessaire à la comparaison de signatures a toujours fourni un résultat inférieur à 1 ms lorsque celle-ci est réalisée à l'aide du logiciel *Scilab*.

Nous pouvons en conclure que la signature formée des descripteurs de vecteurs d'indices de texture offre plusieurs avantages par rapport à la signature formée des descripteurs de matrices de co-occurrences normalisées : tout d'abord, elle est plus compacte et la comparaison de deux signatures est donc très rapide. Dans le cas de la constitution des matrices origine-destination, notamment, ceci représente un atout puisque le nombre de signatures à stocker et le nombre de comparaisons à effectuer constituent l'un des verrous technologiques relatifs à une exploitation en ligne.

5.5 Conclusion

Dans ce chapitre, nous avons présenté une méthodologie de comparaison de séquences de signatures d'images composées chacune de plusieurs vecteurs d'attributs.

Nous avons montré que la taille des vecteurs d'attributs employés influence la quantité de ressources nécessaires à la mise en oeuvre de ces méthodes.

Dans le cas où l'on utilise des vecteurs d'attributs de grande taille, formés à partir des matrices de co-occurrences normalisées, nous obtenons des scores de pertinence satisfaisants, notamment lorsque les signatures sont constituées à partir de séquences d'images dont les couleurs sont représentées dans les plans chromatiques (r, g) et (l_1, l_2). De plus, le temps de calcul nécessaire à la constitution des signatures est assez rapide, c'est pourquoi cette première approche semble être appropriée à la localisation automatique de personnes.

Nous avons ensuite caractérisé chaque séquence d'images par des descripteurs de séquences d'indices de texture, chaque indice étant calculé à partir de matrices de co-occurrences chromatiques. Les ressources nécessaires à la comparaison de ces descripteurs sont très faibles. C'est pourquoi cette approche semble adaptée

à la création des matrices origine-destination qui requiert un nombre élevé de comparaisons de signatures.

Les scores de pertinence obtenus par la comparaison des descripteurs d'indices de texture sur la base de données sont meilleurs que ceux obtenus à partir de la comparaison de descripteurs de matrices de co-occurrences, excepté dans le cas où les couleurs sont représentées dans le plan chromatique (H, S).

L'interprétation de la comparaison de deux matrices de co-occurrences n'est pas la même que la comparaison entre deux vecteurs d'indices extraits de ces matrices. Les matrices de co-occurrences décrivent "directement" la distribution des couleurs ou les textures des objets présents dans l'image. Une faible distance entre deux matrices de co-occurrences signifie une forte probabilité que le contenu des deux images soit similaire. Par contre, une faible distance entre deux vecteurs d'indices de texture ne signifie pas une forte probabilité que ces matrices soient similaires mais seulement que ces matrices présentent les mêmes caractéristiques.

L'interprétation d'un indice de texture extrait d'une matrice de co-occurrences est qu'il permet de quantifier un concept relatif à cette texture, comme par exemple le contraste ou l'homogénéité, à l'aide d'une grandeur scalaire.

Nous allons maintenant nous intéresser à une autre manière de caractériser, toujours à l'aide d'une valeur scalaire, l'information de texture couleur contenue dans une séquence d'images, sans employer le calcul d'indices de texture à partir de matrices de co-occurrences.

Chapitre 6

Théorie algorithmique de l'information

6.1 Introduction

Dans ce chapitre, nous abordons une théorie utilisée jusqu'ici dans d'autres domaines que celui de la comparaison d'images ou de séquences d'images : la théorie algorithmique de l'information[Del94, LV97].

Dans le premier paragraphe, nous exposons la théorie algorithmique de l'information, ou théorie de la complexité de description. Nous présentons les limites de la théorie statistique de l'information de Shannon, puis nous définissons la notion qui est à la base de la théorie algorithmique de l'information : la complexité de description de Kolmogorov.

Dans le deuxième paragraphe, nous introduisons les différents outils utilisés dans le cadre de cette théorie, à savoir des algorithmes de compression de données sans pertes. Nous y découvrons que ces algorithmes de compression peuvent être regroupés selon différentes catégories en fonction de leurs propriétés, des types de données qu'ils sont en mesure de traiter et des différents principes sur lesquels ils sont basés.

Ensuite, dans le troisième paragraphe, nous exposons les méthodes existantes de calcul de similarité basées sur la théorie de la complexité. Ces méthodes exploitent les algorithmes de compression selon le principe de la compression relative qui est expliqué.

Nous présentons les essais de calcul de ressemblance effectués selon ce principe, simplement entre deux images. Nous mettons en évidence ses limites et montrons en quoi sa mise en oeuvre ne permet pas une utilisation dans le cas de la comparaison de séquences-personnes.

6.2 La théorie algorithmique de l'information de Kolmogorov

La **théorie algorithmique de l'information** de Kolmogorov a pour objet d'étude la complexité de description d'un objet représenté sous forme de données binaires. Nous présentons tout d'abord la **théorie statistique de l'information** fondée sur la notion d'entropie de Shannon, puis nous mettons en évidence les limites de cet outil lorsque l'on désire l'utiliser pour caractériser deux données en vue de les comparer.

Nous définissons ensuite la complexité algorithmique de Kolmogorov et montrons en quoi celle-ci mesure de manière absolue l'information contenue dans une donnée. Nous exposons également les difficultés liées aux propriétés de cette théorie.

Enfin, nous revenons sur la théorie statistique de l'information afin de la situer par rapport à la théorie algorithmique de l'information, et montrons en quoi l'entropie de Shannon peut être considérée comme un cas particulier de la complexité de Kolmogorov.

6.2.1 Théorie probabiliste de l'information

Nous avons déjà évoqué l'entropie comme un indice statistique permettant de caractériser une image : à partir d'un vecteur d'attributs normalisé (comme un histogramme ou une matrice de co-occurrences), c'est-à-dire qui contient des probabilités d'occurrences ou de co-occurrences dont la somme vaut 1, la formule

$$ent(\mathbf{x}[I]) = \sum_{j=1}^{j=v} \mathbf{x}[I](j) \times log(\mathbf{x}[I](j))$$

permet de calculer une valeur scalaire.

Nous présentons ici la théorie qui est fondée sur cette notion qui sera définie plus formellement. Bien qu'elle puisse concerner n'importe quel type de données

formées de plus de deux symboles comme un texte mais aussi une image numérique couleur, dans le cadre de cette théorie on parle simplement de *messages* composés de *symboles*.

6.2.1.1 Entropie de Shannon

La *théorie de l'information* de Shannon se rencontre également sous le nom de *théorie de la transmission*[Sha48] ou *théorie de la communication*[SW49] car elle avait pour premier objet d'étude la transmission de données en télécommunication.

Lorsqu'un émetteur transmet un message à un récepteur, cette théorie permet de déterminer une représentation plus compacte du message à transmettre, tout en permettant au récepteur de reconstituer le message d'origine.

Shannon fut le premier à s'intéresser à ce qui constituera les fondements de sa théorie : la notion de *quantité d'information* contenue dans un message. Il montra que cette quantité détermine la limite au-delà de laquelle on ne peut représenter ce message sous une forme plus compacte, sous peine de rendre impossible sa reconstitution.

Pour un message de n symboles, il est possible de calculer la quantité moyenne d'informations qu'il contient grâce à la probabilité d'apparition dans ce message de chacun des symboles qui le composent. L'**entropie de Shannon**, notée E, mesure cette quantité. Elle se calcule à partir de l'histogramme H qui contient les fréquences d'apparition des n symboles qui composent le message :

$$E(H) = -\sum_{i=1}^{n} (p_i \times \log(p_i)) \tag{6.1}$$

avec p_i les n composantes de l'histogramme H.

6.2.1.2 Limites

Notons que deux histogrammes différents peuvent présenter la même entropie. En effet l'entropie résulte d'une sommation à partir des composantes de l'histogramme : si l'on inverse l'ordre dans lequel sont présentées ces composantes, on obtient un histogramme très différent du premier (sauf si celui-ci est symétrique). De même, si l'on "mélange" aléatoirement ces composantes, la somme ne change pas.

Du point de vue de la théorie de l'information cela n'a aucune conséquence car le contenu du message importe peu. Par contre, lorsqu'il s'agit de comparer deux données, l'utilisation de l'entropie comme seul indice représente une source d'erreurs. C'est la raison pour laquelle, comme nous l'avons vu pour l'exemple des images, plusieurs indices différents sont nécessaires pour constituer une signature.

Cependant la théorie statistique de l'information de Shannon représente une première approche de la notion d'*information* contenue dans une donnée. Une autre théorie, plus récente, vise à déterminer cette notion de manière plus précise en se basant, non plus sur les statistiques, mais sur l'algorithmique.

6.2.2 Théorie algorithmique de l'information

6.2.2.1 Complexité de description

La démarche initiale de Kolmogorov visait plusieurs objectifs [Kol65] : il désirait tout d'abord proposer des fondements complémentaires à la théorie des probabilités qu'il avait axiomatisée en 1933, mais également expliquer la notion de suite aléatoire. Pour ce faire, il lui fallait formaliser les notions de *simple* et de *complexe*.
L'idée principale qui est au centre de la théorie algorithmique de l'information peut s'énoncer de la manière suivante :

- est simple ce qui est facile à décrire,
- est complexe ce qui est difficile à décrire,
- est aléatoire ce qui est *le plus* difficile à décrire

La notion de complexité de Kolmogorov s'appuie sur celles de *programme* et de langage universel.

6.2.2.2 Langages, langages universels et programmes

Définition 7 (Langage) *Les langages de programmation permettent de définir les ensembles d'instructions effectuées par l'ordinateur lors de l'exécution d'un programme. Il existe des milliers de langages de programmation, la plupart d'entre eux étant réservés à des domaines spécialisés. Ils font l'objet de recherches constantes dans les universités et dans l'industrie.*

Définition 8 (Programme) *Un programme informatique indique*

à un ordinateur ce qu'il devrait faire. Il s'agit d'un ensemble d'instructions qui doivent être exécutées dans un certain ordre par un processus.

Définition 9 (Langage universel) *Un langage universel permet d'écrire un programme pour toute fonction calculable. Une fonction est calculable s'il existe une machine de Turing qui calcule cette fonction. Dans les années 30-40 on montre que la notion de fonction calculable ne dépend pas du modèle de calcul choisi, pourvu qu'il soit suffisamment général.*

Par exemple C, C++, Ada, Java sont des langages universels : une fonction calculable par une machine de Turing[Tur50] est programmable dans ces langages. Il existe cependant d'autres modèles que celui des machines de Turing, mais tous aboutissent au même ensemble de fonctions calculables, considéré comme impossible à dépasser par un modèle de calcul raisonnable (Thèse de Church-Turing[Chu36]). Ceci signifie qu'il n'existe pas de langage plus puissant qui permette de calculer plus de fonctions. Il est en revanche très facile de trouver des langages moins puissants.

6.2.2.3 Entropie algorithmique de Kolmogorov

La complexité de description de Kolmogorov[Kol65] (ou entropie algorithmique) d'un objet fini x est la longueur du plus court programme permettant le calcul de x, lorsque ce programme est écrit dans un langage universel.

Soient δ une suite binaire et λ un langage universel ; la complexité de Kolmogorov de δ est définie par : $K_\lambda(\delta) = min\{lg(p) : \lambda(p) = \delta\}$ où $lg(p)$ désigne la longueur du programme p, et $\lambda(p)$ le résultat de l'exécution du programme p écrit dans le langage universel λ.

Si l'on considère qu'un programme est assimilable à une description, $K_\lambda(\delta)$ représente la longueur de la meilleure description que l'on puisse obtenir de l'objet δ. La complexité de Kolmogorov est indépendante, à une constante additive près, du langage dans lequel le programme minimal p est écrit.

Elle représente une mesure absolue et objective de la quantité en information contenue dans un objet. Intuitivement, $K(\delta)$ représente la quantité minimale d'information nécessaire à la génération de δ, et ne peut excéder la longueur de cet objet (à une constante près).

6.2.2.4 Difficultés liées aux propriétés théoriques

Il est prouvé que la complexité de Kolmogorov est non-calculable en raison de l'indécidabilité de l'arrêt d'un programme [Tur36]. Précisément cela signifie qu'il n'existe pas de programme p qui pour tout δ donné calcule $K(\delta)$ (en un temps fini).

Cela n'interdit pas d'approcher $K(\delta)$ par valeurs supérieures décroissantes au sens large, ni même pour certains δ particuliers de calculer exactement $K(\delta)$.

Puisque nous ne savons pas quels sont, parmi les programmes suceptibles d'engendrer δ, ceux qui s'arrêtent, on ne peut pas les essayer tous. Nous ne pouvons donc pas avoir de certitude quant à la minimalité de la longueur d'un programme engendrant une donnée δ.

Cela ne signifie pas pour autant qu'il est impossible d'utiliser la complexité de Kolmogorov. En pratique, nous pouvons calculer une approximation de $K(\delta)$ par valeurs supérieures[LV97]. Ainsi, pour une donnée δ, on peut définir une suite $K_1(\delta), K_2(\delta), \ldots, K_n(\delta), \ldots$ qui converge vers $K(\delta)$ de manière décroissante.

Il n'est cependant pas possible d'évaluer la qualité d'une telle approximation : l'erreur $|K_n(\delta) - K(\delta)|$ ne peut pas être évaluée.

6.2.3 Informations de Shannon et de Kolmogorov

La complexité algorithmique de Kolmogorov est aussi appelée entropie algorithmique. Le lien avec l'entropie de Shannon n'est pas uniquement "linguistique" mais il existe véritablement une relation que nous nous contentons d'introduire ici.

L'entropie de Shannon représente le contenu en information lorsque les probabilités d'apparition des symboles sont différentes. Elle est spécifique au codage de Shannon qui peut être considéré comme optimal dans son cadre parfaitement délimité où chaque symbole est simplement remplacé par un autre symbole codé de manière différente.

L'entropie de Kolmogorov étend cette notion en considérant l'entropie de Shannon comme un cas particulier dans lequel la notion de langage universel disparaît au profit d'un langage spécifique qui est le codage de Shannon. Ainsi, il est possible de considérer la théorie de Shannon comme une théorie probabiliste de l'information qui est contenue dans la théorie algorithmique de l'information de Kolmogorov[Del94].

La théorie abordée par Kolmogorov est donc plus générale que la théorie de Shannon et elle donne une mesure absolue de la quantité d'information d'un objet individuel, contrairement à une mesure moyenne de la qualité d'information nécessaire à la transmission d'un objet.

Nous allons voir maintenant comment approcher de manière pratique cette notion théorique de quantité d'information algorithmique ou complexité.

6.3 Complexité et compression

6.3.1 Liens entre la complexité de description et la compression de données

Nous ne pouvons pas en général déterminer le plus petit programme qui permet de produire une donnée lorsqu'il est exécuté dans un langage universel. Cependant nous pouvons déterminer ce plus petit programme lorsqu'il est exécuté sur certains langages particuliers.

Si l'on considère qu'un algorithme de compression/décompression est assimilable à un langage donné, le plus petit programme capable d'engendrer la donnée x dans ce langage est simplement le fichier compressé, qui est bien le plus petit qui puisse exister pour cet algorithme de compression/décompression. Notons enfin qu'il n'est pas nécessaire de décompresser le fichier puisque seule la longueur du programme (donc celle du fichier compressé) nous intéresse dans ce cas précis.
Il existe un grand nombre d'algorithmes de compression de données. On peut dans un premier temps les classer en deux catégories. D'une part on trouve ceux qui sont capables de reproduire les données à l'identique, d'autre part ceux qui altèrent plus ou moins ces données. Selon le type de données à compresser et selon les besoins de l'utilisation on fera appel à l'une ou l'autre de ces deux catégories d'algorithmes.

Deux exemples populaires d'algorithmes de compression dits *avec pertes* sont le Jpeg pour les images et le Mp3 pour la musique. La différence entre les données d'origine et les données décompressées est le plus souvent imperceptible par un être humain. Cependant, puisque ces données ne sont pas reconstituées exactement à l'identique, ces algorithmes de compression ne sont pas adaptés à la théorie de Kolmogorov.

Nous nous intéresserons uniquement à l'autre catégorie d'algorithmes, dits *sans pertes*, car la définition de la complexité $K_\lambda(\delta)$ d'une donnée δ relativement à un langage λ précise implicitement que le programme fournit exactement la donnée δ et non une donnée δ' qui lui ressemble.

