No d'ordre 1494

USTL

LIFL Laboratoire d'Informatique Fondamentale de Lille

Thèse

présentée à

L'Université des Sciences et Technologies de Lille

pour l'obtention du titre de

Docteur en Informatique

par

Thomas Alexandre

Manipulation de données multimédia dans la carte à micro-processeur : application à l'identification biométrique et comportementale.

Soutenue le 23 février 1995 devant le jury :

Jean-Paul Delahaye, Président.
Christian Carrez,
Doug Tygar, Rapporteurs.
Vincent Cordonnier,
Philippe Maes,
Pierre Paradinas, Examinateurs .

UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE
U.F.R. d'I.E.E.A. Bât M3. 59655 Villeneuve d'Ascq CEDEX
Tél. 20.43.47.24 Fax. 20.43.65.66

UNIVERSITE DES SCIENCES
ET TECHNOLOGIES DE LILLE

DOYENS HONORAIRES DE L'ANCIENNE FACULTE DES SCIENCES

M. H. LEFEBVRE, M. PARREAU

PROFESSEURS HONORAIRES DES ANCIENNES FACULTES DE DROIT ET SCIENCES ECONOMIQUES, DES SCIENCES ET DES LETTRES

MM. ARNOULT, BONTE, BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, DECUYPER, DEHEUVELS, DEHORS, DION, FAUVEL, FLEURY, GERMAIN, GLACET, GONTIER, KOURGANOFF, LAMOTTE, LASSERRE, LELONG, LHOMME, LIEBAERT, MARTINOT-LAGARDE, MAZET, MICHEL, PEREZ, ROIG, ROSEAU, ROUELLE, SCHILTZ, SAVARD, ZAMANSKI, Mes BEAUJEU, LELONG.

PROFESSEUR EMERITE

M. A. LEBRUN

ANCIENS PRESIDENTS DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

MM. M. PARREAU, J. LOMBARD, M. MIGEON, J. CORTOIS, A.DUBRULLE

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

M. P. LOUIS

PROFESSEURS - CLASSE EXCEPTIONNELLE

M. CHAMLEY Hervé	Géotechnique
M. CONSTANT Eugène	Electronique
M. ESCAIG Bertrand	Physique du solide
M. FOURET René	Physique du solide
M. GABILLARD Robert	Electronique
M. LABLACHE COMBIER Alain	Chimie
M. LOMBARD Jacques	Sociologie
M. MACKE Bruno	Physique moléculaire et rayonnements atmosphériques

M. MIGEON Michel	EUDIL
M. MONTREUIL Jean	Biochimie
M. PARREAU Michel	Analyse
M. TRIDOT Gabriel	Chimie appliquée

PROFESSEURS - 1ère CLASSE

M. BACCHUS Pierre	Astronomie
M. BIAYS Pierre	Géographie
M. BILLARD Jean	Physique du Solide
M. BOILLY Bénoni	Biologie
M. BONNELLE Jean Pierre	Chimie-Physique
M. BOSCQ Denis	Probabilités
M. BOUGHON Pierre	Algèbre
M. BOURIQUET Robert	Biologie Végétale
M. BRASSELET Jean Paul	Géométrie et topologie
M. BREZINSKI Claude	Analyse numérique
M. BRIDOUX Michel	Chimie Physique
M. BRUYELLE Pierre	Géographie
M. CARREZ Christian	Informatique
M. CELET Paul	Géologie générale
M. COEURE Gérard	Analyse
M. CORDONNIER Vincent	Informatique
M. CROSNIER Yves	Electronique
Mme DACHARRY Monique	Géographie
M. DAUCHET Max	Informatique
M. DEBOURSE Jean Pierre	Gestion des entreprises
M. DEBRABANT Pierre	Géologie appliquée
M. DECLERCQ Roger	Sciences de gestion
M. DEGAUQUE Pierre	Electronique
M. DESCHEPPER Joseph	Sciences de gestion
Mme DESSAUX Odile	Spectroscopie de la réactivité chimique
M. DHAINAUT André	Biologie animale
Mme DHAINAUT Nicole	Biologie animale
M. DJAFARI Rouhani	Physique
M. DORMARD Serge	Sciences Economiques
M. DOUKHAN Jean Claude	Physique du solide
M. DUBRULLE Alain	Spectroscopie hertzienne
M. DUPOUY Jean Paul	Biologie
M. DYMENT Arthur	Mécanique
M. FOCT Jacques Jacques	Métallurgie
M. FOUQUART Yves	Optique atmosphérique
M. FOURNET Bernard	Biochimie structurale
M. FRONTIER Serge	Ecologie numérique
M. GLORIEUX Pierre	Physique moléculaire et rayonnements atmosphériques
M. GOSSELIN Gabriel	Sociologie
M. GOUDMAND Pierre	Chimie-Physique
M. GRANELLE Jean Jacques	Sciences Economiques
M. GRUSON Laurent	Algèbre
M. GUILBAULT Pierre	Physiologie animale
M. GUILLAUME Jean	Microbiologie
M. HECTOR Joseph	Géométrie
M. HENRY Jean Pierre	Génie mécanique
M. HERMAN Maurice	Physique spatiale
M. LACOSTE Louis	Biologie Végétale
M. LANGRAND Claude	Probabilités et statistiques

M. LATTEUX Michel	Informatique
M. LAVEINE Jean Pierre	Paléontologie
Mme LECLERCQ Ginette	Catalyse
M. LEHMANN Daniel	Géométrie
Mme LENOBLE Jacqueline	Physique atomique et moléculaire
M. LEROY Jean Marie	Spectrochimie
M. LHENAFF René	Géographie
M. LHOMME Jean	Chimie organique biologique
M. LOUAGE Françis	Electronique
M. LOUCHEUX Claude	Chimie-Physique
M. LUCQUIN Michel	Chimie physique
M. MAILLET Pierre	Sciences Economiques
M. MAROUF Nadir	Sociologie
M. MICHEAU Pierre	Mécanique des fluides
M. PAQUET Jacques	Géologie générale
M. PASZKOWSKI Stéfan	Mathématiques
M. PETIT Francis	Chimie organique
M. PORCHET Maurice	Biologie animale
M. POUZET Pierre	Modélisation - calcul scientifique
M. POVY Lucien	Automatique
M. PROUVOST Jean	Minéralogie
M. RACZY Ladislas	Electronique
M. RAMAN Jean Pierre	Sciences de gestion
M. SALMER Georges	Electronique
M. SCHAMPS Joël	Spectroscopie moléculaire
Mme SCHWARZBACH Yvette	Géométrie
M. SEGUIER Guy	Electrotechnique
M. SIMON Michel	Sociologie
M. SLIWA Henri	Chimie organique
M. SOMME Jean	Géographie
Melle SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. THIEBAULT François	Sciences de la Terre
M. THOMAS Jean Claude	Géométrie - Topologie
M. THUMERELLE Pierre	Démographie - Géographie humaine
M. TILLIEU Jacques	Physique théorique
M. TOULOTTE Jean Marc	Automatique
M. TREANTON Jean René	Sociologie du travail
M. TURRELL Georges	Spectrochimie infrarouge et raman
M. VANEECLOO Nicolas	Sciences Economiques
M. VAST Pierre	Chimie inorganique
M. VERBERT André	Biochimie
M. VERNET Philippe	Génétique
M. VIDAL Pierre	Automatique
M. WALLART Françis	Spectrochimie infrarouge et raman
M. WEINSTEIN Olivier	Analyse économique de la recherche et développement
M. ZEYTOUNIAN Radyadour	Mécanique

PROFESSEURS - 2ème CLASSE

M. ABRAHAM Francis	
M. ALLAMANDO Etienne	Composants électroniques
M. ANDRIES Jean Claude	Biologie des organismes
M. ANTOINE Philippe	Analyse
M. BALL Steven	Génétique
M. BART André	Biologie animale
M. BASSERY Louis	Génie des procédés et réactions chimiques
Mme BATTIAU Yvonne	Géographie
M. BAUSIERE Robert	Systèmes électroniques
M. BEGUIN Paul	Mécanique
M. BELLET Jean	Physique atomique et moléculaire
M. BERNAGE Pascal	Physique atomique, moléculaire et du rayonnement
M. BERTHOUD Arnaud	Sciences Economiques
M. BERTRAND Hugues	Sciences Economiques
M. BERZIN Robert	Analyse
M. BISKUPSKI Gérard	Physique de l'état condensé et cristallographie
M. BKOUCHE Rudolphe	Algèbre
M. BODARD Marcel	Biologie végétale
M. BOHIN Jean Pierre	Biochimie métabolique et cellulaire
M. BOIS Pierre	Mécanique
M. BOISSIER Daniel	Génie civil
M. BOIVIN Jean Claude	Spectrochimie
M. BOUCHER Daniel	Physique
M. BOUQUELET Stéphane	Biologie appliquée aux enzymes
M. BOUQUIN Henri	Gestion
M. BROCARD Jacques	Chimie
Mme BROUSMICHE Claudine	Paléontologie
M. BUISINE Daniel	Mécanique
M. CAPURON Alfred	Biologie animale
M. CARRE François	Géographie humaine
M. CATTEAU Jean Pierre	Chimie organique
M. CAYATTE Jean Louis	Sciences Economiques
M. CHAPOTON Alain	Electronique
M. CHARET Pierre	Biochimie structurale
M. CHIVE Maurice	Composants électroniques optiques
M. COMYN Gérard	Informatique théorique
Mme CONSTANT Monique	Composants électroniques et optiques
M. COQUERY Jean Marie	Psychophysiologie
M. CORIAT Benjamin	Sciences Economiques
Mme CORSIN Paule	Paléontologie
M. CORTOIS Jean	Physique nucléaire et corpusculaire
M. COUTURIER Daniel	Chimie organique
M. CRAMPON Norbert	Tectolique géodynamique
M. CURGY Jean Jacques	Biologie
M. DANGOISSE Didier	Physique théorique
M. DE PARIS Jean Claude	Analyse
M. DECOSTER Didier	Composants électroniques et optiques
M. DEJAEGER Roger	Electrochimie et Cinétique
M. DELAHAYE Jean Paul	Informatique
M. DELORME Pierre	Physiologie animale
M. DELORME Robert	Sciences Economiques
M. DEMUNTER Paul	Sociologie
Mme DEMUYNCK Claire	Physique atomique, moléculaire et du rayonnement
M. DENEL Jacques	Informatique
M. DEPREZ Gilbert	Physique du solide - christallographie

M. DERIEUX Jean Claude	Microbiologie
M. DERYCKE Alain	Informatique
M. DESCAMPS Marc	Physique de l'état condensé et cristallographie
M. DEVRAINNE Pierre	Chimie minérale
M. DEWAILLY Jean Michel	Géographie humaine
M. DHAMELINCOURT Paul	Chimie physique
M. DI PERSIO Jean	Physique de l'état condensé et cristallographie
M. DUBAR Claude	Sociologie démographique
M. DUBOIS Henri	Spectroscopie hertzienne
M. DUBOIS Jean Jacques	Géographie
M. DUBUS Jean Paul	Spectrométrie des solides
M. DUPONT Christophe	Vie de la firme
M. DUTHOIT Bruno	Génie civil
Mme DUVAL Anne	Algèbre
Mme EVRARD Micheline	Génie des procédés et réactions chimiques
M. FAKIR Sabah	Algèbre
M. FARVACQUE Jean Louis	Physique de l'état condensé et cristallographie
M. FAUQUEMBERGUE Renaud	Composants électroniques
M. FELIX Yves	Mathématiques
M. FERRIERE Jacky	Tectonique - Géodynamique
M. FISCHER Jean Claude	Chimie organique, minérale et analytique
M. FONTAINE Hubert	Dynamique des cristaux
M. FORSE Michel	Sociologie
M. GADREY Jean	Sciences économiques
M. GAMBLIN André	Géographie urbaine, industrielle et démographie
M. GOBLOT Rémi	Algèbre
M. GOURIEROUX Christian	Probabilités et statistiques
M. GREGORY Pierre	I.A.E.
M. GREMY Jean Paul	Sociologie
M. GREVET Patrice	Sciences Economiques
M. GRIMBLOT Jean	Chimie organique
M. GUELTON Michel	Chimie physique
M. GUICHAOUA André	Sociologie
M. HAIMAN Georges	Modélisation,calcul scientifique, statistiques
M. HOUDART René	Physique atomique
M. HUEBSCHMANN Johannes	Mathématiques
M. HUTTNER Marc	Algèbre
M. ISAERT Noël	Physique de l'état condensé et cristallographie
M. JACOB Gérard	Informatique
M. JACOB Pierre	Probabilités et statistiques
M. JEAN Raymond	Biologie des populations végétales
M. JOFFRE Patrick	Vie de la firme
M. JOURNEL Gérard	Spectroscopie hertzienne
M. KOENIG Gérard	Sciences de gestion
M. KOSTRUBIEC Benjamin	Géographie
M. KREMBEL Jean	Biochimie
Mme KRIFA Hadjila	Sciences Economiques
M. LANGEVIN Michel	Algèbre
M. LASSALLE Bernard	Embryologie et biologie de la différenciation
M. LE MEHAUTE Alain	Modélisation,calcul scientifique,statistiques
M. LEBFEVRE Yannic	Physique atomique,moléculaire et du rayonnement
M. LECLERCQ Lucien	Chimie physique
M. LEFEBVRE Jacques	Physique
M. LEFEBVRE Marc	Composants électroniques et optiques
M. LEFEVRE Christian	Pétrologie
Melle LEGRAND Denise	Algèbre
M. LEGRAND Michel	Astronomie - Météorologie
M. LEGRAND Pierre	Chimie
Mme LEGRAND Solange	Algèbre
Mme LEHMANN Josiane	Analyse
M. LEMAIRE Jean	Spectroscopie hertzienne