6.3.2 Algorithmes de compression de données sans pertes

Nous allons donc nous intéresser à plusieurs méthodes de compression de données [Sal04]. Quelle que soit la méthode, compresser une donnée consiste à représenter sous forme descriptive l'information qu'elle contient. Le but est de faire en sorte que la donnée compressée occupe moins de place que la donnée brute. Dans le cas contraire, le compresseur ne remplit pas son objectif et n'est pas approprié.
Une fois la donnée compressée, son stockage ou sa transmission est possible à moindres coûts. Ce principe d'économie dans la représentation des données est appliqué dès 1838 avec l'apparition du code morse où les lettres les plus fréquentes en Anglais sont codées sur un seul signal (e='.' et t='-').

6.3.2.1 Compresseurs statistiques : codes à longueur variable

Les compresseurs statistiques sont basés sur la théorie de l'information de Shannon. Le premier algorithme de compression de données binaires est d'ailleurs dû à Shannon[SW49] et Fano[Fan61]. Leur principe consiste à représenter les données à l'aide de codes de longueur variable.
Notons que le morse est basé sur le même principe. Ces codes sont déterminés à partir de la fréquence d'apparition des motifs élémentaires qui composent les données : plus un motif est fréquent, plus le code utilisé pour le représenter est court. Par exemple dans la langue française, la lettre e est la plus fréquente. Un compresseur statistique appliqué sur un texte rédigé en français lui attribuerait le code le plus court. A l'inverse, les lettres w ou k auraient des codes longs. Si le texte était écrit en anglais, les fréquences d'apparitions des lettres seraient différentes ; celles-ci seraient donc représentées suivant un jeu de codes différent.

Ce principe a été repris par Huffman[Huf52] en 1951 pour donner son célèbre algorithme basé sur les arbres du même nom. La différence se situe uniquement au niveau de la construction de l'arbre, qui est beaucoup plus rapide chez Huffman.

La compression d'une donnée nécessite ici deux lectures et comporte trois étapes. La première étape est l'analyse statistique : une première lecture permet de déterminer les fréquences d'apparition des symboles.

Prenons par exemple la chaine `ABABCEAFGA`, en considérant que l'on dispose de l'alphabet formé des lettres allant de `A` à `G`. Il y a huit symboles possibles, ce qui signifie que le texte d'origine peut être codé avec 3 bits pour chaque symbole.

Si l'on prend le codage qui est fourni dans le tableau 6.1,

A	B	C	D	E	F	G	H
000	001	010	011	100	101	110	111

TAB. 6.1 – *Codage d'origine de la chaîne "ABABCEAFGA".*

la chaine binaire est "000001000001010100000101110000", de longueur 30 bits. On établit ensuite le tableau 6.2 qui comptabilise les fréquences de chaque lettre.

A	B	C	E	F	G	D	H
4	2	1	1	1	1	0	0

TAB. 6.2 – *tableau des fréquences de chaque lettre de la chaîne "ABABCEAFGA".*

La seconde étape est la construction d'un arbre binaire en fonction des fréquences (voir figure 6.1). Chaque feuille de l'arbre représente un des symboles. Plus la fréquence d'un symbole est importante, plus la feuille correspondante de l'arbre se trouve proche de la racine.

Une fois cet arbre constitué, il permet de déterminer les nouveaux codes qui seront affectés à chaque symbole. L'arbre est d'abord représenté sous la forme du tableau 6.3.

Il constitue la première partie du message. Sans cet arbre, il n'est pas possible de décoder la donnée. Ces codes sont fournis dans le tableau 6.4.

Enfin, la troisième étape est le codage proprement dit des données. La donnée est parcourue une seconde fois pour remplacer chaque symbole par son nouveau code.

La chaine codée est : "10110100100011100010000011" de longueur 25 bits.

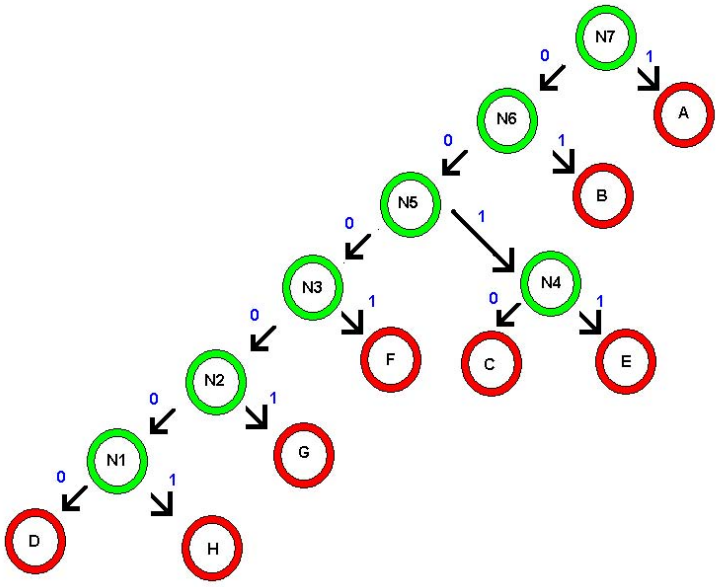


FIG. 6.1 – *Principe de compression statistique*

N7	N6 + A	10
N6	N5 + B	6
N5	N3 + N4	4
N4	C + E	2
N3	N2 + F	2
N2	N1 + G	1
N1	D + H	0

TAB. 6.3 – *Représentation de l'arbre des fréquences de la chaîne "ABABCEAFGA" sous forme de tableau.*

L'entropie de Shannon d'un symbole donne donc le nombre de bits qui seront nécessaires au codage de ce symbole, tandis que l'entropie de la donnée (succession de symboles) fournit le nombre moyen de bits qui seront nécessaires au codage de cette donnée.

Par exemple si une donnée binaire contient la succession des résultats de tirages à pile ou face, l'entropie moyenne est de 1 bit car les deux symboles (0/1 ou pile/face sont équiprobables à 1/2 contre 1/2). Si maintenant la pièce est truquée de sorte que la probabilité des résultats est de 1/3 pour pile (0) et 2/3 pour face (1), on s'attend à rencontrer davantage de 1 que de 0. Il y a moins

A	1
B	01
C	0010
D	000000
E	0011
F	0001
G	00001
H	000001

TAB. 6.4 – *Nouveaux codes de la chaîne "ABABCEAFGA".*

d'incertitude donc moins d'information. L'entropie moyenne est alors de 0.918 bits.

Pour une pièce non truquée l'entropie vaut :

$$E(H) = -\frac{1}{2} \times log_2(\frac{1}{2}) - \frac{1}{2} \times log_2(\frac{1}{2}) = 1bit$$

Pour une pièce truquée l'entropie vaut :

$$E(H) = -\frac{1}{3} \times log_2(\frac{1}{3}) - \frac{2}{3} \times log_2(\frac{2}{3}) = 0.918bit$$

Nous venons de présenter une première famille d'algorithmes de compression dont le principe consiste à recoder tous les symboles du message en utilisant un code à longueur variable, mais en conservant l'ordre dans lequel ils se présentent et en les traitant de manière indépendante.

Nous allons nous intéresser maintenant à une seconde famille d'algorithmes de compression qui sont capables, cette fois, de repérer et d'exploiter des répétitions de symboles.

6.3.2.2 Compresseurs à dictionnaires

Le premier compresseur à dictionnaire a été conçu en 1977 par Abraham Lempel et Jacob Ziv[ZL77]. Cet algorithme nommé LZ77 a été le point de départ de plusieurs variantes comme l'algorithme LZ78 ou encore le plus connu, l'algorithme LZW conçu en 1984 par Terry Welch[Wel84]. On parle même aujourd'hui de la famille des compresseurs LZ* pour désigner les compresseurs à dictionnaires.

Le principe de compression repose sur l'utilisation d'un dictionnaire qui contient les références aux parties déjà traitées de la donnée à compresser. Le dictionnaire fait donc partie intégrante du message, contrairement à Shannon où l'arbre de codage est distinct des données codées.

Les algorithmes LZ* sont des compresseurs qui repèrent les répétitions de successions de motifs. Par exemple, toujours dans un texte en français, les mots 'de', 'les', 'des' ne seraient représentés qu'une fois et chacune de leurs occurrences serait remplacée par une référence à la première occurrence du mot.

Le principe paraît surprenant à première vue puisque l'on code les symboles sur davantage de bits que dans le message d'origine. En effet chaque caractère est codé avec une taille supérieure à la taille d'origine.

On dispose alors de davantage de symboles ce qui permet généralement de ne coder qu'une seule fois une suite de symboles même si celle-ci se répète, puisque les codes suivants feront seulement référence à sa première occurrence.

Le principe de cette seconde famille d'algorithmes est d'exploiter les répétitions de successions de plusieurs symboles. Leur point commun est l'utilisation d'un dictionnaire.

Nous allons maintenant présenter une troisième famille de compresseurs, plus récente.

6.3.2.3 Compresseurs arithmétiques

Nous avons d'abord mentionné les compresseurs de Shannon / Huffmann mais en toute rigueur ceux-ci sont davantage des encodeurs plutôt que de véritables compresseurs. En effet ils se contentent de recoder les données d'une manière différente et pertinente, permettant ainsi de représenter la même information en utilisant moins de bits. Chaque symbole est représenté sur une longueur de bits : une fois compressé le fichier contient "la même" suite de symboles, sans exploiter les répétitions de suites de symboles comme peuvent le faire les algorithmes à dictionnaires.

Les compresseurs arithmétiques[WNC87] reposent sur un principe similaire à celui des codes à longueur variable, mais consistent cette fois à construire le code complet du message sous la forme d'un seul nombre réel plutôt que d'associer un code spécifique à chaque symbole de l'alphabet.

La construction du code commence, comme pour le codage de Huffmann, par le relevé des probabilités d'occurrence $p(\alpha_i)$ de chaque symbole. Puis, à chaque symbole est associé un sous-intervalle Q de $[0,1[$ de longueur $p(\alpha_i)$, de sorte que :

$$Q_{\alpha_0} = [0, p(\alpha_0)[$$
$$Q_{\alpha_1} = [p(\alpha_0), p(\alpha_1)[$$
$$\dots$$

Ainsi, à chaque symbole α_i est associé l'intervalle :

$$Q_{\alpha_i} = [\sum_{q=0}^{q=i-1} p(\alpha_q), \sum_{q=0}^{q=i} p(\alpha_q)\}[$$

Le codage d'une séquence consiste à générer une suite d'intervalles emboîtés convergeant vers un réel de $]0,1[$ en prenant pour chaque symbole émis le sous-intervalle Q_{α_i} déterminé par sa probabilité d'occurrence dans l'intervalle courant correspondant aux événements déjà codés.

Bien que ce principe soit simple et efficace en théorie, il n'est pas réalisable en pratique sur une machine informatique car la précision numérique nécessaire est a priori infinie.

Il existe cependant une adaptation de l'algorithme à l'arithmétique entière, mais nous n'entrerons pas dans le détail de sa description. Nous pouvons retenir que les codages arithmétiques représentent une classe émergente d'algorithmes de compression qui concurrencent peu-à-peu les codes à longueur variable du type Huffmann.

Plusieurs variantes existent et, sous ces formes modifiées, le codage arithmétique est de plus en plus préconisé par les grands standards de compression.

De plus, ils sont particulièrement bien adaptés aux signaux numériques car ils sont basés sur une modélisation du signal à compresser. Ainsi, il est plus efficace de compresser d'une part un modèle simple qui approche le signal accompagné d'autre part d'une erreur de prédiction entre le modèle et le signal, plutôt que le signal d'origine.

6.3.2.4 Transformation de Burrows-Wheeler

Le compresseur bzip2 tend à remplacer progressivement le classique gzip. Développé par Burrows et Wheeler[Fen96], il permet d'obtenir des taux de compres-

sion supérieurs de 5 à 10% par rapport à son concurrent. Ce résultat est d'autant plus surprenant qu'il ne provient pas directement des performaces de l'algorithme employé, mais d'une simple transformation des données avant de les compresser.

L'objectif de cette transformation des données est de les représenter sous une forme plus facile à compresser.

Le principe est le suivant : sur un bloc de données monodimensionnelles, on commence par déterminer toutes les permutations circulaires possibles. Sur un bloc de N caractères, il y a donc N permutations soit N chaînes. Ces chaînes sont ensuite triées dans l'ordre lexicographique. On repère alors la position de la chaîne d'origine. Puis on constitue une nouvelle donnée de N caractères en parcourant la dernière colonne. C'est cette chaîne qui sera compressée.

Prenons par exemple la chaîne `propre`. Les six permutations possibles sont :

1. `propre`
2. `roprep`
3. `oprepr`
4. `prepro`
5. `reprop`
6. `epropr`

on les trie selon l'ordre lexicographique :

1. `epropr`
2. `oprepr`
3. `prepro`
4. `propre`
5. `reprop`
6. `roprep`

on note que la chaîne d'origine se trouve en quatrième position.

lorsqu'on lit la dernière colonne de haut en bas, on obtient la chaine `rroepp`. On remarque qu'elle comporte des régularités : certaines lettres se suivent et sont identiques, la chaine sera donc plus facile à compresser.

Pour reconstituer la donnée d'origine, il suffit de disposer de la dernière colonne et de la position de la donnée d'origine.

Il est a priori étonnant que cette chaîne formée à partir de la dernière colonne se compresse si bien. L'explication est la suivante : le tri regroupe les lettres identiques en première colonne ; la dernière colonne contient donc les lettres qui précèdent celles de la première colonne. Ainsi, les lettres de la dernière colonne sont regroupées par petits paquets qui sont souvent formés des mêmes lettres. En français, par exemple, la lettre h est souvent précédée de la lettre t.

6.3.3 Compression d'images sans pertes

Nous avons vu précédemment plusieurs techniques de compression de données sans pertes. Jusqu'à maintenant nous nous sommes appuyés sur la compression de textes, dans un souci de simplicité, et nous pouvons aisément étendre les principes à des données binaires non nécessairement textuelles. Cependant ces techniques concernent des données monodimensionnelles.

La compression d'image [Inc03] diffère en ce sens qu'une image est une donnée bidimensionnelle. Nous nous attendons à ce que les méthodes de compression sans pertes tirent parti de cet aspect, or la plupart des compresseurs d'images sans pertes exploitent peu ou pas du tout cet aspect.

Remarquons enfin que certains formats d'images se prétendent sans pertes alors que seul l'algorithme de compression employé est sans pertes. Ainsi, lors de la conversion d'un format brut vers un format compressé, la phase de compression est précédée par un filtrage de l'image afin de réduire le bruit. C'est le cas, par exemple, du format GIF.

6.3.3.1 Jpeg sans pertes = Shannon

Le format JPEG est un standard international qui date de 1990. C'est un des formats d'images les plus connus à l'heure actuelle. A l'origine, il n'a pas été conçu pour la compression sans pertes mais plutôt, comme le mp3, pour reconstituer une image si peu différente de l'image d'origine que l'être humain n'est pas capable de remarquer cette différence. Nous ne détaillerons pas son fonctionnement et ses nombreuses variantes car celles-ci sont assez complexes. Précisons simplement qu'il fait appel à la transformée de Fourier ou la transformée en cosinus discrète, ou encore à la compression fractale pour la norme jpeg2000.

Notons enfin que le principe de Huffman s'applique aux données transformées ; il s'agit ici de compresser des données qui ont subi une transformation particulière

qui les rend plus aisément compressibles. L'intérêt de ce format est qu'il prend en compte l'aspect bidimensionnel de l'image.

Toutefois, seul le format particulier JPEG-LS[WSS00] est sans pertes. Celui-ci étant un format propriétaire, il est relativement rare et bien qu'il soit possible de décompresser une image codée dans ce format, nous n'avons pas trouvé de logiciel gratuit permettant la compression.

6.3.3.2 GIF et PNG = LZ*

Le format PNG (Portable Network Graphics) a été mis au point en 1995 afin de fournir une alternative libre au format propriétaire GIF basé sur l'algorithme de compression LZW. Ainsi PNG est également un acronyme récursif pour PNG's Not Gif. PNG n'exploite pas exactement l'algorithme LZW ; il offre en outre des taux de compression meilleurs que le format GIF.

Son principe ne diffère pas de celui de l'algorithme de compression de données monodimensionnelles : une image comporte de nombreuses répétitions de mêmes couples de pixels, chaque couple de pixels est représenté une fois et chacune de ses répétitions y fait référence.

Ces couples de pixels sont ce que mesurent les matrices de co-occurrences de Haralick, mais dans une direction donnée au lieu d'un voisinage 3×3 puisque l'algorithme parcourt l'image ligne par ligne uniquement. Bien que performant dans le cas de la compression sans pertes, ce format n'exploite donc pas l'aspect bidimensionnel des données.

6.3.3.3 PCX et TIFF = RLE

Les formats TIFF et PCX sont basés sur la compression RLE (Run Length Encoding). Cet algorithme repère les successions de motifs identiques pour les remplacer par un seul de ces motifs accompagné du nombre de répétitions. Ainsi on remplacerait aaaaa par 5a.

Peu efficace sur du texte, cet algorithme peut fournir de bons résultats sur certaines images, notamment les images de synthèse. Ici encore, il est à rapprocher d'un attribut d'images employé en traitement d'images vu dans le chapitre 3, appelé isosegment ou "run-length", qui mesure les longueurs de successions de pixels identiques.

6.3.3.4 Autres compresseurs particuliers

Nous avons vu que les formats de compression d'images sans pertes les plus populaires n'exploitent finalement pas l'aspect bidimensionnel des données, ceci en raison de la complexité d'un tel algorithme.

Nous avons trouvé plusieurs compresseurs capables d'exploiter cette propriété, mais sans pour autant obtenir de détails sur l'algorithme employé, ni même pouvoir disposer d'un programme implémentant ces algorithmes.

En effet, si la fonction de décompression d'un algorithme est souvent libre, gratuite, ou intégrée aux logiciels de visualisation d'images, on peut regretter que la fonction de compression soit payante ou difficile à trouver.

Parmi ces algorithmes de compression d'images couleur 24 bits, nous pouvons citer l'implémentation *HP* du format JPEG-LS ou encore les logiciels BOA 0.58b, BMF 0.5 ou UHIC 1.0.

D'autres programmes de compression de données binaires proposent une option qui permet à l'utilisateur de préciser que les données à compresser sont des images mais nous ne savons pas s'ils exploitent l'aspect bidimensionnel des données :

- Eri32 3.7fre
- Arhangel 1.39a6
- UFA32 0.04b1
- RKIVE 1.92b1
- UHarc 0.2
- ESP 1.92
- WinRar 2.50

6.3.3.5 Synthèse

Nous avons vu qu'il existe de nombreux algorithmes de compression de données sans pertes. Leur nombre et leur variété s'expliquent d'abord par la diversité des types de données : il y a autant de familles de compressseurs que de familles de données. Mais elle s'explique également par l'expérience accumulée par les chercheurs depuis plusieurs années.

Nous avons vu également que tout décompresseur est un langage particulier qui est capable d'exécuter des programmes particuliers (les fichiers compressés) pour produire des données. Ainsi la complexité algorithmique d'une donnée, relativement à un langage particulier, est la taille du fichier compressé à l'aide de

l'algorithme de compression qui implémente ce langage. Cependant il n'est pas possible de déterminer la complexité de Kolmogorov d'une donnée à l'aide d'un compresseur, nous ne pouvons que nous en approcher car nous ne pouvons pas essayer tous les compresseurs possibles.

Nous allons maintenant découvrir comment la théorie de la complexité de Kolmogorov permet de comparer deux données à l'aide de compresseurs sans pertes particuliers.

6.4 Distance Informationnelle et comparaison par compression relative

Après avoir présenté la théorie algorithmique de l'information de Kolmogorov et montré le lien avec la compression de données sans pertes, nous abordons une autre notion susceptible d'être exploitée dans nos travaux : la complexité relative[BGL+98]. Définie récemment par un groupe de chercheurs (C. Bennett, P. Gacs, M. Li, P. Vitanyi et W. Zurek) issus de disciplines variées (thermodynamique, mathématiques et informatique), celle-ci fait aujourd'hui l'objet de recherches très actives. Nous exposons ensuite une méthode de comparaison de deux données basée sur cette notion et dont la mise en oeuvre s'appuie sur l'utilisation de compresseurs.

6.4.1 Distance Informationnelle et complexité relative

La distance informationnelle, ou complexité relative, repose sur un principe simple que nous commencons par énonçer. Puis nous introduisons la distance de transformation qui a permis de définir la distance informationnelle.

6.4.1.1 Principe de Description de Longueur Minimale

Le Principe de Description de Longueur Minimale [Ris78] s'énonce de la manière suivante : *La meilleure théorie pour rendre compte d'un ensemble de données est celle qui minimise la somme de :*

- *la longueur, en bits, de la description de la théorie ; et*
- *la longueur, en bits, de la donnée lorsqu'elle est codée par cette théorie.*

Ceci signifie que pour décrire une donnée on préfèrera toujours le modèle dont la description est la plus courte possible pour expliciter la manière dont cette donnée est obtenue.

6.4.1.2 Distance de Transformation [VL00]

En se basant d'une part sur le principe de description de longueur minimale qui décrit une donnée, et en considérant, d'autre part, que la transformation d'une donnée δ en une donnée γ constitue elle-même une donnée, la distance de transformation entre une donnée δ et une donnée γ est définie par la quantité d'information nécessaire pour produire γ lorsque l'on connaît δ, autrement dit pour exprimer γ en fonction de δ[Del94].

Si les deux données δ et γ se ressemblent, exprimer γ en fonction de δ nécessite beaucoup moins d'informations que s'il faut exprimer γ seulement, car il suffit de recopier les parties identiques de δ. Par contre si δ et γ sont très différents, le fait de connaître δ n'apporte rien et il faut presque autant d'informations pour exprimer γ seulement que pour exprimer γ d'après δ.

6.4.1.3 Complexité de Kolmogorov relative

En se basant sur le principe de description minimale et la complexité de Kolmogorov[VL00], Bennett, Zurek, Gacs, Li et Vitanyi définissent la distance informationnelle[BGL+98] ou complexité de Kolmogorov relative d'une donnée γ par rapport à une donnée δ par *la longueur du plus petit programme p qui permet de passer de δ à γ lorsque ce programme est écrit dans un langage universel λ*. On note $K(\gamma|\delta)$ cette longueur.

$$K(\gamma|\delta) = min\{lg(p) : \lambda(p, \delta) = \gamma\} \tag{6.2}$$

6.4.1.4 Applications

Cette notion de complexité relative est déjà exploitée dans des domaines tels que la comparaison de génomes en bioinformatique[VDR99], la comparaison de textes littéraires[CV05] ou de copies d'examens pour détecter les fraudes et la comparaison de morceaux de musique[CVDW04].

Ainsi, la classification de génomes permet de reconstituer les relations phylogénétiques entre les espèces animales conformément à la taxonomie établie par les biologistes ; la classification d'oeuvres littéraires permet de regrouper les textes selon leur auteur, et la classification de morceaux de musique regroupe les oeuvres en fonction de leurs compositeurs.

A l'heure actuelle, les résultats obtenus sont assez satisfaisants dans ces domaines. Par contre, l'application à la comparaison d'images est encore peu développée bien que des travaux soient en cours, notamment sur la reconnaissance optique de caractères manuscrits[CV05].

6.4.2 Compression Relative

Nous allons maintenant nous intéresser à la mise en oeuvre de ce principe théorique. Celle-ci se fait en utilisant certains algorithmes de compressions sans pertes qui possèdent une propriété particulière car celle-ci est exploitée dans le cadre de la comparaison de données par compression relative.

Nous présentons d'abord cette propriété avant de détailler la mise en oeuvre de la compression relative, puis nous terminons par une explication du succès de ce principe.

6.4.2.1 Compresseurs normaux

Un compresseur normal est un compresseur de données sans pertes qui possède une propriété particulière. Il est capable de repérer toute répétition, quelle que soit sa longueur.

En pratique, si l'on compresse une donnée Δ formée d'une donnée δ suivie de la même donnée δ, le compresseur détecte cette répétition et utilise très peu d'informations pour compresser la seconde partie de Δ.

Il lui suffit d'indiquer au décompresseur qu'il doit recopier une seconde fois ce qui a déjà été décompressé, ce qui nécessite très peu de bits. Ainsi, $K(\Delta) = K(\delta\delta) \approx K(\delta)$.

Parmi les compresseurs normaux, nous pouvons citer deux programmes libres : PPMZ, qui est l'algorithme utilisé par [LV97] et UHARC qui présente l'avantage de fournir trois algorithmes de compression différents, chacun respectant la propriété de normalité.

6.4.2.2 Mise en oeuvre de la Compression Relative

La mise en oeuvre de la compression relative d'une donnée γ par rapport à une donnée δ se fait de la manière suivante [CV05, CVDW04] : dans un premier temps on compresse les deux données δ et γ, de manière indépendante. Ensuite, on crée une troisième donnée formée de δ suivi de γ que l'on compresse également.

Notons $\delta\gamma$ cette donnée.

Puisque la taille d'un fichier compressé équivaut à la longueur d'un programme écrit dans un certain langage correspondant à l'algorithme de compression, on obtient alors trois longueurs de programmes soit trois complexités de description différentes : $K(\delta)$, $K(\gamma)$ et $K(\delta\gamma)$.

Dans le cas où $\delta = \gamma$ et à condition que le compresseur utilisé soit normal on a : $K(\delta) = K(\gamma) \simeq K(\delta\gamma)$, la très légère différence étant due au fait qu'il faut tout de même indiquer à l'algorithme l'instruction *"recopier une seconde fois ce qui vient d'être produit"*.

6.4.2.3 Explications

Nous avons vu qu'un compresseur normal est capable de détecter le fait qu'une donnée est composée de deux données identiques, si tel est le cas. Ceci représente un cas simple de répétition : en compressant la donnée $\delta\gamma$ il est capable de repérer ce qui, dans la donnée γ, se répète dans la donnée δ. Les compresseurs normaux sont donc en mesure de repérer certaines formes de redondance au sein d'une donnée et ainsi de supprimer certaines informations inutiles dans la description de cette donnée.

Si δ et γ sont très différents, il y a très peu ou pas du tout de redondances lors de la compression de la seconde partie de $\delta\gamma$ correspondant à δ. Le fait de fournir δ n'est d'aucune utilité pour compresser γ et n'apporte rien dans la compression de $\delta\gamma$. La complexité $K(\delta\gamma)$ est donc proche de $K(\delta) + K(\gamma)$.

En revanche, si δ et γ se ressemblent, la compression de la seconde partie de $\delta\gamma$ exploite les nombreuses redondances en faisant références aux parties de δ. Fournir δ est alors très utile pour compresser γ et facilite donc la compression de $\delta\gamma$.

Ainsi, la complexité $K(\delta\gamma)$ est sensiblement inférieure à $K(\delta) + K(\gamma)$.

6.4.3 Distance de Compression Normalisée

Nous avons présenté le principe selon lequel la compression relative permet d'obtenir une indication sur le contenu commun en information de deux données. Cependant, il reste à appliquer ce principe dans une formule mathématique qui

fournit une grandeur numérique qui représente la ressemblance entre ces deux données.

Nous définissons ici plus formellement la façon dont on calcule la distance entre deux objets en fonction des valeurs $K(\delta)$, $K(\gamma)$ et $K(\delta\gamma)$. La formule proposée par [LCL+04] définit la distance de compression normalisée qui s'exprime de la manière suivante :

$$d_K(\delta,\gamma) = 1 - \frac{K(\delta) + K(\gamma) - K(\delta\gamma)}{\max(K(\delta), K(\gamma))} \tag{6.3}$$

Les termes de normalisation sont utiles dans le cas où les ordres de grandeur de δ et de γ sont différents, par exemple si δ est deux fois plus long que γ.

Cette quantité $d_K(\delta,\gamma)$ possède asymptotiquement les propriétés d'une distance métrique au sens mathématique :

- Lorsque $\delta = \gamma$ on a bien $d_K(\delta,\gamma) = 0$ (en fait $d_K(\delta,\gamma) = o(|\delta|)$.
- La distance est symétrique : $d_K(\delta,\gamma) = d_K(\gamma,\delta)$.
- L'inégalité triangulaire est respectée : $d_K(\delta,\gamma) \leq d_K(\delta,\eta) + d_K(\eta,\gamma)$.
- Les valeurs sont comprises entre 0 et 1.

Notons enfin qu'il existe plusieurs variantes à cette formule, mais qu'elle sont toujours basées sur les trois quantités $K(\delta)$, $K(\gamma)$ et $K(\delta\gamma)$.

6.4.4 Essais et limites de la compression relative appliquée aux séquences-personnes

A l'heure actuelle, la compression relative et la distance de similarité sont essentiellement exploitées pour la comparaison de données monodimensionnelles comme des séquences génétiques et des textes littéraires ou encore des morceaux de musique convertis en textes[CVDW04]. Nous avons essayé d'appliquer cette méthode à des données bidimensionnelles, à savoir des images. Nous présentons ici notre démarche et montrons les limites de la compression relative.

6.4.4.1 Mise en oeuvre en deux dimensions

Dans un premier temps, nous exploitons pleinement l'aspect bidimensionnel des données. Afin de comparer deux images à l'aide de la compression relative, nous considérons que la concaténation de deux images (figures 6.2 et 6.3) consiste simplement à les juxtaposer afin d'en former une seule. Ceci peut être fait de deux manières différentes, comme l'indiquent les figures 6.4 et 6.5.

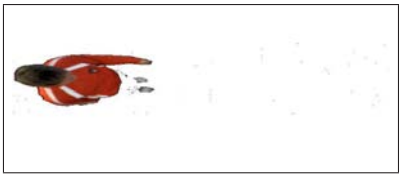


FIG. 6.2 – *Image complète représentant l'individu $\mathcal{G}$*

FIG. 6.3 – *Image complète représentant l'individu $\mathcal{E}$.*

FIG. 6.4 – *Concaténation de deux images pour en former une seule, dans le sens horizontal*

FIG. 6.5 – *Concaténation de deux images pour en former une seule, dans le sens vertical*

La compression proprement dite consiste soit à convertir les images dans un des formats graphiques sans pertes présentés dans le paragraphe précédent, soit à compresser les fichiers à l'aide de compresseurs de données binaires.

Le premier inconvénient d'ordre pratique, rencontré lors de la mise en oeuvre,

est dans la génération de la donnée $\delta\gamma$. L'idée est de concaténer deux données δ et γ (ici deux images) afin d'en former une troisième. Nous avons donc naturellement envisagé de juxtaposer les deux images pour n'en former qu'une seule. Ceci demande que les images aient le même nombre de colonnes, ou le même nombre de lignes. Bien que, dans le cadre de nos expériences, les images ont toutes la même résolution, cette propriété est assez rare en pratique.

Une autre difficulté rencontrée est indirectement liée à la nature des données dans le cas où nous utilisons des algorithmes de compression d'image, et non des algorithmes de compression de données binaires. Nous avons précisé que les compresseurs d'images sans pertes exploitent rarement l'aspect bidimensionnel de ces données.

Ainsi, le fait de juxtaposer deux images identiques selon la verticale ou selon l'horizontale a son importance et influence le résultat de la compression de l'image résultat. Considérons le cas où les deux images concaténées δ et γ sont la même image, dans le but de tester la propriété de *normalité* des compresseurs d'images sans pertes, c'est-à-dire vérifier que $K(\delta\delta) = K(\delta)$.

Si le compresseur utilisé est celui du format PNG, on s'aperçoit alors que cette propriété n'est respectée que dans le cas où la concaténation se fait dans le sens horizontal. Le format Jpeg2000 sans pertes, lui, ne vérifie la propriété de normalité dans aucun des cas.

Face à la difficulté de trouver un convertisseur d'images exploitant l'aspect bidimensionnel, nous nous sommes tournés vers les algorithmes de compression de données binaires qui disposent d'une option particulière indiquant que les données sont des images : PPM et UHARC. Nous constatons que, cette fois, la propriété de normalité est vérifiée pour les deux types de concaténation (verticale et horizontale).

Cependant, afin d'éprouver les performances de ces algorithmes dans le cas où les deux images sont proches sans toutefois être la même, nous avons concaténé cette fois deux images consécutives d'une même personne. Celles-ci sont donc acquises à 40 millisecondes d'intervalle seulement.