M. LE MAROIS Henri	Vie de la firme
M. LEMOINE Yves	Biologie et physiologie végétales
M. LESCURE François	Algèbre
M. LESENNE Jacques	Systèmes électroniques
M. LOCQUENEUX Robert	Physique théorique
Mme LOPES Maria	Mathématiques
M. LOSFELD Joseph	Informatique
M. LOUAGE Francis	Electronique
M. MAHIEU François	Sciences économiques
M. MAHIEU Jean Marie	Optique - Physique atomique
M. MAIZIERES Christian	Automatique
M. MANSY Jean Louis	Géologie
M. MAURISSON Patrick	Sciences Economiques
M. MERIAUX Michel	EUDIL
M. MERLIN Jean Claude	Chimie
M. MESMACQUE Gérard	Génie mécanique
M. MESSELYN Jean	Physique atomique et moléculaire
M. MOCHE Raymond	Modélisation,calcul scientifique,statistiques
M. MONTEL Marc	Physique du solide
M. MORCELLET Michel	Chimie organique
M. MORE Marcel	Physique de l'état condensé et cristallographie
M. MORTREUX André	Chimie organique
Mme MOUNIER Yvonne	Physiologie des structures contractiles
M. NIAY Pierre	Physique atomique,moléculaire et du rayonnement
M. NICOLE Jacques	Spectrochimie
M. NOTELET Francis	Systèmes électroniques
M. PALAVIT Gérard	Génie chimique
M. PARSY Fernand	Mécanique
M. PECQUE Marcel	Chimie organique
M. PERROT Pierre	Chimie appliquée
M. PERTUZON Emile	Physiologie animale
M. PETIT Daniel	Biologie des populations et écosystèmes
M. PLIHON Dominique	Sciences Economiques
M. PONSOLLE Louis	Chimie physique
M. POSTAIRE Jack	Informatique industrielle
M. RAMBOUR Serge	Biologie
M. RENARD Jean Pierre	Géographie humaine
M. RENARD Philippe	Sciences de gestion
M. RICHARD Alain	Biologie animale
M. RIETSCH François	Physique des polymères
M. ROBINET Jean Claude	EUDIL
M. ROGALSKI Marc	Analyse
M. ROLLAND Paul	Composants électroniques et optiques
M. ROLLET Philippe	Sciences Economiques
Mme ROUSSEL Isabelle	Géographie physique
M. ROUSSIGNOL Michel	Modélisation,calcul scientifique,statistiques
M. ROY Jean Claude	Psychophysiologie
M. SALERNO Françis	Sciences de gestion
M. SANCHOLLE Michel	Biologie et physiologie végétales
Mme SANDIG Anna Margarette	
M. SAWERYSYN Jean Pierre	Chimie physique
M. STAROSWIECKI Marcel	Informatique
M. STEEN Jean Pierre	Informatique
Mme STELLMACHER Irène	Astronomie - Météorologie
M. STERBOUL François	Informatique
M. TAILLIEZ Roger	Génie alimentaire
M. TANRE Daniel	Géométrie - Topologie
M. THERY Pierre	Systèmes électroniques
Mme TJOTTA Jacqueline	Mathématiques
M. TOURSEL Bernard	Informatique
M. TREANTON Jean René	Sociologie du travail

M. TURREL Georges	Spectrochimie infrarouge et raman
M. VANDIJK Hendrik	
Mme VAN ISEGHEM Jeanine	Modélisation,calcul scientifique,statistiques
M. VANDORPE Bernard	Chimie minérale
M. VASSEUR Christian	Automatique
M. VASSEUR Jacques	Biologie
Mme VIANO Marie Claude	
M. WACRENIER Jean Marie	Electronique
M. WARTEL Michel	Chimie inorganique
M. WATERLOT Michel	géologie générale
M. WEICHERT Dieter	Génie mécanique
M. WERNER Georges	Informatique théorique
M. WIGNACOURT Jean Pierre	
M. WOZNIAK Michel	Spectrochimie
Mme ZINN JUSTIN Nicole	Algèbre

Je remercie Jean-Paul Delahaye, Professeur à l'Université de Lille I, qui me fait l'honneur de présider le jury de cette thèse.

Je tiens à remercier Christian Carrez, Professeur au Conservatoire National des Arts et Métiers, pour l'attention qu'il a portée à ce document en acceptant d'être rapporteur de ma thèse et pour ses remarques pertinentes.

Mes remerciements vont également à Doug Tygar, Professeur à l'Université de Carnegie Mellon, USA, pour avoir accepté d'une part d'être mon rapporteur, et d'autre part, pour m'avoir accueilli au sein de son équipe de recherche pendant trois mois durant cette thèse.

J'exprime ma gratitude tout particulièrement à Vincent Cordonnier, Professeur à l'Université de Lille I, pour m'avoir guidé pendant toute la réalisation de ce travail. Le temps qu'il m'a consacré, ses compétences et conseils précieux ont fortement contribué à cet aboutissement.

Je remercie Philippe Maes, Vice-Président de Gemplus Card International, qui me fait l'honneur de participer au jury.

Je tiens également à remercier Pierre Paradinas pour sa participation au jury, mais aussi pour ses conseils et son soutien pendant la réalisation de cette thèse.

Enfin, je souhaite remercier tous ceux qui ont contribué de près ou de loin à ce travail, et particulièrement toute l'équipe de RD2P pour sa perpétuelle bonne humeur et son soutien quotidien.

Table des Matières

Introduction

Il existe aujourd'hui de nombreuses applications qui exploitent les atouts caractéristiques de la carte à micro-processeur: le regroupement de données sur cet unique support de taille réduite assure de façon sécurisée la portabilité d'informations associées à une personne ou un objet. En outre, la carte à micro-processeur facilite l'accès ou l'utilisation d'innombrables applications innondant le marché.

En raison de l'extension grandissante du domaine du multimédia, la carte doit désormais disposer de fonctionnalités plus avancées. Que ce soit en matière de sécurité, de structuration des informations ou d'intégration avec le monde environnant, la carte doit pouvoir manipuler et gérer des données devenues de plus en plus complexes.

Les possibilités de stocker et traiter dans la carte des informations autres que des entiers ou des chaînes de caractères sont nombreuses. Ceci semble particulièrement vrai dans les applications de type dossier portable. Il apparait également un large domaine d'utilisation dans la prise en comptes de données sonores ou visuelles que ce soit à des fins de sécurité, de santé, de travail ou de loisirs.

Si l'un des axes de la recherche sur la carte à micro-processeur s'intéresse à la réalisation d'architectures nouvelles afin d'améliorer les capacités de place mémoire et de puissance de traitement réduites des cartes existant actuellement *[PEYR94]*, d'autres portent sur l'amélioration de l'intégration de logiciels permettant une souplesse d'utilisation accrue des cartes futures destinées à supporter des applications multimédia. C'est sur ce second aspect que s'est orientée notre recherche.

Ce mémoire s'organise autour de quatre chapitres.

Le premier chapitre introduit les concepts de multimédia et Carte à micro-processeur. L'analyse des complémentarités entre ces deux domaines débouche sur les spécifications d'un modèle de données associé à la définition d'une carte adaptée au multimédia.

Nous avons identifié un domaine du multimédia qui coïncide avec l'une des caractéristiques essentielles apportées par la carte, la sécurité: il s'agit de l'identification biométrique et comportementale des individus.

Le second chapitre s'intéresse ainsi au contrôle d'accès utilisant les techniques de la Biométrie *[ALEX94b]*, qui permettent d'identifier un individu sur la base de critères physiologiques (empreintes digitales, reconnaissance vocale, etc...). Après une introduction aux concepts utiles à la compréhension des techniques biométriques, nous détaillons dans ce chapitre les études et travaux que nous avons menés sur ce sujet d'identification.

Le troisième chapitre concerne l'identification comportementale des individus, qui a pour but de renforcer le schéma de sécurité déja fourni par le contrôle d'accès de type mot de passe ou biométrie. Il s'agit de reconnaître un individu au sein d'une application par l'étude de ses pratiques reflétant des habitudes (par exemple reconnaître sa façon de conduire, de travailler sur un ordinateur ou de dépenser quotidiennement son argent). A ce sujet, nous introduisons le concept de «Radar» *[ALEX94a]*, utile à la détection de comportements inhabituels, et détaillons les développements que nous avons réalisés sur cet aspect.

Le chapitre IV se décompose en deux parties:

1. la première a pour objectif la définition d'une structure de données adaptée à la manipulation des données multimédia dans la carte à micro-processeur. Elle s'appuie notamment sur les observations formulées lors des chapitres précédents.

2. la seconde s'intéresse aux traitements associés à la structuration des données. A ce stade nous avons envisagé deux voies:

- Une approche par les **langages de haut niveau**
- Une approche par les **réseaux de neurones**

Pour chacune des deux approches, nous avons réalisé des travaux de recherche portant sur trois aspects multimédia utiles à la carte à puce: il s'agit de **la compression des données** (et en particulier d'une image de type photo d'identité) *[ALEX93]*, **l'identification biométrique** (au travers d'un système de reconnais-

sance de la signature du clavier) et **l'identification comportementale** (portant sur un système dédié au paiement par carte de crédit) *[ALEX94c]*.

L'ensemble des approches envisagées fera l'objet d'évaluations en termes d'occupation mémoire dans la carte et puissance de calcul requise.

De ces parties découlent la formulation d'un noyau commun de primitives destiné à effectuer dans la carte à micro-processeur des traitements variés sur des données multimédia . La fin de ce chapitre sera illustrée par des exemples applicatifs.

Enfin, nous nous tournerons vers de possibles orientations futures dans l'amélioration des travaux que nous avons synthétisés dans ce document.

CHAPITRE I

Le multimédia à la portée des cartes à micro-processeur

I.1 Introduction au multimédia

I.1.1 Qu'est-ce que le multimédia ?

Le multimédia c'est l'exploitation simultanée de données sonores, visuelles et informatiques, ainsi que les techniques de création, stockage, transmission et restitution de données qui permettent cette exploitation simultanée [TERR92].

Cette définition assez générale résume en quelques mots le vaste domaine abordé par le multimédia. Le monde dans lequel nous évoluons est constitué d'images et de sons que nous percevons par l'intermédiaire de nos sens. On a vu ces dernières années un intérêt grandissant dans l'exploitation informatique de ces données visuelles ou sonores, que ce soit à des fins professionnelles (industrie, économie) ou personnelles (loisirs, culture, formation).

Plusieurs critères peuvent répondre à cet engouement:

- La puissance du matériel informatique qui ne cesse de s'accroître et permet désormais de manipuler des données plus complexes que du texte, c'est à dire des images, des sons, dans des temps du même ordre de grandeur que nos interactions avec la réalité..

- L'évolution et la diversification des moyens de transmission de l'information: au téléphone, à la radio, au courrier s'ajoutent de nouveaux systèmes tels que le fax, les réseaux informatiques à haut débit, la télévision (y compris interactive), etc...

- L'apport de nouvelles technologies permettant le stockage de données volumineuses, à la fois concernant l'aspect matériel (par exemple le CD-ROM) et logiciel (les gestionnaires de systèmes documentaires, les bases de données multimédia, utilisant notamment la compression de données).

- La diversification dans les moyens de représentation de l'information. Davantage de représentations imagées sont maintenant accolées aux textes pour une acquisition et mémorisation de l'information plus efficaces.

- L'algorithmique de synthèse d'images et de sons.

Ces innovations technologiques répondent à une extraordinaire croissance de la demande. Les domaines de la santé ou de l'éducation par exemple mais aussi ceux des loisirs ou de la communication commerciale apparaissent comme des consommateurs insatiables.

Nous pratiquons, profitons ou subissons la communication multimédia de façon quotidienne peut être sans le savoir tout comme Monsieur Jourdain faisait de la prose. Par exemple, une campagne publicitaire qui s'appuie sur différents moyens promotionnels, de l'écrit d'une part au travers des magazines, du son et des images animées d'autre part par le biais de la télévision, est une communication multimédia.

Derrière ce terme vaste et flou de multimédia prodiguant l'arrivée du tout numérique se cache une révolution technologique, culturelle, économique qui émerge de la rencontre de deux univers, l'informatique et l'audiovisuel. Il est vrai que les enjeux sont de taille: le marché mondial de l'informatique multimédia en l'an 2001 est estimé a plus de 15 milliards de dollars, la télévision numérique devenant alors le premier produit multimédia avec plus de 33% du marché *[SEME92]*.

I.1.2 Les atouts du multimédia

Les ordinateurs multimédia intègrent désormais des technologies et des fonctionnalités qui ont permis d'aborder le traitement et la transmission de données vidéo-sonores en temps réel et de façon interactive *[ACMM93]*. Les principales évolutions concernent:

- Les supports de stockage: Les méga-octets de mémoire des supports magnétiques (disque dur) s'acquièrent pour des sommes modiques et le stockage sur support optique s'est beaucoup développé avec l'arrivée du CD-ROM *[MASC94]*.

- Les architectures des machines:
  D'après la loi de *Moore*, basée sur une observation fondamentale effectuée il y a plus de 20 ans par Gordon Moore, Président co-fondateur d'Intel, le nombre de transistors intégrés sur un micro-processeur double tous les 2 ans. A cette croissance technologique dans l'intégration du silicium s'ajoute la diversité des micro-processeurs spécialisés (ou co-processeurs) , processeurs de video, d'images, de pixels, de sons. C'est cette répartition fonctionnelle sur des composants relativement autonomes qui autorise des traitements parallèles très complexes et en temps réel, des transactions diverses avec les scanners de documents, appareils photo-numériques, caméscopes, magnétoscopes, CD audio et télévision.

- Les télécommunications: les progrès réalisés sont énormes. On peut citer pour références l'augmentation du nombre de satellites, l'évolution des supports de transmission numérique à haut débit, l'arrivée des «Autoroutes de l'Information». Les innombrables informations circulant sur le réseau Internet, s'adressant d'abord à une majorité de chercheurs et informaticiens, s'ouvrent désormais de plus en plus à un large public et sous des interfaces de visualisation plus accessibles. C'est le cas par exemple pour la recherche de documents utilisant les protocoles de communication du World Wide Web (développés à l'origine par le CERN), qui peuvent être consultés par des outils comme Mosaic mixant texte, sons, images fixes ou animées par le biais de formats hypertexte (voir *[HTML]*).

- Les interfaces, assurant notamment la conversion entre les données analogiques (signaux video ou audio) et numériques (données binaires). Le procédé de conversion, généralement appelé échantillonnage, permet l'acquisition de signaux continus représentant des images ou des sons sous une forme binaire codée exploitable par l'ordinateur (se reporter à *[SAND90]*), comme le montre le schéma suivant:

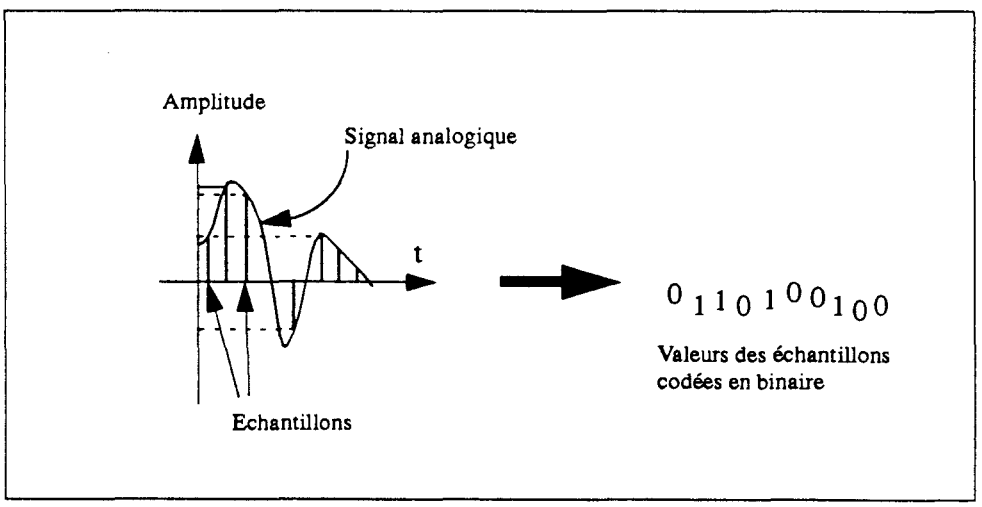


FIGURE 1 Principe de la conversion Analogique/Digitale

Le processus inverse, ou conversion Digitale/Analogique, permet de reconstituer une approximation du signal analogique original. L'amélioration des performances des ordinateurs en termes de vitesse d'exécution et d'espace de stockage permet l'acquisition et le traitement en temps réel d'un plus grand nombre d'échantillons, c'est-à-dire d'une représentation numérique plus fidèle des images ou des sons considérés.

- Les algorithmes de compression/décompression temps réel: ces algorithmes permettent de minimiser l'espace de stockage requis pour la sauvegarde ou la synthèse de sons et d'images, données souvent volumineuses. Le JPEG (Joint Photographic Experts Group) est un comité commun au CCITT et à l'ISO qui a défini et normalisé un algorithme standard pour la compression d'images fixes en couleur, dont le description est détailllée dans *[WALL91]*. Le comité MPEG (Motion Picture Experts Group) s'occupe de la définition d'un standard de normalisation de la video animée et de la transmission de données à très grande vitesse, décrit dans *[LEGA91]*.

Le multimédia ouvre ainsi ses portes à une multitude de domaines d'applications que nous allons détailler dans la partie qui suit.

I.1.3 L'univers des applications

Sans vouloir dresser une liste exhaustive des applications multimédia existantes, on peut dénombrer quelques types d'applications caractéristiques:

- Dans le domaine médical: l'acquisition en temps réel d'images vidéo-sonores et courbes de résultats et leur diffusion par réseau local dans des salles avoisinantes (ou par RNIS dans des centres spécialisés distants) offrent des perspectives d'amélioration de l'efficacité , de la fiabilité, de la productivité des diagnostics médicaux.

  Le multimédia médical concerne également la formation par l'apport d'encyclopédies médicales où les textes sont accompagnés d'images et de documents sonores aménagés de manière interactive pour l'utilisateur. Le développement de simulateurs médicaux permet en outre l'apprentissage de techniques complexes qui peuvent s'avérer dangeureuses si elles sont étudiées sur des patients «réels» *[HOFF94]*.

  On citera également l'archivage, la reconstitution d'images 3D ou encore le télédiagnostic.

- Les applications domestiques: les domaines privilégiés sont ceux de l'éducation (apprentissage des langues étrangères et autres disciplines), des jeux, des loisirs

(photographie, jardinage, sport...) et de la culture (visite de musées, peinture, musique...) *[BATE92]*.

- Le multimédia au service des entreprises: les techniques de compression/décompression des données numériques ouvrent un large éventail d'applications professionnelles, allant d'une simple PREAO (présentation assistée par ordinateur) ou PAO (publication assistée par ordinateur) enrichie par des séquences vidéo-sonores jusqu'aux communications multimédia d'entreprise par réseaux locaux et publics interposés. Ces dernières correspondent entre autres aux utilisations de messageries électroniques, téléconférences, discussions de groupes.

  En outre, l'évolution des interfaces Homme-Machine apporte une amélioration sensible dans le confort de travail et la productivité (par exemple la reconnaissance vocale ou celle de d'écriture manuscrite). Certaines techniques combinent plusieurs aspects, par exemple en améliorant la reconnaissance de la parole par la lecture des lèvres. Enfin, pour le travail dans des lieux demandant un accès sécurisé, le multimédia apporte des solutions d'identification plus fiables que des mots de passe car plus personnelles (la biométrie par exemple).

- Le multimédia concerne tous les professionnels de l'image et du son, la photographie, la publicité, la réalisation de montages ou films vidéo, par exemple utilisant des effets spéciaux et de la synthèse d'images, comme sur la figure suivante:

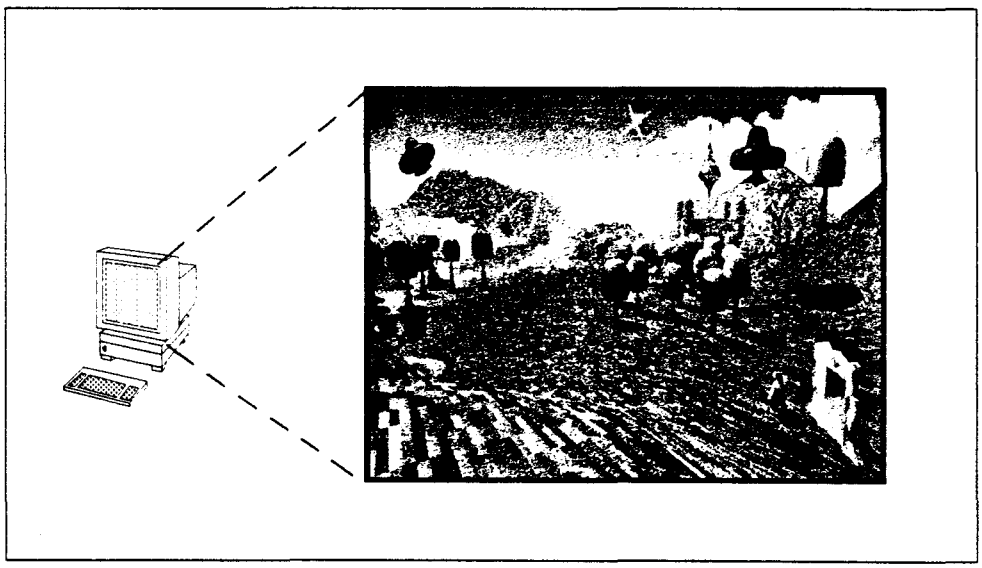


FIGURE 2 Exemple de synthèse d'image

- Dans le domaine militaire: le multimédia s'est principalement développé grâce à l'apport de technologies de pointes telles que la Réalité Virtuelle *[DEDE93]* qui permet l'entraînement de pilotes à la conduite aérienne sous forme de simula-

tion. Cette récente et nouvelle discipline numérique vise à reproduire idéalement toutes les sensations, visuelles, sonores, tactiles, voire olfactives que ressentirait dans la réalité un être humain mis en situation de simulation.

- Enfin, le multimédia trouve aussi ses applications dans le traitement et l'analyse de signaux, la surveillance de processus...

I.2 Introduction à la Carte à Micro-processeur

I.2.1 Définition

Inventée par le français Roland Moreno en 1974 et breveté sous le nom de «procédé et dispositif de commande électronique», le terme de carte à micro-processeur (ou carte à puce) désigne un support électronique de données doté d'une capacité de traitement, qui se présente sous la forme d'une carte de format réduit possédant un micro-processeur et son environnement (mémoires, entrées/sorties). Cette carte rectangulaire normalisée par l'ISO a pour dimensions:

$85,6mm \pm (0,12) \times 53,98mm \pm (0,05) \times 0,75mm \pm (0,02)$

Pour aboutir à la carte à micro-processeur que l'on connait aujourd'hui, la technologie des cartes est passée au travers de 3 générations, comme le montre la figure suivante:

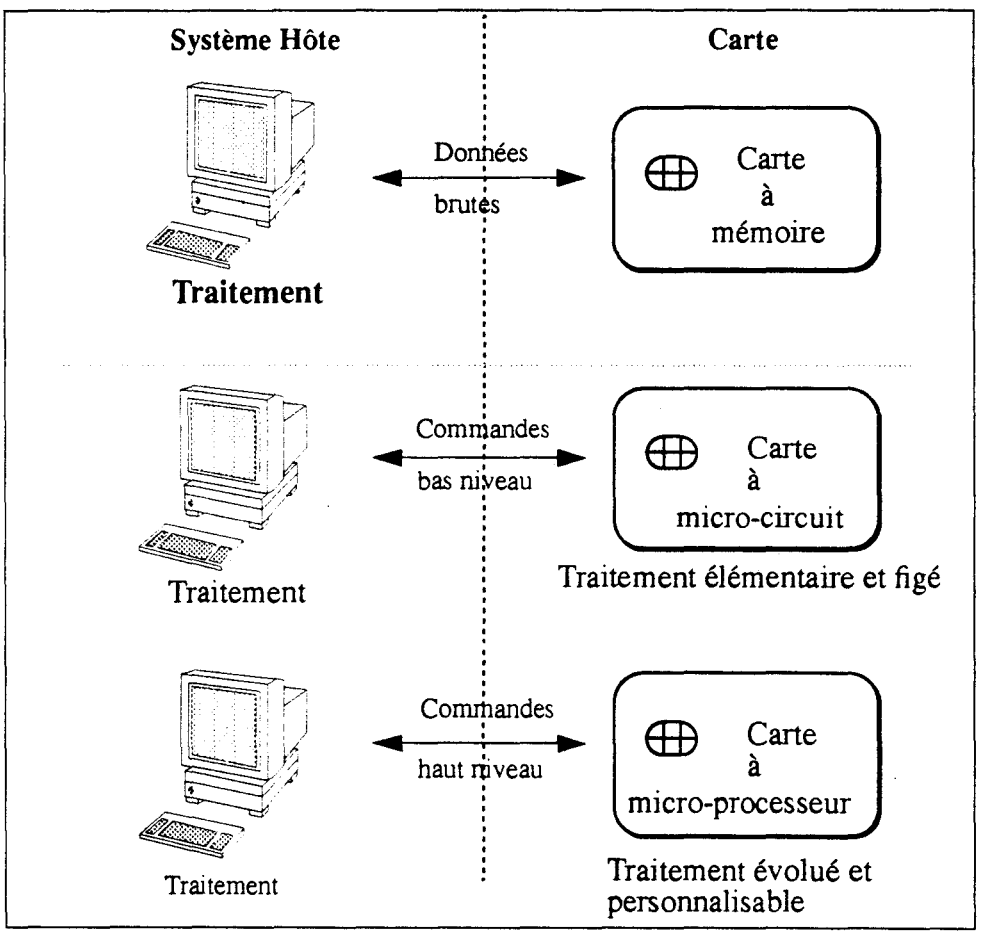


FIGURE 3 Evolution des cartes

La carte à mémoire (par exemple la carte à piste magnétique ou la carte optique), dépourvue de pouvoir de traitement peut être considérée comme un support de stockage de faible capacité, intéressante pour le transport et la mobilité d'informations peu volumineuses.

La carte à micro-circuit ajoute à la mémoire de la carte la possibilité d'effectuer des traitements élémentaires et non évolutifs (la Télécarte fait partie de cette catégorie).

Enfin, la carte à micro-processeur s'assimile à un micro-contrôleur, doté d'une unité de traitement évoluée, de mémoire et d'outils de communication. Elle correspond en fait aux micro-ordinateurs que nous avons l'habitude d'utiliser, différente cependant par son absence d'interfaces (pas de clavier ni d'écran) et sa puissance réduite pour le stockage et la manipulation des données.

I.2.2 Les composantes des cartes à micro-processeur

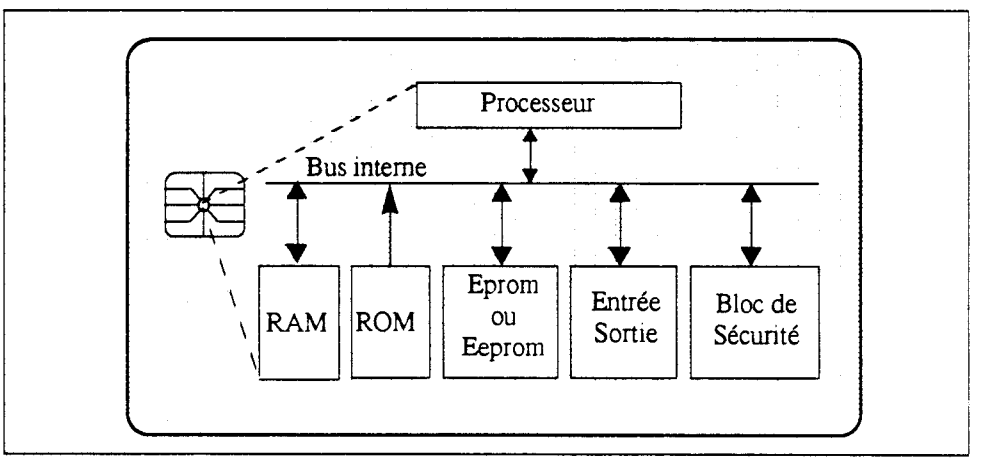


FIGURE 4 Composantes d'une carte à micro-processeur

Une carte à micro-processeur peut se résumer à 3 constituants:

- Un support généralement plastique normalisé par l'ISO.
- Une pastille de contacts, permettant la connexion du micro-processeur avec l'environnement extérieur (pour l'alimentation électrique, l'horloge et les communications).
- Un micro-module, lui-même caractérisé par un ensemble d'éléments:

1. Un micro-processeur 8 bits

Ces types de micro-processeurs (Intel 8051, Motorola 6805), utilisés depuis plus de 10 ans et n'ayant pas été conçus à l'origine pour répondre aux besoins spécifiques des cartes, possèdent en général une faible puissance de calcul. D'autres processeurs arrivés plus récemment tels le Hitachi H8-310 ou le SGS-Thomson ST9 ont marqué une légère évolution.

D'autres processeurs plus performants sont à l'étude actuellement, notamment autour des architectures RISC. Le projet Européen ESPRIT-CASCADE *[PEYR94]* s'intéresse aux futures processeurs pour cartes à micro-processeur et en particulier d'une architecture RISC 32 bits possédant des fonctions avancées pour la sécurité (cryptographie) et l'identification (biométrie intégrée utilisant des réseaux de neurones) (voir également le déliverable *[CASC94]*).