Nous résumons dans le tableau 6.5 les différentes tailles de fichiers obtenus à l'aide de certains algorithmes de compression :

Nous notons g_1 et g_2 deux images consécutives extraites de la séquence personne $\mathcal{G}^{(1)}$, et e_1 une image extraite de la séquence-personne $\mathcal{E}^{(1)}$.

La concaténation verticale des deux images g_1 et g_2 est notée $(g_1g_2)_v$, la concaténation horizontale des deux images g_1 et e_1 est notée $(g_1e_1)_h$. Les tailles sont données en Kilo-octets.

Les cas où la propriété de normalité est (quasiment) vérifiée sont surlignés : on constate que seul le format PNG la vérifie, dans le cas horizontal. En revanche, les compresseurs de données binaires UHARC et PPM la vérifient dans tous les cas. En effet la taille des images $(g_1g_1)_h$ et $(g_1g_1)_v$ compressées par ces algorithmes est quasiment égale à la taille de l'image g_1, compressée par ces mêmes algorithmes.

image	BMP	PNG	Jpeg2000	PPMZ	UHARC-3	UHARC-x	UHARC-z
g_1	649	43	42	43	41	35	43
g_2	649	42	41	43	41	34	43
e_1	649	35	33	35	33	27	35
$(g_1g_1)_h$	1297	47	83	46	41	35	44
$(g_1g_1)_v$	1297	85	82	44	41	35	43
$(g_1g_2)_h$	1297	84	82	84	79	68	83
$(g_1g_2)_v$	1297	85	82	84	80	68	83
$(g_1e_1)_h$	1297	78	78	80	75	61	79
$(g_1e_1)_v$	1297	79	77	79	74	61	78

TAB. 6.5 – *Tailles des images g1, g2, et e1 dans différents formats : en première colonne, le format est sans compression. Les colonnes suivantes fournissent la taille de ces images selon le format dans lequel elles sont enregistrées. Pour ces mêmes formats, les tailles des images concaténées sont également fournies et mettent en évidence, le cas échéant, la propriété de normalité (cas surlignés en vert).*

Ceci signifie que, pour tous les algorithmes de compression d'images, seul le format PNG est capable de repérer la répétition d'une ligne, mais seulement si celle-ci est répétée immédiatement. Par contre, le format PNG n'est pas capable de repérer une seconde occurrence de cette ligne à un endroit quelconque du fichier. D'une certaine manière, la "mémoire" de l'algorithme n'est efficace qu'à court terme.

Cependant, il ne suffit pas de tester la propriété de normalité : il reste encore

à savoir si la compression relative est pertinente.

A titre indicatif, nous avons calculé la distance de similarité entre les images g_1 et g_2 d'une part, et g_1 et e_1 d'autre part, à l'aide de la formule 6.3 et en utilisant les résultats des quatre algorithmes de compression de données binaires du tableau précédent. Les résultats obtenus sont résumés dans le tableau 6.6 :

algorithme	PPMZ	UHARC-3	UHARC-x	UHARC-z
$d_K(g_1, g_2)$	0,05	0,07	0,03	0,07
$d_K(g_1, e_1)$	-0,05	-0,02	0,03	-0,02

TAB. 6.6 – *Distances de similarité entre les images g_1 et g_2, et entre les images g_1 et e_1. Les calculs sont effectués à l'aide de quatre algorithmes de compression de données binaires sans pertes vérifiant la propriété de normalité.*

Nous pouvons constater que, d'une part la formule appliquée fournit des résultats inférieurs à 0, d'autre part la distance entre deux images similaires est très proche de la distance obtenue entre deux images différentes. Rappelons que l'image g_2 est l'image qui suit immédiatement l'image g_1 dans la séquence-personne dont elles sont extraites.

Les résultats obtenus sont assez décevants dans la mesure où on s'attend à ce que la distance entre g_1 et e_1 soit être bien plus importante que la distance entre g_1 et g_2.

6.4.4.2 Mise en oeuvre en une dimension

Dans une image personne, la majorité des pixels sont noirs car ils correspondent à l'arrière-plan de la scène qui a été supprimé lors des pré-traitements. Les pixels représentant véritablement la personne représentent une faible proportion des images. Or nous voulons n'exploiter que les informations utiles.

Notre seconde approche consiste à parcourir une image pour la convertir en une donnée monodimensionnelle tout en ne prenant en compte que les pixels-personnes.

On parcourt l'image ligne par ligne et tous les pixels qui ne sont pas noirs constituent ce que nous définissons par une *donnée-personne*. Une donnée-personne est monodimensionnelle, elle est constituée d'une suite de pixels.

Remarquons que si l'on désire que cette transformation soit réversible, on peut définir un format d'image constituée d'une part de cette suite de pixels, d'autre part des informations nécessaires à la reconstitution de l'image : les coordonnées de chaque début de ligne et le nombre de pixels qui forment cette ligne. Cette seconde partie de l'information n'interviendra pas dans la comparaison par compression relative car elle est peu pertinente.

Puisque l'image numérique est représentée en couleur, il s'agit de prendre en compte les différents canaux. Reprenons l'exemple de l'espace de représentation (r, g) : à partir d'une image-personne, on peut constituer une première donnée-personne en ne tenant compte que des composantes r, puis une seconde donnée-personne en ne tenant compte que des composantes g des pixels. Enfin, on peut également constituer une troisième donnée personne en alternant les pixels r et g.

La comparaison de deux données-personnes à l'aide de la distance de similarité s'effectue en concaténant simplement les deux données-personnes, puis en compressant les données à l'aide d'un compresseur normal comme PPMZ ou UHARC, qui ne sont pas nécessairement adaptés aux données bidimensionnelles.

6.4.4.3 Critiques

Nous ne fournirons pas d'exemple de mesure de similarité obtenue par cette seconde approche car les résultats ne se sont pas montrés plus satisfaisants que lors de l'approche à deux dimensions.

Il semble que, dans les deux cas monodimensionnel et bidimensionnel, les algorithmes de compression disponibles à l'heure actuelle sont de plus en plus performants lorsqu'il s'agit de compresser, mais ne semblent pas assez adaptés à la technique de compression relative pour ce type de données. L'utilisation de cette méthode, à l'heure actuelle, ne peut pas être menée avec succès.

En outre, il existe une limite inhérente à l'implémentation des algorithmes : la taille du dictionnaire, lorsque le compresseur en utilise un, est fixe. C'est-à-dire que, face à la grande diversité des données présentes dans une seule image, le risque est d'avoir rempli le dictionnaire avant même la fin de la première moitié de la donnée résultant de la concaténation des images requête et candidate.

Nous avons vu à quel point les résultats sont décevants, même avec deux images qui se ressemblent très fortement comme deux images successives d'une

séquence-personne.

Enfin, même si les résultats obtenus lors de la comparaison par compression relative étaient satisfaisants, nous ferions face à un problème de mise en oeuvre : en effet cette technique de compression relative nécessite la compression de chaque image-requête, de chaque image-candidate, mais surtout de la concaténation de ces deux images lors de chaque comparaison.

Ceci reviendrait à stocker ou transmettre un grand nombre de données, ainsi qu'à réaliser un grand nombre de compression de fichiers résultant de la concaténation de deux images.

6.5 Conclusion

L'étude des différentes théories de l'information et de la complexité de description, le lien entre complexité et compression de données, ainsi que la richesse des algorithmes de compression qui existent aujourd'hui, nous fournissent un nouveau cadre de réflexion.

D'autres notions issues de la théorie de la complexité n'ont pas été abordées mais sont également applicables à la description et à la comparaison de données. Charles Bennett introduit par exemple la profondeur logique d'un objet comme étant le temps de calcul nécessaire à la production de cet objet sur une machine de Turing universelle à partir de sa description minimale au sens de la complexité de Kolmogorov. La quantité d'information est cette fois le nombre de pas de calcul nécessaires à la production de δ, à partir du plus court programme générant δ. Par exemple, la suite des décimales de π (le premier millier, par exemple) possède une faible complexité de Kolmogorov puisqu'il existe des algorithmes très courts permettant de les générer. En revanche sa profondeur logique sera importante car pour engendrer ces chiffres il est nécessaire d'exécuter un grand nombre de calculs.

Nous avons ensuite vu que la compression relative permet d'établir une mesure de similarité entre deux données binaires. Malheureusement, cette technique semble relativement peu pertinente, à l'heure actuelle, dans le cadre de la comparaison de deux images numériques couleur.

Cependant, cet échec n'est pas définitif et il n'y a aucune raison pour que le principe sur lequel repose la compression relative ne s'applique pas aux images.

Un algorithme de compression d'images sans pertes qui serait capable d'exploiter pleinement l'aspect bidimensionnel des données serait certainement adapté à la compresion relative. La compression fractale constitue peut-être une piste intéressante, à moins qu'un tel algorithme n'ait pas encore été développé. En effet, les efforts déployés dans le domaine de la compression d'images ne sont pas nécessairement orientés vers une application à la compression relative.

Toutefois, même si la compression relative s'avérait pertinente, la mise en oeuvre ne serait pas appropriée dans le cadre de notre travail en raison de la quantité de données qu'il faudrait compresser lors de chaque comparaison de deux images ou de deux séquences-personnes.

Cependant, les essais successifs que nous avons réalisés pour mettre en oeuvre la comparaison par compression relative, notamment en employant différents compresseurs, nous ont permis de remarquer certaines propriétés intéressantes.

Nous nous sommes donc appliqués à chercher une autre méthode basée sur la complexité de description. L'objectif est non seulement d'éviter la compression relative qui nécessite la concaténation des deux données, mais aussi d'obtenir une méthode de comparaison plus efficace.

Chapitre 7

Signatures d'images par indices de complexité

7.1 Introduction

Dans le chapitre précédent, nous avons constaté que l'utilisation de la notion de complexité relative ne nous permet pas d'obtenir de résultats satisfaisants. De plus, même si c'était le cas, la mise en oeuvre par compression relative comporte un inconvénient majeur dans notre cas : la nécessité de concaténer deux données afin de les comparer. Ceci nécessite, d'une part de disposer d'une donnée requête dans son intégralité lorsque l'on désire la comparer à une donnée candidate, d'autre part d'effectuer une nouvelle opération de compression lors de chaque nouvelle comparaison.

Cependant, l'idée de fond de la complexité de description est intrinsèquement juste. Dans ce chapitre, nous proposons d'exploiter la théorie algorithmique de l'information sous une forme originale sans employer la notion de complexité relative. Notre objectif est de montrer qu'elle peut, lorsqu'elle est employée à l'aide d'outils adaptés, fournir des résultats au moins comparables à ceux que fournissent certaines méthodes statistiques dans le cadre de la comparaison d'images ou de séquences d'images à partir de signatures.

Le premier paragraphe introduit le principe sur lequel repose nos travaux. Nous essayons d'établir un parallèle entre, d'une part les indices statistiques calculés à partir de vecteurs d'attributs d'images, et d'autre part la complexité de description d'une image, relative à certains programmes particuliers.

Dans le deuxième paragraphe, nous définissons les notions d'indices de complexité qui peuvent être vus comme un équivalent algorithmique des indices statistiques de texture (le mot "équivalent" étant à interpréter au sens large). Ceci nous permet de proposer une méthode appelée "multicompression" qui nous permet de définir un nouveau vecteur d'attributs d'images dont les attributs sont des taux de compression. Nous expliquons en quoi cette méthode permet la création de signatures compactes et robustes comparables aux signatures constituées d'indices statistiques.

Enfin, dans le troisième paragraphe, nous appliquons la méthode de comparaison par "multicompression" à la comparaison de séquences-personnes à travers l'exemple qui a été utilisé dans le chapitre 6.

7.2 Argument : un parallèle entre indices statistiques et taux de compression

7.2.1 Rappels sur les indices de texture

Dans le troisième chapitre, nous avons découvert différentes méthodes visant à caractériser des images ou des séquences d'images, en vue de les comparer. Ces méthodes sont basées sur la notion de signatures constituées de vecteurs d'attributs statistiques d'images.

Dans un premier temps nous avons présenté des vecteurs d'attributs formés à partir de statistiques calculées sur une image couleur pour constituer sa signature. Un histogramme couleur normalisé est un vecteur d'attributs qui décrit le contenu d'une image en fonction de la probabilité d'apparition des composantes colorimétriques de ses pixels, tandis qu'une matrice de co-occurrences couleur décrit la connexité entre les pixels de cette image et nous renseignent sur la texture représentée dans cette image. Les longueurs de plage constituent un troisième exemple de vecteur d'attributs.

Nous avons également vu que d'autres attributs peuvent être calculés à partir des matrices de co-occurrences ou des longueurs de plages : les indices de texture qui caractérisent les propriétés de texture. Chaque indice de texture est une grandeur scalaire qui caractérise le contenu d'une image selon un critère qui lui est propre, un vecteur d'indices de texture constitue ainsi une signature assez

robuste et surtout très compacte dans le cadre de la comparaison de séquences. Ainsi, l'indice d'homogénéité est d'autant plus élevé que l'on retrouve souvent les mêmes couples de pixels, tandis que l'entropie caractérise le désordre que peut présenter une texture.

Nous avons ainsi vu, dans le chapitre 5, que l'utilisation de vecteurs d'indices permet de créer une signature encore plus compacte dont les résultats sont assez satisfaisants.

7.2.2 Un taux de compression est un indice de complexité

L'indice statistique le plus simple qui permettrait de caractériser une image numérique couleur serait l'entropie calculée à partir de son histogramme couleur. Il ne caractériserait que les probabilités d'apparition des couleurs, et nous ne l'avons d'ailleurs pas rencontré dans la littérature. Cependant, nous pouvons remarquer qu'il s'agit bien de la notion d'entropie statistique définie par Shannon, et qui permet de quantifier l'information présente dans une donnée à l'aide d'une valeur scalaire et selon un critère défini : plus celle-ci est importante, plus le "taux de désordre/complexité" de la donnée est élevé.

D'une certaine manière, les indices statistiques calculés à partir d'un vecteur d'attribut décrivant une image nous renseignent sur le degré de redondance dans les données présentes dans cette image, selon certains critères qui dépendent des vecteurs d'attributs employés. Ainsi, un indice statistique quantifie certaines répétitions qu'un vecteur d'attributs est capable de détecter. Selon le vecteur d'attributs utilisé, les répétitions repérées ne sont pas les mêmes.

D'autre part, nous savons que la complexité de description d'une image peut être mesurée par la longueur du plus petit programme qui permet de générer cette image. L'utilisation de compresseurs de données sans pertes fournit un moyen de calculer une approximation de cette complexité de description.

Un compresseur de données vise à repérer les répétitions présentes dans une image afin de la représenter sous une forme plus compacte en éliminant les informations redondantes donc inutiles. Il n'existe aucun compresseur universel capable de repérer toutes les formes de répétitions, c'est pourquoi nous avons vu plusieurs algorithmes de compression différents. Chaque algorithme étant capable de repérer certaines redondances, nous pouvons considérer qu'en pratique, la complexité de description d'une donnée dépend du compresseur employé.

Nous pouvons donc considérer que le taux de compression $K_\lambda[I]$ d'une image I associé au compresseur λ représente un attribut d'image caractérisant cette image selon un critère propre à ce compresseur λ.

L'interprétation de cette valeur scalaire unique qui caractérise le contenu de l'image peut être rapprochée de l'interprétation d'un indice statistique qui caractérise également le contenu de cette image par une grandeur numérique scalaire.

7.2.3 Exemples d'indices de complexité

Nous pouvons par exemple remarquer que l'entropie d'une image en niveaux de gris, calculée à partir de son histogramme, est à mettre en correspondance avec le taux de compression de cette même image lorsque l'algorithme de Huffmann est utilisé. D'une certaine manière, ces deux valeurs caractérisent la distribution des probabilités d'apparition de motifs élémentaires : les pixels.

Un second parallèle peut être établi entre les indices calculés sur les longueurs de plages (isosegments ou *run-length*) et ce que permet de mesurer la compression RLE (*run-length encoding*) : dans les deux cas il s'agit de caractériser une image à partir des répétitions de motifs identiques (les pixels) : la mesure des longueurs de plages décrit des répétitions, tandis que ces répétitions sont exploitées par l'algorithme de compression.

Enfin, un troisième exemple concerne les répétitions d'arrangements spatiaux de motifs élémentaires. Les compresseurs à dictionnaires recherchent ces motifs élémentaires et éliminent leurs redondances, tandis que les co-occurrences ou les corrélogrammes les caractérisent.

7.2.4 Corrélations entre indices de textures et de complexité

Les figures 7.1 et 7.2 constituent un bon exemple du lien qui existe entre certains indices de textures calculés à partir des matrices de co-occurrences des images d'une séquence et les taux de compression de ces images, obtenus avec certains algorithmes de compression.

Chacune des courbes représente la trajectoire des points-images d'une séquence-personne. L'abscisse de chaque point i représente l'indice de complexité du canal

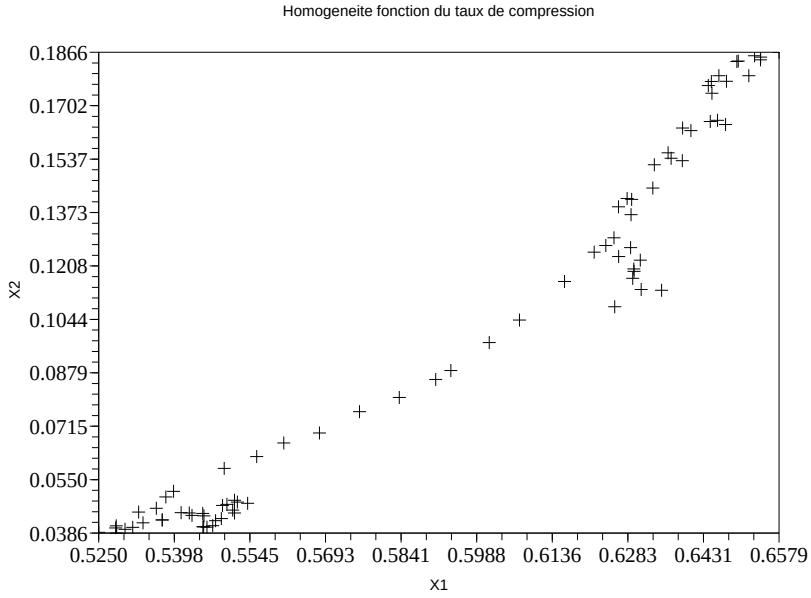


FIG. 7.1 – *Relation entre l'indice d'homogénéité d'une texture et le taux de compression obtenu par compression arithmétique. Les abscisses des points représentent les taux de compression obtenus avec un algorithme de compression arithmétique appliqué au canal r des données-personnes de la séquence $\mathcal{E}^{(2)}$. Les ordonnées des points représentent l'homogénéité des matrices de co-occurrences M_{rr} des images-personnes correspondantes.*

r de l'image-personne i et son ordonnée représente un indice de texture calculé à partir de la matrice de co-occurrences $M_{rr}[i]$. Pour la première courbe, l'indice de texture utilisé est l'homogénéité et pour la seconde courbe, il s'agit de l'entropie.

On constate que, pour la personne $\mathcal{E}^{(2)}$ et le canal r, ces deux indices de texture sont fortement corrélés avec le taux de compression obtenu par l'algorithme de compresssion arithmétique.

Cet exemple est toutefois exceptionnel : en pratique, de telles corrélations sont très rares.

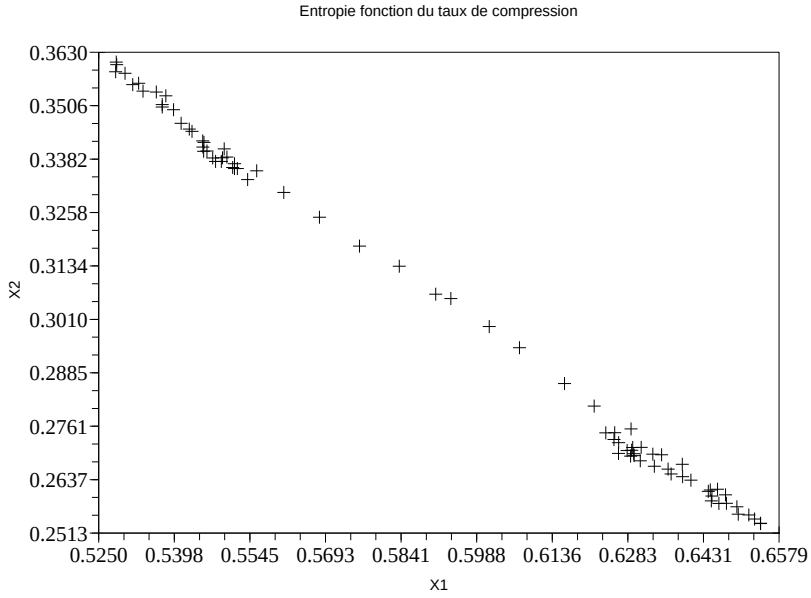


FIG. 7.2 – *Relation entre l'indice d'entropie d'une texture et le taux de compression obtenu par compression arithmétique. Les abscisses des points représentent les taux de compression obtenus avec un algorithme de compression arithmétique appliqué au canal r des données-personnes de la séquence* $\mathcal{E}^{(2)}$. *Les ordonnées des points représentent l'entropie des matrices de co-occurrences* M_{rr} *des images-personnes correspondantes.*

7.3 Signatures d'images par multicompression

7.3.1 Compresser = comprendre

La complexité de Kolmogorov est une mesure théorique car elle est basée sur l'universalité du langage employé : on ne peut donc pas déterminer le plus court programme capable d'engendrer une donnée δ.

Toutefois il est possible de mesurer la longueur du plus petit programme capable de produire cette donnée δ en utilisant un langage donné, non nécessairement universel. Ainsi, chaque compresseur détermine, pour une donnée, le plus petit programme à exécuter dans le langage propre à ce compresseur.

On peut alors considérer que chaque compresseur repère certaines régularités, certains permettant par exemple de mesurer indirectement certains indices proches de l'entropie, de l'homogénéité ou encore de l'uniformité d'une texture.

Il est donc possible de considérer chaque compresseur comme une "boîte noire", sans se soucier véritablement de la structure sous-jacente qu'il caractérise. Pour ce faire, il faudrait examiner les sources du programme pour en découvrir l'algorithme.

De plus, certains algorithmes résultent de l'évolution et de l'expertise du domaine de la compression. D'une certaine manière, la population des algorithmes a évolué par "sélection technologique". Certains compresseurs permettent donc de mesurer aisément des structures qui seraient très compliquées à mettre en oeuvre si l'on voulait les décrire à l'aide de vecteurs d'attributs.

Enfin, notons que deux implémentations différentes d'un même algorithme peuvent donner deux résultats différents.

Nous pouvons ainsi considérer que l'ensemble des taux de compression mesurés lorsque l'on applique plusieurs algorithmes de compression sans pertes sur une même image permettent de caractériser celle-ci de la même manière qu'avec plusieurs indices de texture caclulés à partir de vecteurs d'attributs.

Il est donc possible que l'ensemble de plusieurs valeurs différentes de complexité de description d'une même donnée possède un certain pouvoir discriminant dans le cas de la comparaison. Afin de le savoir, nous allons donc appliquer ce principe à la comparaison de signatures de séquences-personnes.

7.3.2 Signatures de données binaires par vecteur d'indices de complexités

Dans notre démarche, nous supposons dans un premier temps que si deux données δ et γ ont des contenus similaires, il y a une forte probabilité pour que leurs taux de compressions respectifs soient proches, à condition bien entendu que le compresseur employé pour mesurer le taux de compression soit le même.

Ici encore la réciproque est fausse : deux taux de compression identiques ne signifie pas nécessairement que les données se ressemblent. Cependant cela constitue un *indice* (au sens littéral) et qu'il y a *une certaine* probabilité que ces données se ressemblent.

Si l'on multiplie le nombre de compresseurs, et si deux données ont des taux de compression proches pour chaque compresseur, la probabilité d'une similarité devient importante.

Définition 10 (Indices de complexité) *Un indice de complexité K_λ est calculé à partir de la compression d'un fichier à l'aide d'un algorithme de compression sans pertes λ. Il représente le ratio entre la longueur, en bits, du fichier compressé et la longueur, en bits, du fichier non compressé.*

Il ne s'agit pas du taux de compression car, selon les données et le compresseur employé, il arrive que la taille du fichier compressé soit supérieure à celle du fichier d'origine : le compresseur est tout à fait inefficace puisqu'il augmente la taille des données, et il est alors possible d'obtenir des taux de compression négatifs. Un indice de complexité, en revanche, est toujours positif.

.

Définition 11 (Vecteur d'indices de complexités) *Un vecteur d'indices de complexité $\mathbf{x}_K[I]$ d'une image I contient n_K indices de complexité. Il est obtenu à partir de la compression du fichier contenant la représentation binaire de la données-personne issue de I, effectuée à l'aide de n_K algorithmes de compression sans pertes.*

7.3.3 Interprétation et critiques

Nous proposons de comparer des séquences-personnes en utilisant des vecteurs d'indices de complexité caractérisant chacun une image des séquences-personnes, de la même manière que nous l'avons fait dans les chapitres précédents, c'est-à-dire en employant les deux méthodes de comparaison de séquences de vecteurs d'attributs. Cette démarche est similaire au cas où les vecteurs d'attributs employés étaient des indices de texture calculés à partir des matrices de co-occurrences caractérisant ces mêmes images.

Nous précisons qu'il s'agit d'une démarche expérimentale. En effet elle consiste à observer et mesurer des données, sans jamais savoir exactement ce qui est mesuré, sauf dans ces cas parfaitement identifiés comme celui de la compression de Shannon/Huffmann.

Cependant, contrairement aux indices statistiques qui sont corrélés, nous cherchons à employer des compresseurs dont les résultats sont indépendants.

La quantité de valeurs scalaires nécessaires pour représenter un vecteur d'indices de complexités est comparable à celle des vecteurs d' indices statistiques : ici la taille d'un vecteur d'indices de complexité est proportionnelle au nombre

de compresseurs employés (une dizaine).

Un avantage dans la mise en oeuvre est que le résultat est obtenu de manière très simple, contrairement aux indices statistiques dont le calcul nécessite la création des vecteurs d'attributs. Ici, il suffit presque d'appliquer un algorithme de compression sur une image-personne et de lire la taille du fichier compressé. Nous verrons cependant qu'une rapide phase de pré-traitements est nécessaire, et contribue à augmenter considérablement la rapidité des traitements. Nous avons décidé de travailler à partir des données-personnes introduites dans le chapitre précédent.

De plus, les performances atteintes par les algorithmes sont souvent très correctes : leur optimisation résulte de plusieurs années d'expérience des chercheurs et des développeurs. Ces algorithmes permettent ainsi de remplacer de manière astucieuse des calculs qui seraient assez délicats à développer si l'on désirait rechercher des structures complexes parfaitement identifiées.

Il reste cependant encore à trouver la bonne combinaison d'algorithmes, en étudiant les résultats obtenus avec chacun d'entre eux de manière indépendante, et en déterminant les corrélations entre les différents résultats. Comme nous avons employé onze algorithmes de compression, certains sont probablement redondants.

7.4 Comparaison de deux ensembles de vecteurs d'indices de complexité

De manière expérimentale, nous avons vérifié sur notre base de séquences que certains compresseurs permettent effectivement d'apparier deux séquences dont les contenus sont similaires, et de distinguer deux séquences dont les contenus se ressemblent peu.

Nous présentons tout d'abord la manière dont nous avons constitué la signature d'une séquence-personne grâce aux vecteurs d'attributs composés de taux de compression.

Nous appliquons ensuite les deux méthodes de comparaison de séquences de vecteurs d'attributs que nous avons vues dans le cinquième chapitre et utilisées

jusque là, aux deux cas de l'exemple de référence, avant de fournir les scores de pertinence atteints par chacune de ces deux méthodes.

7.4.1 Constitution des séquences de vecteurs d'indices de complexités

7.4.1.1 Pré-traitement des images-personnes

A partir d'une séquence-personne représentée dans l'espace couleur (r, g), nous constituons trois séquences de données-personnes de la même manière que dans le chapitre précédent : nous parcourons chaque image ligne par ligne et retenons uniquement les pixels-personnes dont nous retranscrivons les valeurs dans un fichier appelé donnée-personne, soit en ne retenant qu'un canal de couleur, soit en alternant les deux canaux.

A chaque image de la séquence correspondent trois fichiers texte, chacun contenant les lettres de l'alphabet allant de 'A' à 'O' correspondant aux valeurs des composantes couleur de 0 à 15 : nous codons les données sur les 4 bits de poids fort. Par exemple, la donnée-personne d_r constituée à partir d'une image représentée dans le plan-couleur (r, g) contient les valeurs des composantes r des pixels-personnes, tandis que la donnée-personne d_{rg} contient l'alternance des valeurs des composantes r et g des pixels-personne.

Ainsi à la séquence-personne $\mathcal{G}^{(1)}$ correspondent les trois séquences de données-personnes $D_r[\mathcal{G}^{(1)}]$, $D_g[\mathcal{G}^{(1)}]$, $D_{rg}[\mathcal{G}^{(1)}]$.

Notons que nous pouvons constituer une donnée personne en conservant le nombre de niveaux sur lesquels sont échantillonnés les canaux, soit 256, ou encore réduire le nombre de niveaux à ou 64. Il suffit simplement de n'utiliser que 16 ou 64 symboles en codant les données sur 4 ou 6 bits au lieu des 256 symboles disponibles pour chaque octet écrit.

Les ressources sont gérées par les programmes et ceux-ci sont parfaitement optimisés. Dans la mesure où les algorithmes sont compatibles avec la compression de données binaires, il est possible d'effectuer les mesures sur ces trois valeurs possibles des niveaux (256, 64 et 16). Les ressources nécessaires à la compression des données-personnes sont sensiblement les mêmes dans les trois cas de quantification puisqu'il s'agit de compresser directement un fichier de taille proportionnelle

au nombre de pixels-personnes, quel que soit le nombre de niveaux utilisés.

Par contre, nous avons constaté que les ordres de grandeur des indices de complexité sont naturellement très différents selon le nombre de niveaux employés. Ceci est parfaitement compréhensible puisque si une donnée-personne est constituée en échantillonnant les valeurs sur 16 niveaux seulement, elle comporte beaucoup plus de répétitions que sur 64 ou 256 niveaux ce qui sera plus avantageux pour la discrimination.

7.4.1.2 Multicompression

A partir d'une séquence de données-personnes, nous calculons les composantes des vecteurs de complexité en compressant chaque fichier texte contenant une donnée-personne à l'aide de plusieurs algorithmes. Les vecteurs d'indices que nous avons employés pour caractériser une donnée-personne sont au nombre de onze. Chaque indice de complexité est calculé à partir du résultat de l'application d'un algorithme de compression de données binaire sur cette donnée-personne, en calculant le taux de compression obtenu.

Toutefois, le nombre de logiciels de compression est plus restreint puisque deux d'entre eux fournissent plusieurs algorithmes différents.

Dans le tableau 7.4.1.2, nous décrivons les onze indices ainsi que le détail des algorithmes employés pour les constituer.

Tous ces programmes sont libres et disponibles sur internet. On peut les trouver aux adresses suivantes, certains étant accompagnés de publications :

- UHARC : http ://www.klaimsoft.com/winuha/
- PPMZ2 : http ://www.cbloom.com/src/ppmz.html
- Coder : http ://sourceforge.net/projects/compressions
- acfile : http ://www.cipr.rpi.edu/ said/FastAC.html
- ac : http ://www.boden.de

Bien que certains le permettent en fournissant le code source, nous ne nous sommes pas intéressés à l'implémentation de chaque algorithme. Au contraire, nous utilisons chacun d'eux comme simple "instrument de mesure" fournissant un algorithme de compression, sans chercher à en saisir le détail de son implémentation.