2. Une mémoire RAM (Random Access Memory)

C'est la mémoire de travail, utilisée pour le stockage des résultats provisoires des différents calculs et pour la création d'une mémoire tampon d'entrée/sortie. Cette mémoire n'est alimentée que lorsque la carte est connectée.

3. Une mémoire ROM (Read Only Memory)

Cette mémoire contient le code de l'application ainsi que le système d'exploitation de la carte. Son continu est fixé pendant la fabrication et ne peut plus être modifié ultérieurement.

4. Une mémoire EEPROM (Electrically Erasable Programmable Read Only Memory)

Elle est utilisée pour le stockage des données permanentes, car contrairement à la RAM cette mémoire est non volatile. Son contenu peut être effacé et reprogrammé électriquement.

5. Un bloc de sécurité

Il assure les contrôles relatifs à la sécurité physique du composant en termes de détection de lumière, détection de variations de la fréquence d'horloge... et permet ainsi de faire face à des attaques physiques. Certains composants sont en outre dotés d'une «matrice de sécurité», décrite dans *[CARO94]*.

6. Des entrées / sorties

Les entrées/sorties se font en mode série en implémentant sur une position de la mémoire RAM une liaison 1 bit vers l'extérieur. C'est le système d'exploitation en ROM qui se charge de la réalisation fonctionnelle du port série.

Le tableau suivant résume l'évolution des différentes mémoires :

Mémoire	1981	1993
ROM	1600 octets	10 Koctets
RAM	36	256 octets
EPROM	1 Koctets	-
EEPROM	-	8 Koctets

FIGURE 5 Capacités de stockage des différentes mémoires

Pour davantage de détails sur la constitution des cartes à puce, consulter *[BRIG88]*.

I.2.3 Un aperçu des applications

Aujourd'hui les applications cartes se multiplient et abordent de plus en plus de domaines. Il semblerait que l'on puisse néanmoins les classer en 3 groupes de la manière suivante *[CRIN90]:*

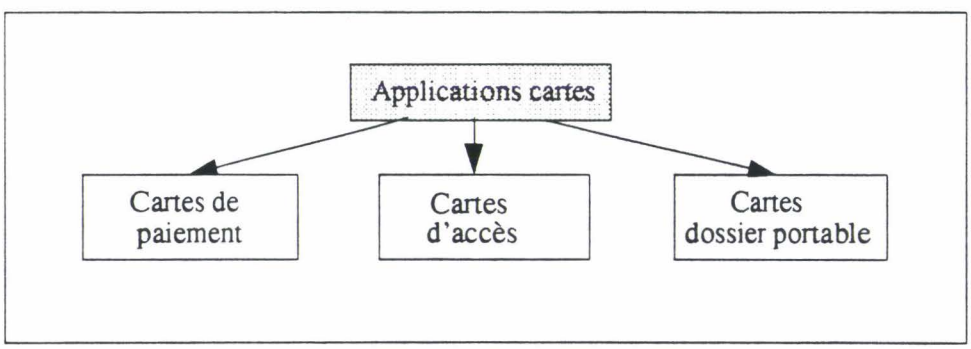


FIGURE 6 Répartition des applications cartes

La plupart de ces applications étant détaillées dans *[CARO94]*, nous voulons ici seulement rappeler les principales:

1. Les cartes de paiement

Elles peuvent être de type pré-paiement (aussi connues sous le nom de porte-monnaie électronique) ou de type paiement différé. L'exemple le plus classique de pré-paiement est la Télécarte, celui de paiement différé est la Carte de Crédit.

2. Les cartes d'accès

Ce type de carte joue le rôle d'identification ou d'authentification. Elles permettent à un utilisateur d'accéder par exemple à un serveur (accès logique) ou à un lieu sécurisé (accès physique) par le biais d'un mot de passe ou d'un système d'identification biométrique (voir chapitre II).

3. Les cartes de type Dossier Portable

Cette famille de cartes, en général plus puissante en ressources de calcul et plus riche en place mémoire, permet de stocker et manipuler des données variées dans un environnement sécurisé. Un dossier portable peut contenir par exemple des informations sur la scolarité d'un individu (carte étudiante), des résultats d'examens médicaux (la carte santé *[PAPI94]*), des données relatives à un véhicule (la carte transport).

Le nombre d'applications de ce type ne cesse d'augmenter. C'est pourquoi on s'intéresse de plus en plus aux possibilités de traitement évolué dans la carte. De nombreuses cartes sont déjà disponibles et utilisées sur le marché. Certaines se caractérisent par le fait de ne pas être spécifiques à une application particulière mais sont destinées à de nombreuses applications. Il s'agit par exemple de la Carte CQL (Card Query Language) *[PARA94] [GORD92]*, similaire à un gestionnaire de bases de données embarqué sur la carte et disposant d'outils d'interrogation faisant partie du standard SQL (voir également *[GRIM92]*).

I.3 La carte à micro-processeur adaptée au multimédia

I.3.1 Analyse des complémentarités entre multimédia et carte

Les applications futures des cartes sont destinées à répondre aux besoins croissants en matières de mobilité et sécurité. Etant donné l'ampleur prise par le multimédia dans le monde informatique d'aujourd'hui, il paraît intéressant de caractériser le rôle que la carte peut jouer dans de telles applications. Inversement,

il faut souligner les fonctionnalités multimédia qui peuvent enrichir une carte d'usage général.

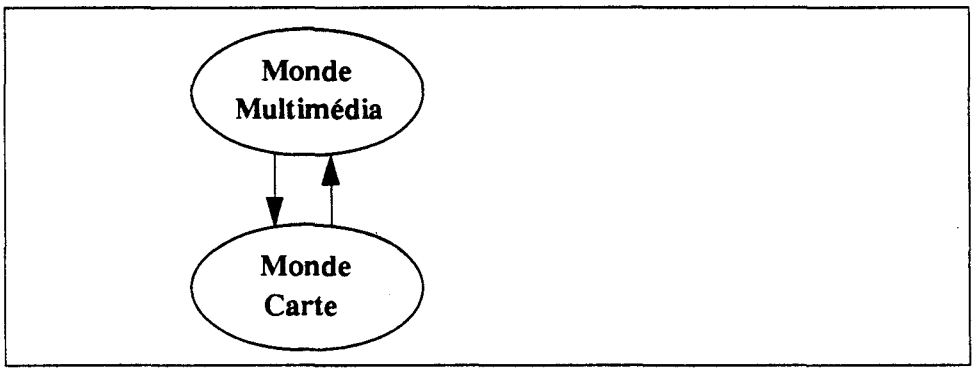


FIGURE 7 Complémentarités entre multimédia et Carte à puce

Le multimédia déja présent sur la plupart des ordinateurs de bureau exploite des données visuelles et sonores qui souvent représentent de gros volumes d'informations en comparaison avec les traditionnelles données de type texte. Si la carte à micro-processeur ne peut en aucun cas rivaliser avec les architectures des machines multimédia couramment utilisées (processeurs 80486, Pentium, Power PC...) étant données ses faibles capacités de stockage et traitement, elle peut néanmoins leur apporter plusieurs atouts:

- la mobilité: la taille réduite d'une carte, tenant aisément dans une poche ou même dans un portefeuille, lui permet de véhiculer des informations utiles à son porteur. Il peut s'agir de données d'identification, de données médicales ou administratives ou de toute autre forme d'information de petite taille.

- la sécurité: cet atout majeur devient de plus en plus important dans la mesure ou de nombreuses applications multimédia mettent en jeu des systèmes de paiement ou tout simplement requièrent une identification de la part de leurs utilisateurs.

De son côté, le multimédia peut enrichir une carte à puce, notamment en renforçant l'aspect sécurité ou en améliorant la gestion des données embarquées. On peut citer par exemple, pour l'aspect sécurité, l'identification au moyen de photos d'identité ou de données biométriques, et pour l'aspect gestion les techniques de représentation (description algorithmique, arborescente...) ou de compression des données.

Nous avons choisi de qualifier une telle carte de Carte à Micro-processeur adaptée au Multimédia dans la mesure où les techniques que nous employons pour enrichir la carte de nouvelles fonctions de sécurité font partie intégrante du Multimédia. En effet, la biométrie regroupe des algorithmes de stockage et de traitement de signaux audio ou video (reconnaissance de la voix, du visage). La compression

d'images de type JPEG est également l'une des composantes ayant fortement contribué à l'explosion du monde multimédia.

I.3.2 Objectifs de la présente thèse

Le but de ce travail est de définir une carte à micro-processeur destinée à supporter des applications multimédia. Deux stratégies s'offrent à nous pour aborder le problème:

- Une approche orientée applications: pour une application multimédia donnée et présentant un intérêt pour la carte à micro-processeur, il s'agit de l'implémenter en simplifiant le stockage et le traitement des données de façon à répondre aux exigences sévères de la carte en termes de temps de réponse, place mémoire, etc...

- Une approche orientée fonctions: à partir d'un certain nombre d'observations, d'études et de développements liés à des applications multimédia, il s'agit d'identifier un modèle de données compatible avec une large gamme d'applications et relevant d'un jeu de commandes unique ou du moins unifié autour d'un noyau et d'éventuelles extensions plus spécifiques. Les données et les traitements doivent bien entendu être totalement compatibles.

Nous avons privilégié la seconde approche parce qu'elle tend à définir une carte générale ouverte à des applications variées et évolutives et non pas un cas particulier de carte.

Le champ d'applications visé ne relève que d'un sous-ensemble du multimédia. On abandonne dès le départ l'idée de mettre en oeuvre des modèles et fonctions standards tels que ceux qui servent de supports aux applications multimédia traditionnelles.

De ce fait, les techniques basées sur de gros volumes d'informations, proches de la synthèse d'image par exemple (modélisation de volumes 3D, calcul de surfaces cachées, d'éclairement, aliasing) sont immédiatement éliminées. De telles approches réclament de puissants processeurs et des méga-octets de mémoire. Cela réduira évidemment les possibilités d'applications à ce que l'on peut espérer stocker et traiter dans une carte. Il est possible que la plupart des applications s'en satisfassent car on tentera de limiter la complexité des objets et des traitements à ce qui semble réellement utile aux applications les plus vraisemblables.

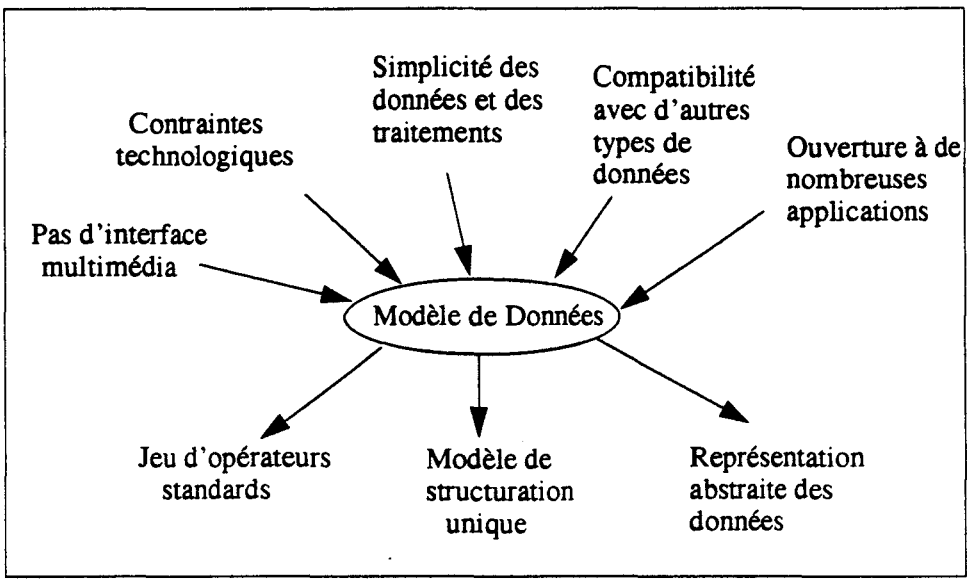


FIGURE 8 Le modèle de données

Le modèle de données présentera les caractéristiques suivantes:

- La prise en compte des contraintes d'implémentation, en particulier des performances limitées des cartes actuelles ou à venir prochainement en termes de capacité de stockage et puissance de traitement.

- Une structure interne supportant une certaine complexité toutefois modérée par le respect d'une exigence de simplicité vis-à-vis de l'application.

- Des données synthétiques regroupant de manière implicite et automatique un grand nombre de scalaires dans une structure unique de référence.

- La capacité à supporter des opérateurs variés.

- L'ouverture à de nombreuses applications: pour une extension à des applications de type multimédia, les modèles de données des cartes actuelles sont tout à fait inadéquats (voir *[ISO94]*). Par exemple, les fichiers élémentaires de travail sont à contenu inconnu dans la carte et donc manipulables uniquement de l'extérieur.

- La prise en compte de contraintes technologiques telles que la transmission série, la faiblesse des opérateurs cablés, l'organisation de la mémoire.

Nous avons identifié un domaine du multimédia qui coïncide avec l'atout majeur apporté par la carte à micro-processeur, la sécurité: il s'agit de l'identification biométrique et comportementale des individus.

Dans les 2 chapitres qui suivent, nous allons détailler les travaux de recherche que nous avons menés sur ces sujets d'identification. Certains correspondent à des études de systèmes existants, d'autres sont des travaux que nous avons réalisés: nous nous efforcerons de bien faire ressortir la distinction.

Le chapitre IV établit dans un premier temps la définition de la structure de données que nous avons élaborée à partir des observations notamment des chapitres II et III, puis détermine dans un second temps les traitements associés à ces données. A ce stade nous avons exploré 2 voies principalement:

- une approche par les langages de haut niveau.
- une approche par les réseaux de neurones.

CHAPITRE II *L'Identification Biométrique*

Introduction

Depuis la Rome Antique, il a toujours été indispensable de développer des techniques pour vérifier l'identité des individus. Aujourd'hui, parallèlement à la multiplication et la diversification des services proposés sur les systèmes interactifs sécurisés s'accentuent les besoins en matière d'identification. Les différents partenaires touchés par ce besoin d'identification peuvent être:

- **Un ordinateur**

- **Un serveur**

- **Un terminal**

- **Un individu**

Dans le cas de l'identification de l'individu, les techniques permettant de procéder à la vérification de son identité sont nombreuses. En effet, un individu peut s'identifier par:

- ce qu'il sait (mot de passe, PIN (Personal Identification Number) ou autre)
- ce qu'il est (ses particularités biométriques)
- ce qu'il fait (son comportement)

On peut critiquer le «ce qu'il sait» en raison des possibilités de clonage, d'oubli, de copie explicite, d'essais systématiques (par exemple le balayage de tous les PIN ou mots de passe), etc...

Ces derniers éléments cités permettent de souligner l'intérêt des deux autres moyens d'identification. Ce chapitre s'intéresse aux aspects biométriques (le «ce qu'il est»), le suivant traitera l'approche comportementale («ce qu'il fait»).

A l'origine, le terme de **Biométrie** désigne l'analyse statistique de mesures effectuées sur des phénomènes biologiques. En matière de sécurité informatique, cette appellation a été rapidement attribuée à la description des technologies utilisées dans l'identification personnelle des individus à partir de caractères physiologiques ou comportementaux.

Dans une première partie, nous allons introduire les notions communes à tout processus d'identification biométrique et nécessaires à leur exécution et évaluation.

La seconde partie souligne l'intérêt et le rôle de la carte dans les systèmes d'identification biométrique.

La troisième s'intéresse à la description d'un certain nombre de solutions que nous avons soit étudiées ou réalisées afin d'en extraire les caractéristiques communes. Il s'agit principalement des processus suivants:

- La reconnaissance de la voix
- La géométrie de la main
- L'identification du visage
- Les empreintes digitales
- Le fond de l'oeil
- La dynamique de la signature manuscrite
- La dynamique de la signature clavier

Cette partie débouche sur une comparaison des systèmes en termes de fiabilité, rapidité d'exécution des algorithmes et consommation mémoire, mais aussi en termes de coût et de perception de la part des individus à qui ils sont destinés (simplicité d'utilisation, sensation de sécurité ou d'insécurité vis-à-vis d'un système...).

II.1 Définition des principales notions

L'identification biométrique au sens où elle sera perçue dans la suite de ce document concerne l'identification d'un individu à l'aide de l'une ou plusieurs de ses caractéristiques physiologiques (son visage, son empreinte digitale...) ou comportementales (sa signature par exemple).

Le terme de système d'identification biométrique désigne le matériel et logiciel nécessaires au processus d'identification. On peut dénombrer 2 étapes majeures dans l'appréhension de tout système biométrique:

- La constitution de la Référence: il s'agit pour un individu donné d'établir un profil d'une ou plusieurs de ses caractéristiques (cela peut être une image de son empreinte digitale ou un ensemble de paramètres décrivant sa manière de réaliser une signature). Cette Référence lui est propre et représente son identité.

- La phase de Vérification (ou de Test): lors de cette étape, un profil "Candidat" est proposé au système, qui, par comparaison avec la Référence, détermine le degré de ressemblance des deux profils et valide ou non l'identification.

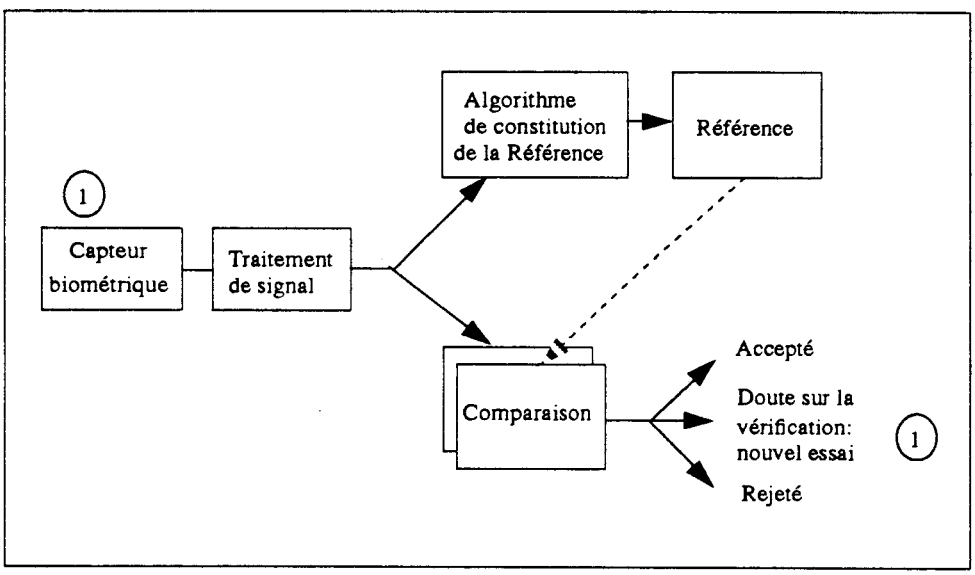


FIGURE 9 Les composantes d'un système d'identification biométrique

Une variante à l'identification biométrique, appelée Reconnaissance biométrique, consiste à rechercher dans une base de données (constituée de plusieurs profils de référence), le profil qui ressemble le plus au Candidat proposé. C'est le cas par exemple lorsqu'à partir d'empreintes digitales prélevées sur les lieux d'une enquête, le système biométrique recherche au sein d'une vaste base de données

d'éventuels suspects, ceux dont le profil semble correspondre le plus à celui prélevé.

Etant donnée la différence existant entre deux profils prélevés d'un même individu (si petite soit elle), un système d'identification biométrique ne peut jamais garantir à 100% le résultat de son identification. Ce dernier aspect le différencie de tous les systèmes basés sur les mots de passe et aussi utilisés comme moyen d'identification pour sécuriser un accès.

Par exemple, retirer de l'argent dans un guichet automatique à l'aide d'une carte de crédit requiert un mot de passe ou PIN (Personal Identification Number). De façon similaire au processus d'identification biométrique, le PIN donné au guichet par un utilisateur (le Candidat) peut être comparé au PIN inscrit dans la carte (la Référence). Pour valider l'accès, les deux PIN doivent correspondrent exactement.

Ceci n'est pas le cas pour les systèmes biométriques qui pour un même individu enregistrent des profils différents à chaque fois (soit parce que les caractères physiologiques ou comportementaux d'un individu ont évolué, soit parce que les capteurs biométriques (un micro, une caméra, une tablette graphique) ou les conditions d'acquisition des données (bruit ambiant dans l'enregistrement d'une voix par exemple) sont différents.

Ceci nous amène donc à définir les notions de "Faux Accepté" et de "Vrai Rejeté". Il s'agit des deux cas d'erreurs possibles éventuellement commises par un système d'identification biométrique.

- Faux Accepté (FA): ce cas se produit lorsqu'un individu au profil "Candidat" est identifié par un système biométrique comme correspondant à un profil "Référence" qui ne lui appartient pas (le terme Faux est associé en général au Fraudeur).

- Vrai Rejeté (VR): il apparaît lorsqu'un individu au profil "Candidat" **n'est pas** identifié par un système biométrique comme correspondant à un profil "Référence" qui pourtant lui appartient (le Terme Vrai est associé au Véritable porteur du profil "Référence").

- De ces deux aspects découlent immédiatement les notions de **Taux de Faux Acceptés (TFA)** et **Taux de Vrais Rejetés (TVR)** qui permettent d'établir le degré de fiabilité d'un système biométrique. Ces deux taux sont liés de la façon suivante:

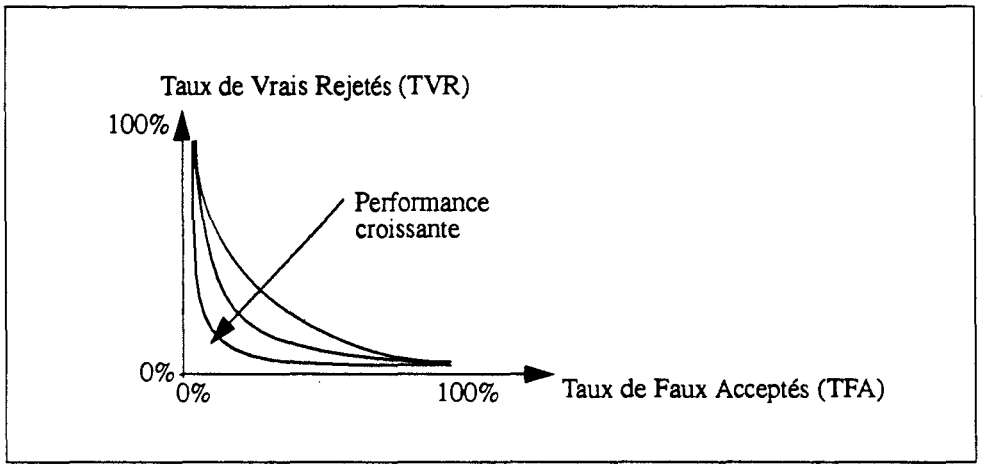


FIGURE 10 Relation entre le Taux de Faux Acceptés (TFA) et le Taux de Vrais Rejetés (TVR)

Chaque courbe sur le graphique correspond à un système biométrique différent. Il appartient au propriétaire du système d'identification considéré de choisir le point de réglage sur la courbe correspondant à ses besoins d'identification au sein de son application: les banquiers imposent en général un TVR inférieur à 1/100000 pour des applications destinées à un large public (le controle d'accès sur les cartes bancaires notamment, en remplacement du PIN code), même si dans ce cas le TFA (taux de fraude) n'en est que davantage réhaussé (un client voulant avoir accès à sa carte par exemple ne doit en aucun cas être gêné par un système qui ne l'identifie pas).

L'espace d'identification

Nous avons dit précédemment que pour valider l'identification lors de la phase de Vérification, le système évalue un degré de «Ressemblance» entre le Test et la Référence. Pratiquement , cela correspond au calcul d'une distance d entre un ensemble de valeurs numériques constituant la Référence et un ensemble de valeurs numériques constituant le Test. On supposera donc que l'information utile à l'identification se compose d'un ensemble limité de n valeurs numériques. Ainsi l'espace d'identification sera défini dans R^n.

Un Test T représente un point de cet espace. Le plus souvent on réduira les dimensions de l'espace en réalisant des compositions linéaires des valeurs initiales. De même, une Référence R figure comme un point particulier, et l'identification consiste à constater que R et T sont proches, voire confondus.

Par exemple, la carte bancaire propose un espace à une dimension d'entiers compris entre 0 et 9999 et n'accepte que la stricte identité (d=0).

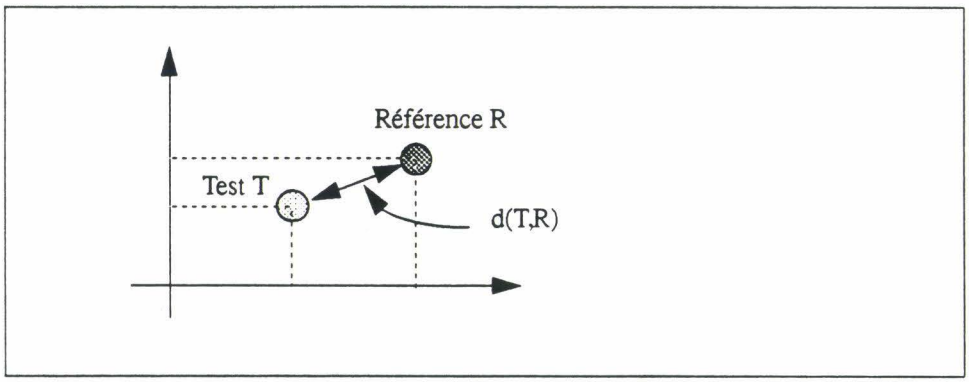


FIGURE 11 L'espace d'identification

La Boule d'identification

Contrairement à l'exemple bancaire donné ci-dessus, la Référence n'est en général pas reproductible identiquement à elle-même, en raison des bruits qui viennent entâcher la production du signal biométrique, comme évoqué précédemment. C'est pourquoi on associe à un individu une Boule B(R,d) au sens topologique du terme, de façon à ce que tout Test qui se trouve à une distance de R inférieure à une distance seuil d soit reconnu comme ressemblant à R et donc identifie l'émetteur.

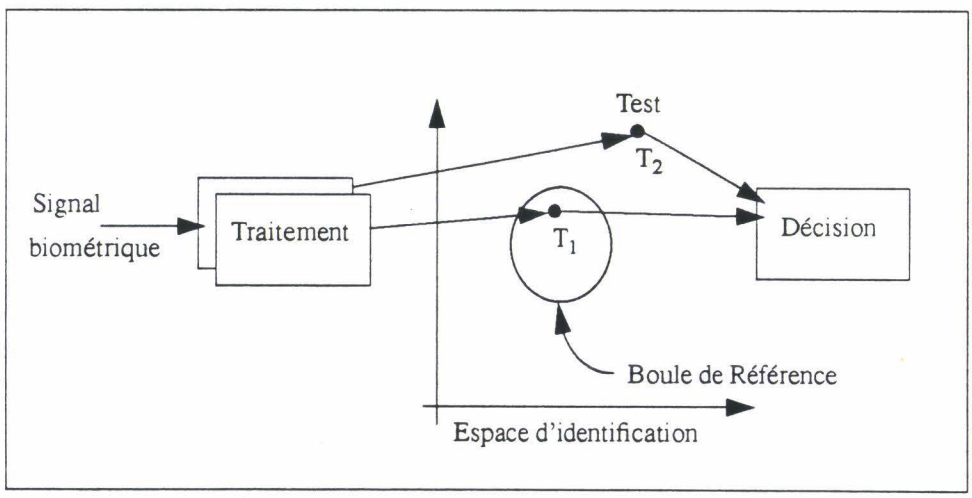


FIGURE 12 Le processus d'identification

La dimension de la boule ou, plus précisément, son rayon, détermine la tolérance sur les échantillons de Test. Une boule large contiendra tous les Tests mais risque de recouvrir le territoire délimité par une boule voisine.

La notion de «Vrai Rejeté» (VR) décrite précédemment correspond au cas où un individu donné produit un Test en dehors de sa boule de Référence.

Le «Faux Accepté» (FA) correspond au cas où un individu autre que le détenteur de la boule de Référence produit un Test à l'intérieur de celle-ci.

Pour les boules réduites de la figure ci-dessous (grisées), le Taux de Faux Acceptés (TFA) est nul ou très faible entre les deux individus A et B. En revanche TVR est élevé. Pour les boules larges, TVR est nul mais TFA devient important, autorisant soit une incertitude sur l'identité réelle de l'individu présentant un échantillon de Test, soit une fraude de l'un ou l'autre.

On voit donc l'importance de l'expérimentation sur un système d'identification d'individu.