Après cette étape de multicompression, nous obtenons une séquence de vec-

teurs d'indices de complexité pour chaque séquence de données-personnes. Puisque nous travaillons sur des images représentées dans un plan chromatique, nous disposons de trois séquences de vecteurs d'indices de complexités de taille $v_K = 11$:

$$\begin{array}{l} \{\mathbf{x}_{K_r}[I_i]\},\ i = 1 \ldots n \\ \{\mathbf{x}_{K_{rg}}[I_i]\},\ i = 1 \ldots n \\ \{\mathbf{x}_{K_g}[I_i]\},\ i = 1 \ldots n \end{array} \tag{7.1}$$

7.4.2 Comparaison par descripteurs

La signature d'une séquence-personne S est alors composée des 9 vecteurs d'attributs suivants :

$$\Psi_{L_{(r,g)}}[S] = \begin{pmatrix} \mathbf{min}_{K_r}[S] & \mathbf{med}_{K_r}[S] & \mathbf{max}_{K_r}[S] \\ \mathbf{min}_{K_{rg}}[S] & \mathbf{med}_{K_{rg}}[S] & \mathbf{max}_{K_{rg}}[S] \\ \mathbf{min}_{K_g}[S] & \mathbf{med}_{K_g}[S] & \mathbf{max}_{K_g}[S] \end{pmatrix} \tag{7.2}$$

Pour comparer les signatures requête et candidate, nous calculons ici encore la moyenne des proximités entre les descripteurs de vecteurs d'attributs homologues en employant la norme L_1.

index	nom de l'indice de complexité	nom de l'algorithme	nom du logiciel
1	PPM-Z	PPM	PPMZ2
2	UHARC-3	ALZ (qualité optimale)	UHARC
3	UHARC-X	PPM	UHARC
4	UHARC-Z	LZP	UHARC
5	Coder-H	Huffmann	Coder
6	Coder-A	Arithmétique	Coder
7	Coder-7	LZ77	Coder
8	Coder-8	LZ78	Coder
9	Coder-W	LZW	Coder
10	Acfile	Arithmétique	Acfile
11	Ac	Arithmétique	Ac

TAB. 7.1 – *Les onze algorithmes de compression employés pour constituer un vecteur d'indices de complexité.*

7.4.2.1 Résultats

Le tableau suivant fournit les mesures de proximités obtenues pour l'exemple du chapitre 5 : nous comparons la séquence requête $\mathcal{F}^{(1)}$ aux trois séquences candidates $\mathcal{F}^{(2)}$, $\mathcal{D}^{(1)}$ et $\mathcal{B}^{(1)}$ dans chacun des plans chromatiques :

candidate	(r,g)	(H,S)	(l_1,l_2)
$\mathcal{F}^{(2)}$	0.060	0.066	0.065
$\mathcal{B}^{(1)}$	0.392	0.217	0.155
$\mathcal{D}^{(1)}$	0.081	0.111	0.207

TAB. 7.2 – *Distances mesurées par comparaison de descripteurs de séquences de vecteurs d'indices de complexité entre la séquence-requête $\mathcal{F}^{(1)}$ et les trois séquences candidates $\mathcal{F}^{(2)}$, $\mathcal{D}^{(1)}$ et $\mathcal{B}^{(1)}$, en fonction du plan chromatique dans lequel sont représentées les séquences d'images.*

Comme lors de la comparaison à partir d'indices de textures, nous pouvons constater que la séquence-personne candidate $\mathcal{D}^{(1)}$ est toujours plus éloignée de la requête que la séquence-personne $\mathcal{F}^{(2)}$, bien que l'individu $\mathcal{D}$ ressemble fortement à l'individu $\mathcal{F}$. Nous observons même que les distances mesurées dans le plan-couleur (l_1,l_2) sont très discriminantes.

(r,g)	(H,S)	(l_1,l_2)
0.172	0	0

TAB. 7.3 – *Scores de pertinences de la comparaison par descripteurs de séquences de vecteurs d'indices de complexité appliquée à la base de données de séquences-personnes, en fonction du plan chromatique dans lequel sont représentées les séquences d'images.*

Le résultat du calcul des scores de pertinence est fourni dans le tableau 7.4.2.1. Nous constatons que, en dehors de l'espace chromatique (r,g) la comparaison de signatures par descripteurs de séquences de vecteurs d'indices de complexité fournit des scores parfaits, c'est-à-dire qu'aucune erreur n'est commise lors de la discrimination des séquences-personnes de la base. Cette signature est encore plus pertinente que la signature par descripteur de séquences de vecteurs d'indices de texture.

De toutes les méthodes que nous avons employées lors de nos travaux, c'est celle qui aura fourni les meilleurs scores de pertinence. Ceci est d'autant plus remarquable qu'il ne s'agit que d'une première tentative d'utiliser ce type de signature. Certes, les vecteurs d'indices de complexité contiennent davantage de composantes qu'un vecteur d'indices de texture, la confrontation de ces deux méthodes n'est donc pas tout à fait équitable. Mais dans la mesure où la multicompression reste encore très perfectible, nous pouvons espérer obtenir à l'avenir des résultats encore meilleurs. A la fin de ce chapitre, nous reviendrons plus en détails sur les différentes perspectives que nous proposons.

7.4.2.2 Ressources

Comme nous sommes dans le cas d'une signature par descripteurs, La taille de la signature est fixe et ne dépend que du nombre de compresseurs employés :

$$W(\Psi_K[S_{Req}]) = W(\Psi_K[S_{Can}]) = 3 \times 3 \times v_K$$

avec $v_K = 11$.

Le temps nécessaire pour créer une signature par descripteurs, à partir des trois séquences de vecteurs d'indices de complexité, est fourni par l'équation :

$$T(\Psi_K^F) = v_K \times n_{can} \times log(n_{can})$$

En pratique, nous avons constaté que ce temps de calcul n'excédait pas une à deux millisecondes.

Enfin, l'essentiel du temps de calcul concerne la création des séquences de vecteurs d'indices de complexité. Il est possible de créer les données-personnes au fur-et-à-mesure de l'acquisition. Cependant, le temps nécessaire à la multicompression est conséquent et celle-ci ne peut être réalisée en ligne. C'est pourquoi la signature de séquences-personnes par multicompression est essentiellement adaptée à la création des matrices origine-destination.

Nous avons estimé, sur un exemple, le temps de calcul nécessaire à l'application de chacun des onze algorithmes, toujours sur la machine équipée d'un processeur cadencé à 1.4 GHz.

Le premier programme, PPMZ-2, nécessite un temps de calcul prohibitif puisque, pour compresser une seule donnée-personne, le temps mesuré s'élève à près d'une

seconde. Cependant le temps d'exécution mesuré est celui du programme complet, or celui-ci vérifie son résultat en décompressant le fichier qu'il vient de compresser.

Le second programme le plus coûteux en temps est celui qui correspond à l'algorithme LZ77, celui-ci met 400 ms pour compresser une donnée personne.

Les trois programmes UHARC ainsi que celui qui implémente l'algorithme LZW mettent entre 100 et 200 ms.

Enfin, les compression par Huffman, LZ78 et arithmétique sont les plus rapides puisqu'ils ne nécessitent "que" environ 50 à 60 ms chacun.

Nous constatons que la mise en oeuvre en temps réel de la multicompression demandera une sélection drastique des programmes, et que leur optimisation sera nécessaire.

7.5 Explication proposée concernant l'efficacité des résultats

Compte tenu de la pertinence des résultats obtenus à partir d'une séquence de signatures d'images par multicompression, lorsque l'on adopte l'approche de comparaison de deux séquences de vecteurs d'attributs par descripteurs, nous pouvons nous demander pourquoi la technique de compression relative fournit des résultats si peu satisfaisants.

Nous proposons ici une explication qui s'articule autour de trois points.

7.5.1 Données symboliques

Le premier point est que les données sur lesquelles sont appliquées généralement les techniques de compression relative sont, dans chacun des cas, des données symboliques.

La comparaison par compression relative de textes, de musique ainsi que de séquences génétiques consiste à comparer ces données symboliques. Cette technique s'avère efficace en raison de la propriété de normalité des algorithmes de compression qui sont employés : si deux données symboliques sont "presque" identiques, c'est que de nombreuses parties communes à ces deux données sont

rigoureusement identiques. Ce sont ces redondances que l'algorithme de compression va exploiter.

Deux données sont "presque" identiques, si par exemple l'on change une lettre d'un mot pour une autre lettre (une faute de frappe), une note d'une mélodie (une fausse note) ou un gène dans une séquence génétique (une mutation). Dans les trois cas, nous comprenons bien que, si seul un symbole diffère, la ressemblance entre les deux données reste importante.

7.5.2 Erreurs de recopie

Le second point à remarquer découle du premier : supposons que l'on cherche à recopier le contenu d'une donnée δ vers une donnée δ', mais que l'on commette une erreur de recopie. Puisque les données sont symboliques, il peut s'agir d'une faute de frappe lors de la recopie d'un mot, d'une fausse note lors de la transcription d'une partition, ou encore une mutation génétique.

L'erreur commise entraine une différence dans l'interprétation du message mais l'amplitude de cette différence n'est pas proportionnelle à la différence entre le symbole qui a été remplacé et le nouveau symbole.

Par exemple, que l'on remplace un 'a' par un 'b' ou un 'z', que l'on remplace un 'do' par un 'ré' ou un 'sol', ou que l'on remplace 'A' par 'C', 'G' ou 'T' dans l'écriture d'un gène, le résultat est le même : l'interprétation de la donnée acquiert simplement un sens différent.

Ceci est naturel puisque les symboles ne représentent pas des quantités. (ce qui n'est qu'en partie valable pour les notes, mais suffisamment dans le cadre de cette argumentation).

7.5.3 Données quantitatives

Les données images sont de nature complètement différentes : chaque symbole représente l'intensité de la composante couleur d'un pixel. La modification d'un symbole (par exemple la valeur 0=noir d'un pixel) pour un nouveau symbole a des répercussions différentes selon que le nouveau symbole vaut 1=gris très foncé, ou s'il vaut 255=blanc. Du point de vue de la compression relative, si l'on compare une image entièrement noire à une image entièrement blanche on obtiendra la même mesure de proximité que si on compare une image noire à une image

gris foncé.

Chaque "petite variation" entre les composantes couleur présentes dans une image I et une image J fait que l'algorithme de compression aura des difficultés à trouver dans I et dans J de grandes plages de données rigoureusement identiques, même si I et J sont deux images consécutives d'une même séquence.

Ceci est particulièrement le cas pour des images numériques où les "petites variations" sont très fréquentes, même si elles ne sont que de faible importance en termes quantitatifs. L'algorithme se contente finalement d'exploiter de nombreuses petites répétitions qui sont présentes dans les deux images.

La compression de la concaténation de I et de J, comme nous l'avons vu dans le chapitre précédent, permet plutôt de constater que deux images se ressemblent si elles ont la même complexité au sens du compresseur employé : on a alors souvent $K(I) + K(J) = K(IJ)$, que I et J soient similaires ou différentes.

C'est la raison pour laquelle la compression relative s'est avérée inutile : il est plus simple et apparemment plus efficace de comparer deux images en comparant directement leurs complexités respectives en rendant cette comparaison plus robuste grâce à plusieurs compresseurs. On se contente simplement de vérifier que $K_\lambda(I) = K_\lambda(J)$ pour $j = 1 \dots v_K$ compresseurs.

7.6 Perspectives

Pour finir, nous proposons quelques perspectives de travaux qui n'ont pas été effectués.

7.6.1 Constitution des données-personnes

7.6.1.1 Codage des valeurs

Le premier point concerne la manière de constituer les données-personnes à partir d'une image-personne. Nous avons déjà mentionné le fait que ces données sont codées sur 16 niveaux seulement mais peuvent l'être sur davantage de niveaux. Puisque les algorithmes employés servent à compresser des données binaires, ceux-ci doivent fonctionner "quel que soit" le nombre de niveaux et les ressources nécessaires ne sont pas beaucoup plus importantes. Il faut tenir compte du fait que les algorithmes sont faits pour traiter au maximum 256 symboles dif-

férents. Toutefois, des essais effectués en codant les données-personnes sur 256 niveaux ne se sont pas montrés satisfaisants. Cependant, il est possible de coder les données sur 32, 64 ou 128 niveaux.

7.6.1.2 Parcours des images-personnes

Toujours au sujet de la constitution des données-personnes, le second point concerne la manière de parcourir les images-personnes. En effet, dans notre démarche, nous parcourons les images ligne par ligne : nous ne respectons donc pas véritablement l'aspect bidimensionnel des images. Les seules répétitions que repèrent les algorithmes de compression correspondent à une connexité horizontale entre les pixels, et de surcroit dans un seul sens. Ceci n'a pas eu d'incidence dans notre cas puisque les individus filmés se déplacent toujours dans le même sens, cependant nous savons qu'il ne s'agit que d'un cas particulier.

Une autre manière plus pertinente de parcourir les données est de suivre une courbe de la famille des *space-filling curves* [Sag94]. Ces courbes ont la particularité de remplir un plan sans jamais se couper. Parmi celles-ci on peut citer la courbe de Peano ou encore celle de Hilbert. La construction de cette dernière est illustrée sur la figure 7.3.

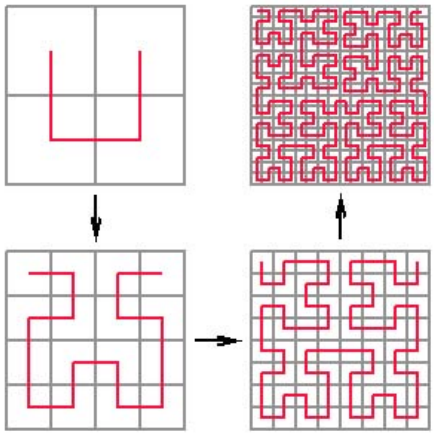


FIG. 7.3 – *Construction de la courbe de Hilbert*

En parcourant les données de cette manière, nous permettons aux algorithmes de compression de repérer des répétitions correspondant aux connexités entre pixels dans toutes les directions possibles.

Dans ce cas, l'essentiel du travail concerne évidemment le développement de l'algorithme de parcours des images-personnes.

7.6.2 Multicompression

L'aspect "multicompression" constitue sans doute la partie la plus intéressante des perspectives.

7.6.2.1 Amélioration

Dans un premier temps, nous avons utilisé onze algorithmes, de manière assez arbitraire. Il serait intéressant d'étudier l'influence de chacun d'eux afin de savoir si certains sont redondants par rapport à d'autres. Il est également possible d'introduire une pondération pour chaque indice de complexité : en effet certains sont peut-être plus pertinents que d'autres.

7.6.2.2 Enrichissement du vecteur de compresseurs

Parallèlement au point précédent, nous pouvons certainement enrichir le vecteur d'indices de complexités en employant d'autres algorithmes. Il existe en effet une grande quantité d'algorithmes de compression, et autant d'implémentations différentes.

7.6.3 Optimisation des performances

En opposition à ce point de vue théorique, l'aspect pratique est tout aussi important et concerne l'amélioration des performances. En effet, nous avons vu que le temps nécessaire à certains algorithmes comme PPMZ ou LZ77 pour compresser une seule donnée-personne ne permet pas une application pratique. Cependant, pour certains programmes de compression, nous avons constaté que le temps d'exécution du programme ne reflétait pas le véritable temps de calcul de l'algorithme de compression proprement dit.

L'analyse et l'adaptation du code source de ces programmes permettrait certainement, dans un premier temps, d'optimiser les performances. On peut, par exemple, remplacer l'écriture dans le fichier de destination par la création d'une structure dont on évalue simplement la taille occupée en mémoire.

7.7 Conclusion

Nous avons d'abord vu qu'il est possible d'établir une certaine mise en correspondance entre l'interprétation d'un indice de texture calculé à partir de la

matrice de co-occurrences d'une image I et l'interprétation du taux de compression de cette même image : celui-ci permet de mesurer certaines propriétés de l'image selon un critère propre au compresseur, relatif à la nature des répétitions que celui-ci est capable de repérer.

Nous avons également vu d'une part que selon l'algorithme de compression employé, le taux de compression d'une image est différent et d'autre part que les différences observées entre les taux de compression obtenus varient selon les images. Nous avons alors proposé d'employer plusieurs algorithmes de compression pour caractériser une image sous la forme d'un vecteur d'indices de complexité.

A partir de ces nouveaux vecteurs d'attributs d'images, nous avons appliqué la méthode de comparaison de séquences de signatures par descripteurs et nous avons constaté que les scores de pertinence atteints par cette méthode sont très satisfaisants.

Bien que cette signature par multicompression soit beaucoup plus compacte qu'une signature formée à partir des matrices de co-occurrences, celle-ci nous a permis de distinguer deux individus différents qui n'étaient pas distingués avec la signature par matrices de co-occurrences normalisées.