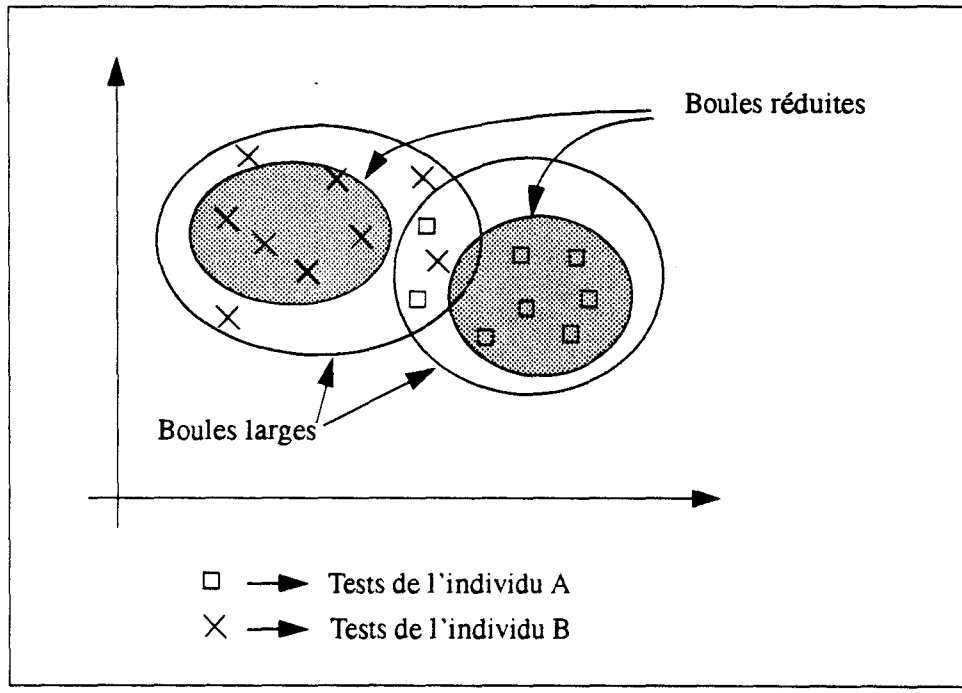


FIGURE 13 La notion de Boule d'identification

L'Evaluation

La mesure et comparaison des performances de divers systèmes d'identification biométrique requièrent enfin l'introduction de **Critères d'Evaluation**, à la fois concernant l'aspect technologique et l'aspect économique et social. On citera, pour l'aspect technologique:

- la fiabilité d'identification: il s'agit de comparer les TVR/TFA des systèmes.
- le temps requis pour la constitution du profil Référence.
- le temps nécessaire à la phase de Vérification.
- la taille de la Référence..
- la complexité de l'algorithme.
- la stabilité de la boule de Référence dans le temps.

Pour l'aspect économique et social:

- le coût du système (souvent lié au type de capteur biométrique utilisé et à la complexité des calculs requis par l'algorithme).
- la perception qu'ont les utilisateurs quant à la fiabilité et la facilité d'utilisation du système.

II.2 Intérêt et rôle de la carte par rapport à l'identification biométrique

II.2.1 Intérêt d'utiliser une carte dans un système biométrique

Comme nous l'avons brièvement évoqué dans la partie précédente, la biométrie peut être abordée de deux manières:

1. une solution aveugle:

Dans cette approche, l'ensemble des Références des différents individus est stocké dans une base de données et l'individu ne fournit que l'échantillon de Test. Le traitement biométrique consiste alors à rechercher, dans la base de données centralisée, s'il existe une Boule de Référence dans laquelle se situe le Test. Afin de ne pas avoir de confusion sur l'identité de l'individu, cela suppose que les Boules de Référence contenues dans la base de données soient disjointes.

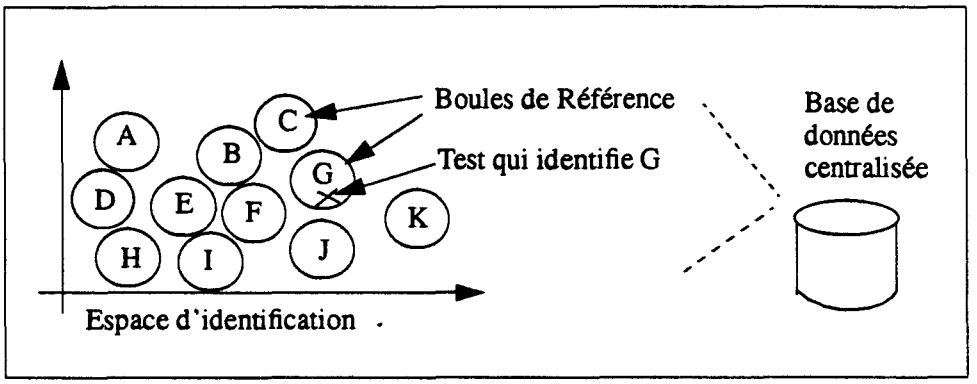


FIGURE 14 La solution centralisée

Plusieurs critiques peuvent être formulées sur cette solution:

- Le traitement biométrique qui doit se faire sur le site centralisé accroît le temps de réponse en raison du temps de communication nécessaire à l'acheminement du Test vers le site central et de la récupération de la réponse de décision.

- La communication évoquée dans le point précédent affaiblit en outre la sécurité du système (on peut crypter bien entendu le Test pour l'envoyer sur le site ainsi que la réponse d'identification mais ce calcul supplémentaire ne fait qu'augmenter le temps de réponse).

- Le système doit disposer d'un mécanisme de scrutation de toutes les Références. On observe que toutes les boules doivent alors être disjointes. Cela est d'autant plus difficile que le nombre de Références est élevé. Il est aussi très délicat de mettre le système à jour car toute nouvelle Référence doit vérifier ce critère. S'il advient qu'il ne puisse être respecté, l'ensemble du système devient caduque.

2. une solution avec référence:

Dans cette approche, l'individu fournit deux informations:

- La Référence sous une forme figée et inviolable (la carte sert alors à stocker cette référence de façon sécurisée)

- Un Test

Dans ce cas, le traitement d'identification biométrique peut s'effectuer de façon locale et consiste à vérifier que le Test fourni appartient à la Boule de Référence.

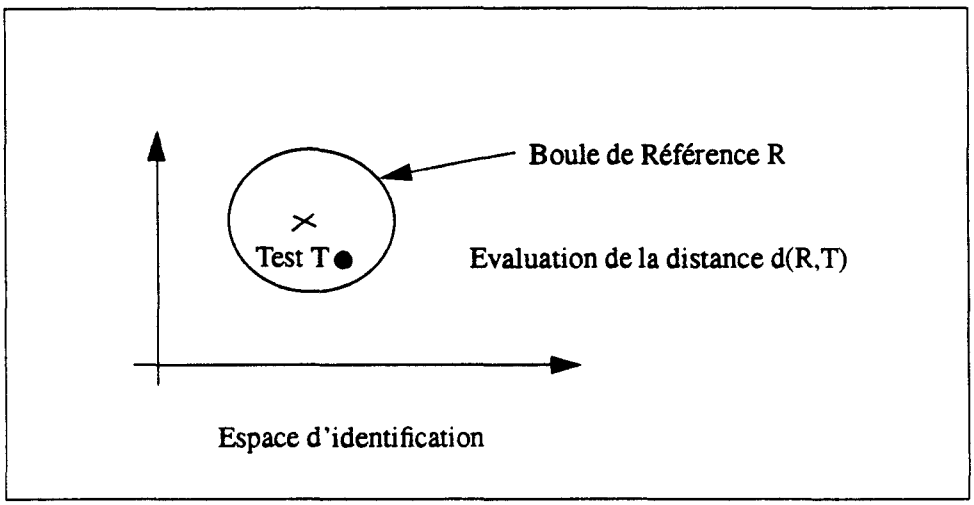


FIGURE 15 La solution avec Référence

On entrevoit immédiatement les avantages apportés par ce procédé:

- La décentralisation permet d'éviter ou de limiter les communications.
- Le traitement est plus réduit puisque circonscrit à une seule Référence.
- Il autorise des recouvrements des Boules de Référence.

Etant données les critiques évoquées concernant chacune des solutions, on préfèrera la seconde approche. A ce stade, la carte permet de stocker les données constituant la Référence et représente ainsi l'identité de l'individu.

II.2.2 L'utilité de la carte dans la sécurité du système biométrique

Le rôle de la carte ne doit pas s'arrêter à la simple portabilité d'informations biométriques de taille réduite. En effet, pour des raisons de sécurité et donc de validation de l'identification, aucun traitement frauduleux ne doit pouvoir modifier ou remplacer tout ou partie de la Référence ou du Test ainsi que l'algorithme de comparaison débouchant sur la décision d'identification. C'est pourquoi le traitement biométrique doit se faire dans un environnement sécurisé.

De la découle l'intérêt d'intégrer à la carte le traitement biométrique afin de ne jamais livrer la Référence à un environnement extérieur, la carte garantissant la validité de l'identification par son inviolabilité.

II.2.3 Les contraintes d'intégration de la biométrie dans la carte

Si l'identification biométrique dans la carte paraît séduisante par la garantie de sécurité qu'elle apporte, elle n'est possible qu'en répondant aux contraintes sévères induites par l'environnement carte, qui sont:

- La puissance de calcul très limitée des cartes actuelles construites autour de processeurs 8 bits (cadencés à 3,57 MHz). A titre d'information, un certain nombre de systèmes biométriques commercialisés sont construits autour de processeurs puissants et spécialisés tels que des DSP (Digital Signal Processing).

- La taille très réduite des mémoires. Les cartes les plus performantes d'aujourd'hui ne sont dotées que d'environ 256 octets de RAM, 10 Koctets de ROM et 16 Koctets d'EEPROM.

- La faiblesse de communication série des cartes avec les équipements extérieurs, limitant ainsi la possibilité d'échange de données biométriques entre ces deux environnements. Les cartes standards du marché fonctionnent à 9600 bauds, les plus puissantes à 19200 bauds.

Nous avons étudié et expérimenté un certain nombre de systèmes biométriques avec pour objectif principal d'une part de vérifier s'ils sont compatibles avec les contraintes de la carte et d'autre part d'identifier des solutions répondant aux contraintes citées ci-dessus. Nous verrons que certains d'entre eux correspondent davantage à ce que l'on peut espérer intégrer dans une carte à micro-processeur en termes de capacité de stockage et de puissance de traitement.

II.3 Etude de systèmes d'identification biométrique

De la grande variété des systèmes d'identification biométrique utilisés actuellement on peut extraire deux principales catégories:

- les systèmes basés sur des critères physiologiques (géométrie de la main, empreintes digitales, fond de l'oeil etc...)

- les systèmes basés sur des critères comportementaux. Il s'agit entre autres de la signature manuscrite, de la dynamique de la signature clavier etc...

Il existe également des systèmes dotés des deux composantes simultanément. C'est le cas pour la reconnaissance vocale qui combine le caractère physiologique de l'organe vocal avec des aspects plus comportementaux tels que le stress par exemple. Parmi tous les procédés biométriques, certains utilisent des outils de

traitement d'images ou de sons globaux, alors que d'autres se font par extraction d'uniquement quelques paramètres caractéristiques. Nous soulignerons ces différences au fur-et-à-mesure des études.

II.3.1 Les systèmes physiologiques

II.3.1.1 Les empreintes digitales

Probablement le système le plus populaire à l'heure actuelle, la recherche des empreintes digitales est utilisée dans une large gamme d'applications *[CART94][IDEN93]*. La plupart des systèmes analyse une empreinte digitale en retenant uniquement les positions et orientations de certaines particularités, qui apparaissent sous la forme de petits cercles ou de Y *[STAR93]*.

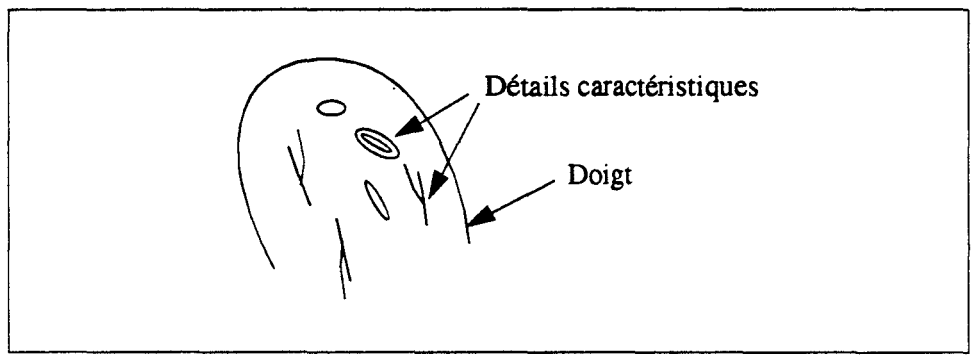


FIGURE 16 Extraction de détails caractéristiques d'une empreinte digitale

En général, seule une zone proche du centre morphologique de l'empreinte est étudiée, celle qui est riche en informations caractéristiques. La saisie de l'empreinte s'effectue à partir d'un capteur optique spécifique, ce qui en fait un système relativement coûteux.

De récents systèmes se sont orientés sur les technologies neuronales. Celui d'EDS, présenté dans *[SHEP94]*, requiert beaucoup moins de calculs que les méthodes basées sur l'extraction des détails de l'empreinte, tout en conservant une grande fiabilité, notamment un taux de «Vrais Rejetés» très faibles, ce qui en fait un système très prometteur pour de futures applications destinées à un large public. En outre, la taille de la Référence ne dépassant pas 10 octets, on peut envisager ce type de procédé sur une architecture de faible capacité telle qu'une carte. Cette dernière approche sera reprise dans la partie portant sur les réseaux neuronaux.

II.3.1.2 La géométrie de la main

Déjà utilisé dans des applications à grande échelle telles que le contrôle à l'immigration aux USA (Aéroport JFK) *[RECO93]*, ce système biométrique possède une très intéressante propriété, celle de la compacité de la Référence, pouvant descendre jusqu'à 9 octets dans le cas de l'exemple cité ci-dessus.

Nous avons poursuivi des recherches et développements sur ce domaine afin de mettre en valeur le type de calculs demandés pour extraire des paramètres caractéristiques de la main pouvant servir à l'identification. Dans cette approche nous utilisons des éléments de géométrie plutôt que des méthodes de reconnaissance de formes globales afin de limiter les calculs dans leur complexité ainsi que la taille de la Référence. Ces deux conditions demeurent très importantes si l'on souhaite pouvoir intégrer l'identification dans la carte.