Conclusion et perspectives

Ce mémoire présente notre contribution à la comparaison d'objets déformables dans des séquences d'images couleur basée sur l'analyse des composantes couleur des pixels représentant ces objets.

Dans le chapitre 1 nous avons exposé les deux problématiques qui seront abordées dans le cadre de la vidéosurveillance et dans celui de l'aide à l'exploitation de sites de transports publics. Nous avons vu que, dans ces deux cas, les objets déformables considérés représentent des personnes filmées selon une vue de dessus lors de leur déplacement dans le champ d'observation de caméras placées dans les sites de transports publics.

La problématique est donc de retrouver les mêmes personnes dans différentes séquences d'images couleur saisies par des caméras observant ces différentes zones.

Dans le chapitre 2, nous avons proposé un cahier des charges répondant aux besoins des exploitants. Nous avons montré la nécessité de développer une méthode permettant de comparer le contenu d'une séquence d'images appelée requête et représentant le passage d'une personne, à une autre séquence d'images appelée candidate, qui représente également le passage d'une personne.

Afin de déterminer s'il s'agit de la même personne ou de deux personnes différentes, la comparaison nécessite la définition de signatures de la personne en mouvement basées sur les propriétés spatio-colorimétriques des pixels la représentant dans les différentes images d'une même séquence.

Les caractéristiques colorimétriques d'un objet représenté par une image sont rassemblées au sein d'une signature couleur. Dans le chapitre 3, nous avons présenté différents espace de représentation de la couleur, puis nous avons passé en revue un certain nombre de signatures d'images composées de un ou plusieurs vecteur d'attributs. Nous avons vu que la ressemblance entre deux signatures

s'évalue en mesurant la similarité entre les vecteurs d'attributs homologues.

La signature que nous avons retenue est composée de matrices de co-occurrences qui permettent de caractériser le contenu d'une image en termes de couleurs mais aussi de textures.

Dans le chapitre 4, nous avons exposé différentes méthodes de comparaison du contenu de deux séquences d'images. Après avoir mis en évidence les problèmes causés par le caractère temporel inhérent aux séquences, nous avons présenté et analysé différentes signatures de séquences d'images rencontrées dans la littérature ainsi que les mesures de proximité qui leur sont appliquées.

Notre intérêt s'est porté sur la signature par descripteurs qui permet de résumer l'information contenue dans une séquence de signatures d'images.

Nous avons ensuite proposé dans le chapitre 5 une première signature de séquence d'images par descripteur appliqué à une séquence de signatures d'images lorsque celles-ci se composent des matrices de co-occurrences chromatiques vues au chapitre 3. Des tests expérimentaux effectués avec des séquences d'images de personnes acquises en laboratoire ont fourni des résultats satisfaisants en termes de reconnaissance de personnes. Pour ce faire, le calcul d'un score de pertinence a permi d'évaluer la pertinence de la méthode de comparaison proposée. Compte tenu de du faible coût en termes de temps de calcul nécessaire à la constitution d'une telle signature, celle-ci semble particulièrement bien adaptée à la localisation de personne dans le cadre de la vidéosurveillance.

Nous avons ensuite appliqué cette méthode à deux séquences de signatures lorsque celles-ci se composent d'un autre type de vecteurs d'attributs calculés à partir des matrices de co-occurrences : les vecteurs d'indices de textures. Les tests expérimentaux ont fourni des résultats sensiblement meilleurs que ceux obtenus en comparant deux signatures par descripteurs constitués à partir des matrices de co-occurrences chromatiques. Le coût en termes de ressources de temps de calcul nécessaire pour constituer cette seconde signature est plus important que celui qui est nécessaire pour constituer la première signature formée directement à partir des matrices de co-occurrences. Cependant, les ressources en quantité de cellules mémoire pour stocker cette seconde signature sont bien plus faibles et la comparaison de telles signatures est bien plus rapide que dans le cas de la première signature. C'est pourquoi cette signature par descripteurs de séquences de vecteurs d'indices de textures semble davantage adaptée au cas de la constitution de matrices origine-destination dans le cadre de l'aide à l'exploitation et

la régulation de trafic.

Nous nous sommes ensuite appliqués à intégrer la prise en compte de l'information couleur par des méthodes issues de la théorie algorithmique de l'information.

Dans le chapitre 6, nous avons présenté la théorie algorithmique de l'information. Nous avons défini la notion de complexité de description de Kolmogorov qui s'appuie sur le principe de description de longueur minimale, puis nous avons exposé les liens entre la complexité de description et la compression de données.

Nous avons ensuite abordé l'application de cette théorie à la comparaison de données. La distance de transformation a d'abord été définie du point de vue théorique, puis nous avons présenté une manière de s'en approcher par compression relative.

Enfin, nous avons mis en évidence les limites de cette méthode lorsque celle-ci est appliquées à deux séquences-personnes.

Dans le chapitre 7, nous avons présenté nos travaux sur la comparaison de signatures de séquences d'images basées sur un vecteur d'attributs d'image original : le "vecteur d'indices de complexités".

L'hypothèse sous-jacente à cette démarche est que l'application d'algorithmes de compression réglés avec les mêmes paramètres à deux séquences similaires fournit des résultats très proches en termes de niveau de compression tandis que l'application à deux séquences différentes fournit des mesures différentes.

Après avoir mis en évidence les liens entre indices de complexités et indices de textures, nous avons défini la notion de vecteur d'indices de complexités obtenus par multicompression. Enfin, pour comparer deux séquences-personnes, nous avons appliqué aux deux séquences de vecteurs d'indices de complexités la méthodes de comparaison de deux séquences de vecteurs d'attributs que nous avons retenue au chapitre 5.

Les résultats expérimentaux obtenus en appliquant cette troisième signature à la base de données de séquences-personnes ont fourni des scores de pertinence très satisfaisants. Cependant, les ressources nécessaires à la constitution de séquences de vecteurs d'indices de complexité sont considérables en termes de temps de calcul ; il n'est donc pas envisageable d'exploiter cette signature dans le cadre de la localisation de personne. En revanche, compte tenu de la faible quantité de ressources mémoire nécessaires pour le stockage de cette signature, ainsi que le nombre peu important d'opérations élémentaires employées pour comparer de telles signatures, leur utilisation peut être envisagée pour la constitution de ma-

trices origine-destination.

Ce premier essai d'application de la théorie de la complexité, différente de la technique de compression relative, ouvre de nouvelles perspectives. Cependant il nous reste encore à explorer plus en détails cette nouvelle orientation.

L'exploitation de la notion de complexité de description associée à celle de compresseur est une approche qui mérite d'être approfondie, notamment en ce qui concerne le choix des algorithmes de compression utilisés, ainsi que l'importance à accorder à chacun d'eux.

Annexe A

Calcul des scores de pertinence

Afin d'évaluer la pertinence des résultats des différentes méthodes de comparaison de séquences d'images, nous calculons un score de pertinence à partir des résultats obtenus en comparant chaque séquence de la base avec toutes les autres séquences.

Tableau de classement

Nous disposons de $N = 8 \times 4 = 32$ séquences. Pour chaque séquence requête d'index i_{Req} avec i_{Req} variant de 1 à N, nous classons les $N-1$ autres séquences candidates d'index i_{Can}, avec $i_{Can} \neq i_{Req}$ dans l'ordre des ressemblances décroissantes à la séquence requête i_{Req}, et regroupons les résultats dans un tableau de classements.

Calcul du score de pertinence

Nous calculons ensuite, toujours pour chaque séquence requête, la somme $S_{i_{Req}}$ des classements des trois séquences candidates représentant la même personne que cette séquence requête.

Ainsi, si ces trois séquences candidates sont les trois premières, la somme est minimale et vaut $S_{min} = 1 + 2 + 3 = 6$.

Si les séquences requêtes sont les trois dernières, la somme est maximale et vaut alors $S_{max} = (N-1)+(N-2)+(N-3)$ puisqu'il y a 31 séquences candidates possibles. Pour une valeur de $N = 32$ on a $S_{max} = 31 + 30 + 29 = 90$.

L'évaluation du score de pertinence est donnée par :

$$Score = \frac{1}{N}\sum_{i=1}^{N}(\frac{S_i}{S_{min}} - 1)$$

avec ici $N = 32$ et $S_{min} = 6$.

Interprétation

Un score nul est obtenu si aucune erreur n'est commise, c'est-à-dire si pour chaque séquence requête, les trois séquences les plus ressemblantes sont celles qui représentent le même individu. (On a alors $S_i = S_{min}$ pour tout i, donc $\frac{S_i}{S_{min}} - 1 = 0$.

La méthode de comparaison est d'autant moins pertinente que le score est élevé. Pour notre base de 32 personnes, le score maximal atteint pour chaque requête correspondant au pire des cas est de 90. Si celui-ci est atteint pour les 32 requêtes possibles, le pire score que puisse atteindre une méthode sur cette base est donc de

$$Score_{max} = \frac{1}{32}\sum_{i=1}^{32}(\frac{90}{6} - 1)$$

soit

$$Score_{max} = \frac{1}{32}\sum_{i=1}^{32}(15 - 1)$$

d'où une valeur de $Score_{max} = 14$.

Bibliographie

[AGN86] G.R. Arce, N.C. Gallagher, and T.A. Nodes. Median filters : Theory for one or two dimensional filters. In *Advances in Computer Vision and Signal Processing*. JAI Press, 1986.

[BGL+98] C. Bennett, P. Gacs, M. Li, P. Vitanyi, and W. Zurek. Information distance. *IEEE Transactions on Information Theory*, 44(4) :1407–1423, 1998.

[BW84] J.B. Bednar and T.L. Watt. Alpha-trimmed means and their relationship to median filters. *IEEE Trans. Acoust. Speech Signal Processing*, ASSP-32(1) :145–153, Fevrier 1984.

[Cab92] F. Cabestaing. *Détection de contours en mouvement dans une séquence d'images. Conception et réalisation d'un processeur câblé temps réel*. PhD thesis, Université des Sciences et Technologies de Lille, 1992.

[Car95] T. Carron. *Segmentation d'images couleur dans la base Luminance-Teinte-Saturation, approches numériques et symboliques*. PhD thesis, Université de Savoie, 1995.

[CC01] Y.K. Chan and C.C. Chang. Image matching using run-length feature. *Pattern Recognition Letters*, 22(5) :447–455, 2001.

[CC04] Y.K. Chan and C.Y. Chen. Image retrieval system based on color-complexity and color-spatial features. *Journal of Systems and Software*, 71(1-2) :65–70, 2004.

[CH80] R.W. Conners and C.A. Harlow. Toward a structural textural analyser based on statistical methods. *Computer Graphics and Image Processing*, 12 :224–256, 1980.

[Chu36] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58 :345–363, 1936.

[CI02] J. Calic and E. Izquierdo. Efficient key-frame extraction and video analysis. In *Proceedings of the IEEE International Conference on Information Technology*, Coding and Computing (ITCC'02), pages 28–33, Avril 2002.

[CLK00] R. T. Collins, A. J. Lipton, and T. Kanade. Introduction to the special section on video surveillance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8) :745–746, August 2000.

[CN00] P. Campisi and A. Neri. Synthetic summaries of video sequences using a multiresolution based key frame selection technique in a perceptually uniform color space. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2 of *ICIP'00*, pages 299–302, Vancouver, Canada, Septembre 2000.

[col04] Ouvrage collectif. *Encyclopaedia Universalis*. CD-ROM, 2004.

[CP95] J.P Cocquerez and S. Philipp. *Analyse d'images : filtrage et segmentation*. Masson, 1995.

[CSL99] H.S. Chang, S. Sull, and S.U. Lee. Efficient video indexing scheme for content-based retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8) :1269–1279, Decembre 1999.

[CV05] R. Cilibrasi and P.M.B. Vitanyi. Clustering by compression. *IEEE Trans. Information Theory*, 51(4) :1523–1545, 2005.

[CVDW04] R. Cilibrasi, P. Vitanyi, and R. De Wolf. Algorithmic clustering of music based on string compression. *Computer Music Journal*, 28(4) :49–67, 2004.

[CZ00] S.C. Cheung and A. Zakhor. Efficient video similarity measurement and search. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 85–88, Vancouver, Canada, 10–13 Septembre 2000.

[DA00] M.S. Drew and J. Au. Video keyframe production by efficient clustering of compressed chromaticity signatures (poster session). In *Association for Computing Machinery Bulletin*, pages 365–367, Los Angeles, Californie (USA), Octobre 2000.

[DBC+05] X. Desurmont, A. Bastide, C. Chaudy, C. Parisot, J. F. Delaigle, and B. Macq. Image analysis architectures and techniques for intelligent surveillance systems. *IEE Proceedings on Vision, Image and Signal Processing*, 152(2) :224–231, April 2005.

[DDAK99] N.D. Doulamis, A. D. Doulamis, A. D. Avrithis, and S. D. Kollias. A stochastic framework for optimal key frame extraction from mpeg video databases. *Computer Vision and Image Understanding*, 75 :3–24, Juillet/Août 1999.

[Del94] J.P. Delahaye. *Information, Complexité et hasard.* Hermès, Paris, 1994.

[DKD98] J. F. DeMenthon, V. Kobla, and D. Doermann. Video summarization by curve simplification. In *ACM Multimedia 98*, pages 211–218, Bristol, United Kingdom, Septembre 1998.

[DMK+01] Y. Deng, B. S. Manjunath, C. Kenney, M.S. Moore, and H. Shin. An efficient color representation for image retrieval. *IEEE Transactions on Image Processing*, 10(1) :140–147, Janvier 2001.

[DVD96] J.P. Deparis, S. Velastin, and A.C. Davies. Project cromatica. In *Proceedings of the 5th International Conference on Automated People Movers (APM 96)*, pages 227–239, Paris, France, June 1996.

[DWL98a] M.S. Drew, J. Wei, and Z.N. Li. Illumination-invariant color object recognition via compressed chromaticity histograms of color-channel-normalized images. In *Proceedings of the International Conference on Computer Vision*, pages 533–540, Bombay, India, Janvier 1998.

[DWL98b] M.S. Drew, J. Wei, and Z.N. Li. On illumination invariance in color object recognition. *Pattern Recognition*, 31(8) :1077–1087, August 1998.

[Fab01] R. Fablet. *Modélisation statistique non paramétrique et reconnaissance du mouvement dans des séquences d'images ; application à l'indexation vidéo.* PhD thesis, Université de Rennes I, 2001.

[Fan61] M. Fano. *Transmission of Information : A Statistical Theory of Communications.* MIT Press, Cambridge, Mass., 1961.

[Fen96] P. M. Fenwick. The burrows-wheeler transform for block sorting text compression. *Computer Jrnl.*, 39(9) :731–740, 1996.

[FT97] A.M. Ferman and A.M. Tekalp. Multiscale content extraction and representation for video indexing. In *Proceedings of the SPIE Conference on Multimedia Storage and Archiving Systems*, volume 2, pages 23–31, Dallas, Texas, 3–4 Novembre 1997.

[FTM02] A.M Ferman, A.M. Tekalp, and R. Mehrotra. Robust color histogram descriptors for video segment retrieval and identification. *IEEE Transactions on Image Processing*, 11(5) :497–508, Mai 2002.

[FTMK00] A. M. Ferman, A.M. Tekalp, R. Mehrotra, and S. Krishnamachari. Group-of-frame/picture color histogram descriptors for multimedia applications. In *Proceedings of the IEEE International Conference on Image Processing*, pages 65–68, Vancouver (Canada), Septembre 2000.

[GB00] A. Girgenson and J. Boreczky. Time-constrained keyframe selection technique. *Multimedia Tools and Applications*, 12(3) :347–358, Aout 2000.

[GFT97] B. Günsel, Y. Fu, and A.M. Tekalp. Hierarchical temporal video segmentation and content characterization. In *Proceedings of the SPIE Conference on Multimedia Storage and Archiving Systems*, volume 2, pages 46–56, Dallas, Texas, 3-4 Novembre 1997.

[GS99a] T. Gevers and A. W. M. Smeulders. Color-based object recognition. *Pattern Recognition*, 32 :453–464, 1999.

[GS99b] T. Gevers and A. W. M. Smeulders. Content-based image retrieval by viewpoint-invariant color indexing. *Image and Vision Computing*, 17 :475–488, 1999.