Le capteur biométrique utilisé consiste en une tablette graphique accompagnée d'un stylo magnétique. Ce dernier permet d'obtenir le contour de la main sous la forme d'une séquence de points 2D qui sont les données enregistrées à partir desquelles nous avons extrait des paramètres significatifs.

En pratique, si on utilisait un tel système biométrique destiné à une large application, la tablette devrait être remplacée par une caméra ou un scanner évitant ainsi à l'utilisateur toute manipulation inconfortable.

Notre recherche s'est orientée vers la mesure de la longueur des doigts, paramètres discriminants entre les individus. Le procédé le plus simple et rapide pour la mesure de telles longueurs est de faire une recherche des minima et maxima à partir de la courbe 2D des points acquis du contour. Ce procédé toutefois n'est pas très fiable car, dans ce cas, d'une faible variation de l'orientation de la main sur la tablette ou le scanner peut résulter une différence importante dans la position des minima et maxima par rapport à l'extrémité des doigts, comme indiqué sur la figure suivante:

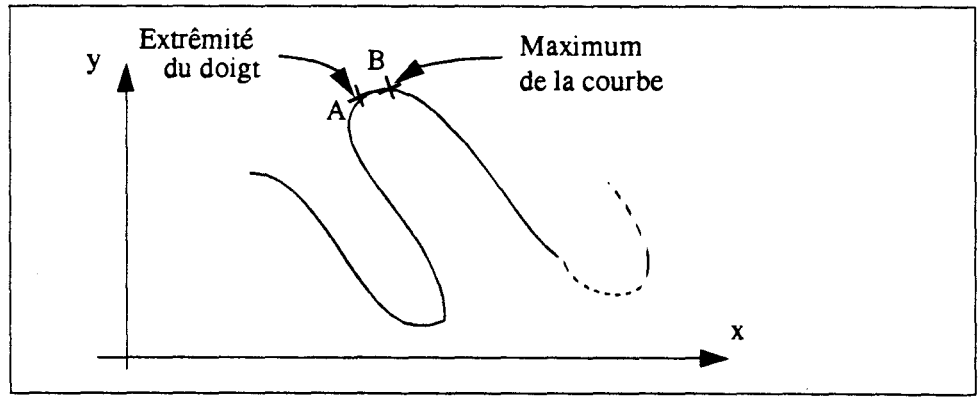


FIGURE 17 Ecart possible entre maximum local et extrémité du doigt

Dans le cas d'une main relativement inclinée par rapport à l'axe des ordonnées, on ne peut plus assimiler maximum local avec extrémité du doigt, sous peine d'obtenir une mesure de longueur très imprécise.

Nous avons donc ajouté quelques étapes de géométrie élémentaire pour assurer une mesure de longueur fiable indépendamment de l'inclinaison de la main (en supposant que l'on donne une liberté d'inclinaison possible de la main pouvant atteindre environ 30 degrés).

Une première étape consiste à rechercher dans la séquence de points les maxima et minima pour avoir les extrémités approximatives des doigts, comme précédemment. On va alors rechercher l'orientation du doigt considéré en traçant tout d'abord une droite horizontale environ 1 cm au dessous du maximum local de la courbe (point A sur la courbe suivante) et une autre environ 1 cm au dessus de l'un des deux minima entourant le maximum A.

Le schéma suivant retrace les constructions géométriques simples associées à la mesure de la longueur d'un doigt (autre que le pouce pour le moment):

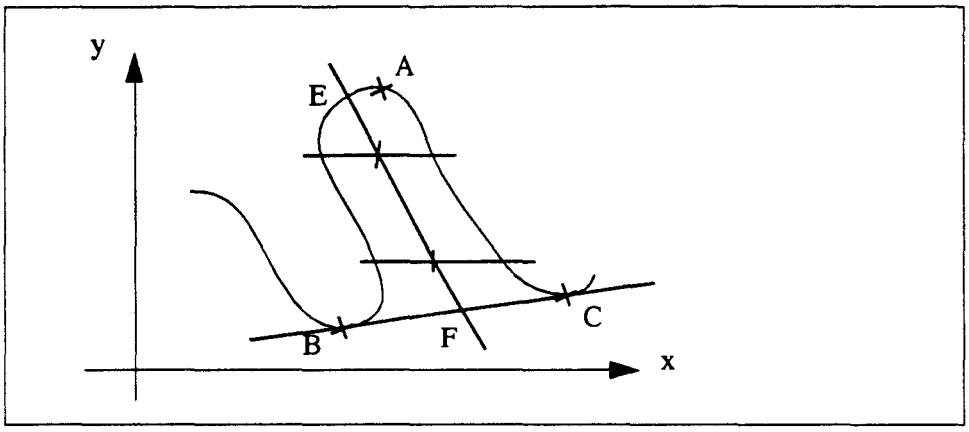


FIGURE 18 Mesure de la longueur d'un doigt indépendamment de l'orientation

Les deux droites horizontales coupent le contour pour former deux segments dont les deux milieux permettent d'obtenir la droite directrice de l'orientation du doigt. Cette direction reste relativement précise même en donnant différentes positions de la main laissant ainsi à l'utilisateur une marge de positionnement de sa main. La longueur du doigt sera alors représentée par la mesure du segment [EF], où E est le point d'intersection de la droite directrice du doigt avec le contour et F désigne l'intersection de cette droite directrice avec la droite formée des deux minima B et C (voir la figure).

Cet exemple de traitement demeure suffisamment simple pour envisager de le réaliser dans la carte. De plus, le fait de ne stocker sous forme de Référence que quelques paramètres tels que les longueurs de doigts permet de conserver une taille

de données très réduite. On peut imaginer de la même manière stocker d'autres critères caractéristiques et invariants, comme la largeur de la main ou encore la largeur des doigts à certains endroits précis (par exemple à 1/3 et 2/3 de la longueur calculée précédemment).

Extraire davantage de paramètres suppose une augmentation de la taille des traitements et de la Référence, mais permet d'améliorer la fiabilité de reconnaissance. Il appartient à l'utilisateur de déterminer quelle complexité et taille de carte il désire en fonction du niveau de sécurité requis par son application.

II.3.1.3 Le fond de l'oeil

L'idée réside dans la reconnaissance de l'image rétinienne, unique pour chaque personne, même dans le cas de jumeaux identiques. Elle suppose un matériel coûteux pour l'acquisition des données. L'utilisateur doit regarder au travers d'un scanner optique et fixer attentivement une cible de façon à ce que la caméra relève à chaque fois la même portion de l'image rétinienne. Cette mise en place pas très confortable donne néanmoins une identification excellente et fait de ce système incontestablement le plus sûr et le moins facilement fraudable des systèmes biométriques.

Nous n'avons pas réalisé de travaux sur ce procédé étant données la forte puissance de calcul et place mémoire requises, incompatibles avec la faible capacité de la carte, et surtout l'importance du matériel nécessité par ce genre d'approche. Le lecteur peut néanmoins se reporter au système d'EyeDentify *[EYED93]* pour de plus amples informations. Outre le fait qu'un tel système soit couteux si on l'envisage pour une application destinée à un large public telle qu'une application bancaire (contrôle d'accès sur carte de crédit), il est inadapté d'un point de vue simplicité d'utilisation et inconfortable dans la capture de l'image rétinienne. Ce dernier aspect sera repris dans la partie évaluation des systèmes.

II.3.1.4 L'identification du visage

Deux types d'approches ont été étudiées pour l'identification du visage:

- La reconnaissance de formes globale utilisant des méthodes de traitement d'images comme la FFT (Fast Fourier Transform) ou les réseaux de neurones
- L'extraction de paramètres invariants pouvant caractériser un visage

La première catégorie demande un environnement puissant pour répondre aux nombreux calculs nécessaires au traitement des pixels décrivant un visage. Des systèmes de ce type sont décrits dans *[NEUR93]* ou *[BOUA92]* (à base de réseaux de neurones) ou encore *[PENT94]*, où il est tenu compte des changements d'éclairement et des divers angles d'orientation du visage.

La seconde nous intéresse davantage car elle est moins consommatrice dans la mesure où seulement quelques paramètres sont comparés dans la phase d'identification, après un traitement préalable servant à extraire quelques points caractéristiques. Par exemple, dans *[KAME93]*, on trouve une méthode d'identification du visage qui utilise uniquement 7 points du visage pour pouvoir faire la différence entre 80 individus.

Les points prélevés peuvent être les suivants:

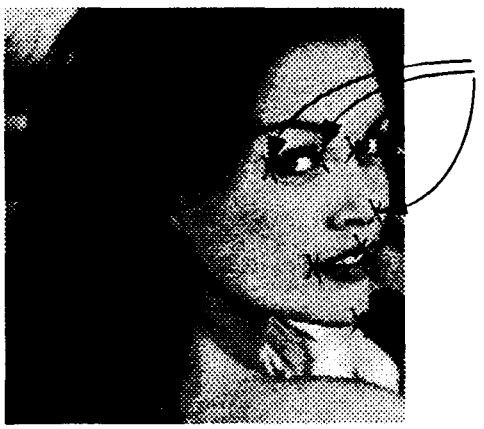


FIGURE 19 Exemple de points utiles à la reconnaissance du visage

Afin de pouvoir effectuer une reconnaissance relativement fiable quelle que soit l'orientation du visage (on supposera que celui-ci reste orienté par un axe vertical mais qu'il peut pivoter de côté jusqu'à environ 45 degrés), de nombreuses méthodes ne calculent pas par exemple l'écartement entre les yeux (qui varie en fonction de l'orientation) mais quelques rapports de distances bien choisis, comme le montre le schéma suivant:

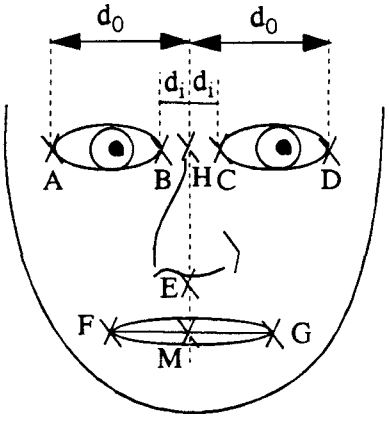


FIGURE 20 Calcul de ratios faisant intervenir les différentes distances

Un exemple de rapport de distances indépendant de l'orientation peut être:

$$R(A,B,C,D) = \frac{\left(d_0 + d_i\right)^2}{4 \times d_0 \times d_i}$$

Ce type de paramètre peut être complété par quelques autres mesures plus difficiles à effectuer, comme la courbure du menton ou la position des oreilles *[KAME93]*.

Nous n'avons pas effectué de développement spécifique à ce type d'identification biométrique car elle nécessite la mise en oeuvre de techniques complexes d'extraction des points caractéristiques à partir d'images obtenues par caméra vidéo. En revanche, on peut noter que le calcul des ratios nécessaires à l'identification et le stockage des paramètres restent à la portée des cartes à puce en termes de calculs et de place mémoire. On peut ainsi imaginer un traitement partiel des données à l'extérieur de la carte, puis le calcul des ratios et de la distance d'identification à l'intérieur de celle-ci. Le lecteur peut se reporter à *[MANJ92]* ou *[KAME93]* pour de plus amples informations sur la reconnaissance du visage.

II.3.2 Les systèmes comportementaux

II.3.2.1 La dynamique de la signature manuscrite

Si certains systèmes de reconnaissance de la signature manuscrite se sont attachés à la recherche de critères géométriques et à la description syntaxique de la signature (positionnement de différentes parties de la signature les unes par rapport aux autres) *[AMMA88]*, la plupart considère désormais davantage les aspects dynamiques invariants enregistrés pendant la saisie de la signature *[PLAM88] [LAMA84]*. Il peut s'agir de la durée totale ou de la vitesse d'exécution, ou encore de l'accélération en certains endroits bien choisis. Il est à noter que certains systèmes utilisent à la fois l'aspect statique et dynamique *[LORE84]*.

Nous nous sommes plus particulièrement intéressés aux techniques dynamiques car elles possèdent l'avantage d'être difficilement imitables.

L'ensemble des données représentant une signature s'assimile en général à une séquence de points 2D enregistrés à l'aide d'une tablette graphique et d'un stylo. Un paramètre supplémentaire peut être la prise en compte de la pression du stylo sur la tablette, donnant ainsi une série d'éléments munis d'une abscisse, une ordonnée et un niveau de pression (qui sera en général représenté graphiquement sous la forme de 256 niveaux de gris *[AMMA86]*). Ces entités (x,y,P) capturées à des intervalles de temps réguliers représentent les échantillons du signal provoqué

par la signature. A partir de ces données de base on peut facilement calculer par dérivation les séquences de vitesse et d'accélération.

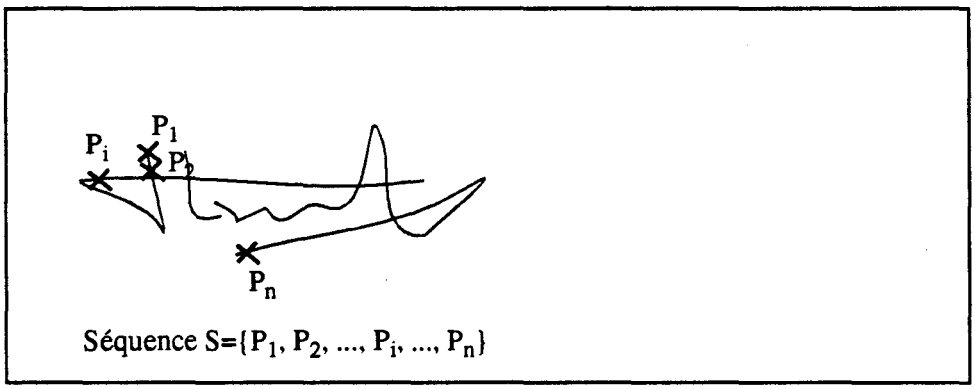


FIGURE 21 Acquisition d'une signature

Une étude comparative sur l'étude respectivement des séquences de position, de vitesse et d'accélération comme moyen de vérifier une signature manuscrite est décrite dans *[PLAM88]*. Elle a été effectuée au moyen de plusieurs types d'algorithmes couramment utilisés pour faire correspondre un signal de Référence à un signal de Test, comme par exemple l'alignement dynamique où la comparaison utilisant des arborescences. Il semblerait que les parties du signal correspondant à l'écriture verticale et représentées dans l'espace des vitesses soient les plus discriminantes.