[GS99c] T. Gevers and A.W.M. Smeulders. The pictoseek WWW image search system. In *Proceedings of the International Conference on Multimedia Communications Systems*, volume 1, pages 264–269, Florence (Italy), Juin 1999.

[GS996] *A comparative study of several color models for color image invariants retrieval*, Aout 1996.

[Har79] R.M. Haralick. Statistical and structural approaches to texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 67(5) :786–804, Mai 1979.

[HJ99] A. Hanjalic and H. Jiang. An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. *IEEE Transactions on circuits and systems for video technology*, 9(8) :1280–1288, Decembre 1999.

[HKa01] C. Heath, L. Khoudour, and al. Report on requirements for project tools and processes (deliverable 4). Technical report, PRISMATICA Project, September 2001.

[HM00a] R. Hammoud and R. Mohr. Gaussian mixture densities for indexing of localized objects in a video sequence. Technical report, INRIA, March 2000. http ://www.inria.fr/RRRT/RR-3905.html.

[HM00b] R. Hammoud and R. Mohr. Mixture densities for video objects recognition. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 2, pages 71–75, Septembre 2000.

[HM00c] R. Hammoud and R. Mohr. A probabilistic framework of selecting effective key frames for video browsing and indexing. In *Proceedings of the International workshop on Real-Time Image Sequence Analysis*, Oulu, Finland, Septembre 2000.

[HM00d] R. Hammoud and R. Mohr. Probabilistic hierarchical framework for clustering of tracked objects in video streams. In *Proceedings of the Irish Machine Vision and Image Processing Conference*, pages 133–140, Queen's University of Belfast, Northern Ireland, Août 2000.

[HS81] B.K.P. Horn and B.G. Schunk. Determining optical flow. *Artificial Intelligence*, 17 :185–203, 1981.

[Huf52] D.A. Huffman. A method for the construction of minimum redundancy codes. *Proceedings of the Institute of Electronics and Radio Engineers*, 40 :1098–1101, 1952.

[Inc03] E. Incerti. *Compression d'Images*. Vuibert, Paris, 2003.

[Ka99] L. Khoudour and al. Détection de chutes sur les voies et d'intrusion en tunnels dans les transports publics. *Recherche, Transports et Sécurité (RTS)*, 62 :92–104, January 1999.

[KAM00] S. Krishnamachari and M. Abdel-Mottaleb. Compact color descriptor for fast image and video segment retrieval. In *Proceedings of the IS&T/SPIE Conference on Storage and Retrieval for Media Databases*, San Jose (California), Janvier 2000.

[KDB00] L. Khoudour, D. Dooze, and J.L. Bruyelle. *Passenger and equipment safety in railway transport*, chapter 6. WIT Press, 2000. ISBN : 1 85312 7841 ISSN 1462-608X.

[Kho03] L. Khoudour. Local camera network for surveillance in public transport. In *Proceedings of the Technological Innovation for Land Trans-*

portation Congress (TILT 2003), pages 347–354, Lille, France, December 2003.

[Kol65] A.N. Kolmogorov. Three approaches to the definition of the concept of quantity of information (en russe). *Problemy Peredachi Informastrii*, 1 :3–11, 1965.

[Kow90] P. Kowaliski. *Vision et mesure de la couleur*. Masson, 1990.

[La96] A. Langlais and al. User needs analysis (deliverable 3). Technical report, CROMATICA Project, September 1996.

[LCL+04] M. Li, X. Chen, X. Li, B. Ma, and P.M.B. Vitanyi. The similarity metric. *IEEE Trans. Information Theory*, 50(12) :3250–3264, 2004.

[LKMP04] T. Leclercq, L. Khoudour, L. Macaire, and J.G. Postaire. Compact color video signature by principal component analysis. In *Proceedings of the International Conference on Computer Vision and Graphics (ICCVG 2004)*, pages 814–819, Warsaw, Poland, septembre 2004.

[LNZS01] T. Lin, C.W. Ngo, H.J. Zhang, and Q.Y. Shi. Integrating color and spatial features for content-based video retrieval. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 592–595, Thessaloniki, Greece, Octobre 2001.

[LPE97] R. Lienhart, S. Pfeiffer, and W. Effelsberg. Video abstracting. *Communications of ACM*, 40 :55–62, Decembre 1997.

[LSV03] B.P.L. Lo, J. Sun, and S.A. Velastin. Fusing visual and audio information in a distributed intelligent surveillance system for public transport systems. *ACTA Automatica Sinica : Special Issue on Visual Surveillance*, 3(29) :393–407, May 2003.

[LV97] M. Li and P. Vitanyi. *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag, New-York, 2nd edition, 1997.

[LZ00] T. Lin and H.J. Zhang. Automatic video scene extraction by shot grouping. In *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 4039–4042, Barcelona (Spain), Septembre 2000.

[LZFS02] T. Lin, H. Zhang, J. Feng, and Q. Shi. Shot content analysis for video retrieval applications. *Journal of Software*, 13(8) :1577–1585, 2002.

[MMKP02] D. Muselet, L. Macaire, L. Khoudour, and J-G. Postaire. Color invariant for person images indexing. In *Proceedings of the first European*

Conference on Colour in Graphics, Image and Vision (CGIV'02), pages 236–240, Poitiers (France), Avril 2002.

[MPE99] MPEG-7 Output Document ISO/MPEG. *MPEG-7 Visual part of eXperimentation Model Version 2.0*, December 1999.

[MS05] G.L. Micheloni, C.and Foresti and L. Snidaro. A network of cooperative cameras for visual surveillance. *IEE Proceedings on Vision, Image, and Signal Processing*, 152 :205–212, April 2005.

[MT00] S. J. Maybank and T. N. Tan. Introduction – surveillance. *International Journal of Computer Vision*, 37(2) :173–173, June 2000.

[MT04] S. J. Maybank and T. N. Tan. Special issue on visual surveillance. *Image and Vision Computing*, 22(7) :Page iii, July 2004.

[Mus05] D. Muselet. *Reconnaissance automatique d'objets sous éclairage non contrôlé par analyse d'images couleur.* PhD thesis, Université des Sciences et Technologies de Lille, 2005.

[MZ98] W.Y. Ma and H.J. Zhang. *Content based image indexing and retrieval.* CRC Press, 1998.

[NP92] R. Nelson and R. Polana. Qualitative recognition of motion using temporal texture. *Computer Vision, Graphics, and Image Processing*, 1(56) :78–99, 1992.

[Ple03] R. Pless. Image spaces and video trajectories : Using isomap to explore video sequences. In *Proceedings of the International Conference on Computer Vision*, pages 1433–1440, Nice (France), 13–16 Octobre 2003.

[PN93] R. Polana and R. Nelson. Detecting activities. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 2–7, New-York (USA), Juin 1993.

[Pol94] R. Polana. *Temporal texture and activity recognition.* PhD thesis, University of Rochester, 1994.

[Pra03] J. D. Prange. Detecting, recognizing and understanding video events in surveillance video. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2003)*, page 4, Miami, Florida, USA, 2003.

[PRT99] *Empirical evaluation of dissimilarity measures for color and texture*, volume 2, Septembre 1999.

[PYL99] I. Park, I. Yun, and S Lee. Color image retrieval using hybrid graph representation. *Image and Vision Computing*, 17 :465–474, 1999.

[PZ99] G. Pass and R. Zabih. Comparing images using joint histograms. *ACM Journal of Multimedia Systems*, 7(3) :234–240, 1999.

[PZM96] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. *ACM Journal of Multimedia Systems*, pages 65–73, 1996.

[RD02] M. Rautiainen and D.S. Doermann. Temporal color correlograms for video retrieval. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 1, pages 267–270, Quebec, Canada, 11-15 Aout 2002.

[RDR04] J. Rey-Debove and A. Rey. *Le nouveau petit Robert*. Dictionnaires Le Robert, 2004.

[RHC97] Y. Rui, T.S. Huang, and S.F. Chang. Image retrieval : Past, present, and future. In *Proceedings of the International Symposium on Multimedia Information Processing*, Taiwan, Décembre 1997.

[Ris78] J. Rissanen. Modeling by shortest data description. *Automatica*, 4 :465–471, 1978.

[RJPR02] P. Remagnino, G. A. Jones, N. Paragios, and C. S. Regazzoni. *Video-Based Surveillance Systems : Computer Vision and Distributed Processing*. Kluwer Academic Publishers, November 2002. ISBN/ISSN 0-7923-7632-3.

[Sag94] H. Sagan. *Space Filling Curves*. Springer-Verlag, 1994.

[Sal04] D. Salomon. *Data Compression (3rd Ed.)*. Springer, 2004.

[Sap90] G. Saporta. *Probabilités, analyse de données et statistiques*. Technip, 1990.

[SB91a] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 1(7) :11–32, 1991.

[SB91b] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 1(7) :11–32, 1991.

[SC96] J.R. Smith and S.F. Chang. Tools and techniques for color image retrieval. In *Storage and retrieval for image and video databases (SPIE)*, pages 426–437, San Diego, California, 28 Janvier–2 Fevrier 1996.

[Sha48] C.E. Shannon. Information theory. *IEEE Transactions on Information Theory*, 1948.

[SMP00] M. Skrzypniak, L. Macaire, and J.G. Postaire. Indexation d'images de personnes par analyse de matrices de co-occurrences couleur. In *Journées d'études et d'échanges Compression et Représentation des Signaux Audiovisuels (CORESA'00)*, pages 411–418, Poitiers (France), Octobre 2000.

[SS94] M. Stricker and M.J. Swain. The capacity of color histogram indexing. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 704–708, Seattle, Washington, Juin 1994.

[Sèv96] R. Sève. *Physique de la couleur. De l'apparence colorée à la technique colorimétrique.* Masson, 1996.

[SW49] C.E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, 1949.

[TAOS93] Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakate. Videomap and videospaceicon : Tools for anatomizing video content. In *Proceedings of the ACM INTERCHI'93*, Conference on Human Factors in Computing Systems, pages 131–141, Amsterdam, Mai 1993.

[Tro91] A. Trouvé. *La mesure de la couleur - Principes, techniques et produits du marché.* Editions CETIM - AFNOR, 1991.

[Tur36] A.M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42) :230–265, 1936.

[Tur50] A.M. Turing. Computing machinery and intelligence. *Mind*, 236 :433–460, 1950.

[UF99] S. Uchihachi and J. Foote. Summarizing video using a shot importance measure and a frame-packing algorithm. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 3041–3044, Phoenix, Arizona, Mars 1999.

[VAB+98] S. Velastin, D. Aubert, B. Boghossian, L. Khoudour, M.A. Vicencio-Silva, J.L. Bruyelle, M. Wherrett, D. Sorrenti, and J.H. Yin. Description of the developed processes. Technical report, CROMATICA project, January 1998.

[Van00] N. Vandenbroucke. *Segmentation d'images couleur par classification de pixels dans des espaces d'attributs colorimétriques adaptés. Application à l'analyse d'images de football.* PhD thesis, Université des Sciences et Technologies de Lille, 2000.

[VDR99] J.S. Varré, J.P. Delahaye, and E. Rivals. The transformation distance : a dissimilarity measure based on movements of segments. *Bioinformatics*, 15(3) :194–202, 1999.

[Vel05] S. A. Velastin. Editorial of special section on intelligent distributed surveillance systems. *IEE Proceedings on Vision Image and Signal Processing*, 152(2) :191, April 2005.

[Vie88] C. Vieren. *Segmentation de scènes dynamiques en temps réel - Application au traitement de séquences d'images pour la surveillance de carrefours routiers.* PhD thesis, Université des Sciences et Technologies de Lille, 1988.

[VK04] S. Velastin and L. Khoudour. Prismatica : A multi-sensor surveillance system for public transport networks. In *Proceedings of the 12th IEE Conference on Road Transport Information and control (RTIC 2004)*, pages 19–25, London, UK, April 2004.

[VKa02] S. Velastin, L. Khoudour, and al. Innovative tools for security in transport (deliverable 7). Technical report, PRISMATICA Project, March 2002.

[VL00] P.M.B. Vitanyi and M. Li. Minimum description length induction, bayesianism, and kolmogorov complexity. *IEEE Trans. Inform. Theory*, 46(2) :446–464, 2000.

[VLS03] S. Velastin, B. Lo, and J. Sun. An intelligent distributed surveillance system for public transport. In *Proceedings of the IEE Symposium on Intelligent Distributed Surveillance Systems (IDSS 2003)*, pages 10/1–10/5, London, UK, February 2003.

[VV05] M. Valera and S. A. Velastin. Intelligent distributed surveillance systems : a review. *IEE Proceedings on Vision Image and Signal Processing*, 152(2) :192–204, April 2005.

[VVS03] M. Valera, S. Velastin, and H. Simpson. An approach for designing a real-time intelligent distributed surveillance system. In *Proceedings of the IEE Symposium on Intelligent Distributed Surveillance Systems (IDSS 2003)*, pages 6/1–6/5, London, UK, February 2003.

[W.90] Enkelmann W. Obstacle detection by evaluation of optical flow fields. In *Proceedings of Conference on Pro-Art on Vision*, pages 146–155, Sophia Antipolis, France, April 1990.

[Wei99] G. Weiss. *Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 1999. ISBN 0-262-23203-0.

[Wel84] Terry A. Welch. A technique for high-performance data compression. *IEEE Computer*, 17(6) :8–19, 1984.

[WNC87] I. Witten, R. Neal, and J. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30 :520–541, 1987.

[WSS00] Weinberger, Seroussi, and Sapiro. The LOCO-I lossless image compression algorithm : Principles and standardization into JPEG-LS. *IEEETIP : IEEE Transactions on Image Processing*, 9, 2000.

[Yah03] I. Yahiaoui. *Construction automatique de résumés vidéos : proposition d'une méthode générique d'évaluation*. PhD thesis, Télécom Paris, 2003.

[YV95] J.H. Yin and A.C. Velastin, S.A. Davies. Image processing techniques for crowd density estimation using a reference image. In *Proceedings of the 2nd Asian Conference on Computer Vision (ACCV 1995)*, pages 489–498, Singapore, December 1995.

[ZL77] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23, 1977.

[ZQZ01] D. Zhang, W. Qi, and H.J. Zhang. A new shot boundary detection algorithm. *Lecture Notes in COmputer Science*, 2195 :63–70, 2001.

[ZRHM98] Y. Zhuang, Y. Rui, T.S. Huang, and S. Metrotra. Adaptative key-frame extraction using unsupervised clustering. In *Proceedings of the IEEE International Conference on Image Processing*, pages 866–870, Chicago, Illinois (USA), Octobre 1998.

[ZT02] J. Zhang and T. Tan. Brief review of invariant texture analysis methods. *Pattern Recognition*, 35(3) :735–747, March 2002.

TITRE en français

Contribution à la comparaison de séquences d'images couleur par outils statistiques et par outils de la theorie algorithmique de l'information

RÉSUMÉ en français

Notre travail concerne le développement d'outils d'aide à la gestion et à la sécurité des transports publics s'appuyant sur l'observation et la surveillance des sites à l'aide de caméras vidéo. dans ce cadre, nous évaluons des méthodes de comparaison automatique de séquences d'images, permettant d'établir si différentes séquences correspondent ou non à l'observation d'un même usager.
Nous proposons ensuite une signature de séquence d'images par descripteurs de séquences de matrices de co-occurrences chromatiques, puis une signature par descripteurs de séquences de vecteurs d'indices de textures.
Nous considérons enfin une approche basée sur la théorie algorithmique de l'information et definissons une signature par vecteurs d'indices de complexite inspirés de la complexite de description de Kolmogorov.

TITRE en anglais

Contribution to the comparison of color images sequences by statistical tools and algorithmic information theory

RÉSUMÉ en anglais

Our work consists in developing tools for assistance to the management and the safety of public transports based on the observation and the monitoring of the sites with video cameras.
We evaluate methods for the automatic comparison of color images sequences in order to determinate if different sequences represent the same person.
We propose then a signature of images sequence by descriptors of a sequence of chromatic cooccurrences matrix, and a signature by descriptors of a sequence of texture indices.
Finally, we consider an approach based on the algorithmic information theory and we define a signature by vector of complexity indices inspired by the kolmogorov complexity.

DISCIPLINE

Automatique et Informatique Industrielle