Un exemple de vérification de la signature manuscrite figure à la fin du chapitre IV. Il souligne le fait qu'un tel procédé d'identification ne requiert que des fonctions de manipulation simples sur des données séquentielles dont la structure est elle-même simple (une séquence de points), de ce fait compatible avec les possibilités des cartes.

II.3.2.2 La dynamique de la signature clavier

Evoquée dans *[CART94]*, la signature clavier consiste à identifier un individu par sa façon de taper sur un clavier. La plupart des systèmes utilisent le temps de frappe entre deux touches consécutives. La phase de constitution de la référence demande à l'utilisateur de taper une même phrase plusieurs fois, comme par exemple «Patrick George vous souhaite une bonne journée». Bien que simple à mettre en oeuvre, ce procédé biométrique possède toutefois un véritable inconvénient: il suppose que les personnes qui l'utilisent soient habituées à la frappe sur un clavier, empêchant ainsi toute application destinée à un large public.

Cet inconvénient majeur peut être résolu si l'on s'intéresse à d'autres critères que le temps de frappe entre deux touches. Nous avons effectué des recherches

sur ce sujet. L'implémentation complète d'un système de ce type est détaillée dans le chapitre IV. Elle possède de nombreux avantages par rapport à la majorité des systèmes biométriques présentés dans ce chapitre répondant aux contraintes d'intégration dans la carte.

II.3.3 Systèmes utilisant les deux aspects

II.3.3.1 La reconnaissance de la voix

L'aspect principal utilisé ici concerne les caractéristiques propres à une voix et non pas la reconnaissance de la parole (l'analyse du texte énoncé). Les cordes vocales d'un individu permettent en général de l'identifier. A ces caractères physiologiques on peut rajouter des critères plus comportementaux tels que l'accent où l'état de la personne au moment où elle parle (essouflement, stress, etc...).

Nous avons étudié la reconnaissance vocale à l'aide de deux systèmes utilisant des approches différentes: le premier utilise du traitement de signal relativement classique et a été développé par Zi plc *[ZIPL92]*, le second utilise une approche originale basée sur l'outil TESPAR (Time Encoded Signal Processing and Recognition) conçu par Domain Dynamics Ltd *[DOMA94]* et sur la technologie neuronale. Non seulement les technologies utilisées sont différentes mais aussi les manières d'effectuer la constitution de la Référence (interface utilisateur) qui demeurent essentielles dans la mise en place de systèmes biométriques.

1. Le système de Zi plc

Basé sur la prononciation de 3 ou 4 mots choisis au hasard parmi un vocabulaire de 250 mots (pour une langue donnée), cette technique d'identification repose sur un échantillonnage du signal vocal à une fréquence de 9.6 Khz. Pour chacun des mots prononcés, 84 trames de 300 échantillons sont enregistrées. La condition de début d'acquisition des données implique que 50 échantillons parmi 200 aient une amplitude supérieure à un seuil fixé. Les différentes étapes de traitement des données sont les suivantes:

- Calcul de 11 coefficients d'autocorrelation pour chaque trame
- Calcul d'un coefficient d'énergie pour chaque trame
- Ces coefficients étant stockés sur 16 bits, la taille mémoire nécessaire au stockage du Template total (4 mots) est de l'ordre de 8 Koctets.

- Afin de faire correspondre correctement les trames deux à deux lors de la comparaison de la Référence avec le Test, un calcul d'alignement dynamique est utilisé *[RABI78]*.

- Pendant la phase de comparaison, seules les trames possédant une énergie suffisante sont utilisées. La distance entre le Test et la Référence est calculée par la méthode «d'Itakura» nécessitant le calcul des coefficients de prédiction linéaire, eux mêmes obtenus à partir des coefficients d'autocorrélation.

Le fait de choisir au hasard les mots à prononcer pour l'authentification permet de limiter la possibilité d'enregistrement préalable de la voix sur un magnétophone. De plus ce système est doté d'un combiné téléphonique spécifique dont l'acoustique est modifiée suivant la façon de le tenir. Ce capteur approprié renforce la sécurité d'identification. D'autres paramètres sont également pris en compte: la tolérance de reconnaissance diminue avec l'habitude d'utilisation. Des tests ont en outre marqué un fonctionnement normal en cas de mal de gorge.

2. L'approche basée sur TESPAR

Contrairement à la plupart des analyses de signaux, l'approche de TESPAR n'effectue pas une transformation du signal temporel en un signal dans le domaine des fréquences (comme la Transformée de Fourier). Le signal va être décrit sous la forme d'une séquence d'éléments pris dans un alphabet constitué de 29 codes.

Soit s(t) le signal étudié. s peut s'écrire sous la forme:

$$s(t) = C + \cos(wt) - \frac{1}{3}\cos(3wt) + ...\text{etc}...$$

En admettant que $\cos(wt) = \frac{1}{2}\left(e^{jwt} + e^{-jwt}\right)$

$$\sin(wt) = \frac{1}{2j}\left(e^{jwt} - e^{-jwt}\right)$$

et en remplaçant e^{jwt} par x, s s'écrit:

$$s(x) = C + \frac{1}{2}\left[x + \frac{1}{x}\right] - \frac{1}{6}\left[x^3 + \frac{1}{x^3}\right] = \frac{1}{6x^3}\left[-x^6 + 3x^4 + 6Cx^3 + 3x^2 - 1\right]$$

Les éléments qui constituent l'alphabet utilisé dans TESPAR sont déterminés à partir des racines de s(x) (c'est-à-dire en résolvant s(x)=0). La courbe suivante donne un exemple de la position des zéros obtenus sur un signal s:

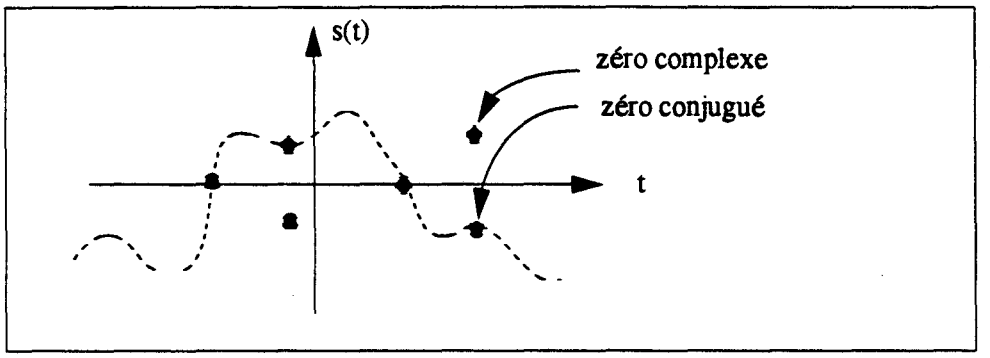


FIGURE 22 Zéros complexes issus de l'étude de s.

A partir de ces valeurs sont établis les éléments de l'alphabet, formant au total 29 symboles. Le profil d'un candidat sera alors constitué par des matrices qui représentent les corrélations des données enregistrées dans la séquence. Dans le cas de 1 dimension, il s'agit des fréquences d'apparition de chacun des 29 symboles. Dans le cas d'une étude à 2 dimensions, la matrice 2D représente les fréquences d'apparition de 2 symboles consécutifs. On peut de la même façon construire des matrices retraçant les corrélations d'ordre quelconque. Ce type d'analyse sera repris dans le chapitre IV sur un autre exemple d'identification biométrique.

La phase d'identification consiste alors à fournir tout ou partie de ces matrices en entrée d'un réseau neuronal qui donnera en sortie la décision de reconnaissance ou non d'un individu (voir chapitre IV sur les réseaux neuronaux).

D'un point de vue pratique, pour l'apprentissage du système, l'utilisateur fournit plusieurs échantillons de sa voix au travers de la prononciation répétitive de la même phrase, choisie de façon à couvrir correctement le spectre vocal. La phrase qui a été retenue pour l'évaluation du système était: «My name is Charles Westlake» *[DOMA94]*.

Dans l'optique de réaliser un système de reconnaissance vocal dans la carte à micro-processeur, la seconde approche est beaucoup plus adaptée aux contraintes engendrées par l'intégration du traitement biométrique. En effet, la première approche nécessite des calculs importants que ce soit pour l'obtention des coefficients d'auto-corrélation ou le calcul de la distance entre la Référence et le Test. De plus, la taille de la Référence à stocker est importante par rapport à la place disponible sur la carte.

En revanche, la seconde approche est nettement moins consommatrice en calculs et place mémoire. Elle possède en outre une particularité intéressante: on peut séparer l'algorithme en deux parties successives presque indépendantes, en

ayant d'un coté le traitement de l'outil TESPAR pour obtenir une codification du signal vocal et de l'autre coté le réseau de neurones pour la prise de décision de l'identification. Cette caractéristique offre la possibilité de n'effectuer dans la carte que la partie neuronale sans compromettre la sécurité de l'identification (voir *[CASC94]*).

La Référence gardée dans la carte et qui retrace l'identité de l'individu consiste alors simplement à stocker les coefficients des poids du réseau neuronal (voir l'introduction aux réseaux neuronaux dans le chapitre IV) qui représentent un volume d'information inférieur à celui de l'approche de Zi plc. Nous verrons dans le chapitre IV que ce type d'approche peut se généraliser à d'autres systèmes biométriques que l'identification vocale.

II.4 Evaluation des systèmes biométriques

Sur la base de plusieurs documents décrivant des systèmes biométriques et de comparaisons réalisées dans le cadre d'études extérieures (se reporter à *[HOLM91]*, *[SHER92]* et *[CASC94]*), nous allons dans cette partie résumer quelques résultats quant à la fiabilité et à l'efficacité des différents systèmes biométriques disponibles sur le marché. Pour cela nous allons reprendre comme critères d'évaluation ceux évoqués dans la partie concernant les définitions de ce chapitre.

Le tableau suivant résume la comparaison de plusieurs systèmes conventionnels d'un point de vue des performances en fonction de la simplicité de mise en oeuvre et du coût de revient approximatif *[CASC94]*.

Système biométrique (Etude de 1992)	TFA/ TVR	Coût	Acceptation des utilisateurs	Capteur biométrique
Géométrie de la main	moyen	$250	bonne	Scanner optique
Empreintes digitales	bon	$750	moyenne	Scanner optique
Fond de l'oeil	excellent	$1000	faible	Scanner optique
Dynamique de la signature	moyen	$100	bonne	Tablette à digitaliser
Signature vocale	moyen	$50	bonne	Microphone

FIGURE 23 Comparaison générale de divers procédés biométriques

De la même façon, on peut comparer les critères techniques associés à l'identification:

Système biométrique	Temps de constitution du Template (en min)	Temps de vérification (en sec)	Taille du Template (en octets)
Géométrie de la main	1-2	<1-3	10-1000
Empreintes digitales	1-2	2-5	200-1500
Fond de l'oeil	1-2	1-2	40-256
Dynamique de la signature	1-2	1-3	100-3500
Signature vocale	1-3	1-2	60-2000

FIGURE 24 Comparaison technique des procédés biométriques

De ces résultats découlent plusieurs remarques quant aux avantages et inconvénients des différents systèmes:

- On peut difficilement envisager la reconnaissance du fond de l'oeil pour des applications destinées à un large public, principalement en raison de sa mauvaise

acceptation auprès des utilisateurs. Cependant, pour des applications à haut degré de sécurité, c'est le seul qui garantisse une identification excellente.

- La signature vocale est intéressante pour une éventuelle application demandant une identification à distance: c'est la seule répondant à ce critère. C'est aussi de ces systèmes celui qui possède le capteur le moins coûteux (seulement un microphone). On peut néanmoins redouter l'évolution des techniques de l'audio numérique: si pour le moment certains tests ont montré qu'un enregistrement ne pouvait suffir à la reconnaissance, les enregistrements numériques risquent tôt ou tard de reproduire une voix de façon si fidèle qu'ils pourraient gêner ce type de reconnaissance.

- La géométrie de la main possède l'avantage d'être bien perçue par les utilisateurs, contrairement à d'autres systèmes comme les empreintes digitales qui traditionnellement ont été associées aux enquêtes criminelles. De plus la taille des données constituant la Référence (paramètres géométriques de la main) est en générale réduite. Cet avantage n'est pas négligeable si l'on envisage une implémentation sur une architecture réduite comme une carte.

- Les techniques comportementales comme la dynamique de la signature sont attirantes par le fait qu'elles paraissent plus difficiles à reproduire que les aspects physiologiques. Les taux TFA et TVR sont à l'heure actuelle encore insuffisants pour garantir une très bonne identification dans le cas de très larges applications (les applications bancaires par exemple).

Conclusion

Si aucun de ces systèmes n'a pour l'instant prédominé sur le marché, c'est surtout parce qu'ils ont tous leurs avantages et leurs défauts. Il appartient alors à l'utilisateur de choisir un système biométrique en fonction de ses besoins d'identification. Cependant, les optimisations possibles sont encore nombreuses et dépendent de la technologie utilisée pour chacun de ces systèmes.

CHAPITRE III *L'Identification Comportementale*

Introduction

Dans le chapitre précédent nous avons décrit et évalué des systèmes biométriques permettant d'identifier un individu par «ce qu'il est». Au delà des systèmes basés sur des critères purement physiologiques, nous avons également abordé ceux utilisant des aspects plus comportementaux (la dynamique de la signature par exemple).

Il convient de différencier l'aspect «biométrie comportementale» étudié précédemment avec le concept «d'identification comportementale» que nous introduisons dans la suite de ce chapitre. Dans le premier cas, il s'agit de reconnaître un individu par son automatisme à réaliser une tâche particulière, telle qu'une signature. Dans le second cas, il s'agit d'identifier un individu par ses pratiques reflétant des habitudes (par exemple reconnaître sa manière de travailler sur un ordinateur, qui peuvent être notamment: ses horaires typiques de connexion par jour, le nombre d'ouvertures de fichiers ou autres critères caractéristiques).

Après la définition de quelques notions associées à l'identification comportementale, nous détaillons dans une seconde partie les travaux de recherche que nous avons menés autour de ce concept, notamment au travers d'un exemple d'identification comportementale appliqué aux systèmes de paiement par carte bancaire, que nous avons baptisé «Radar».

III.1 Définition de quelques notions

III.1.1 L'Identification Comportementale

L'**Identification Comportementale** d'un individu vise à caractériser une personne par l'étude de ses habitudes, de ses pratiques en réponse à des situations particulières qui lui sont propres. Caractériser son comportement peut résider par exemple dans sa façon de conduire, sa façon de dépenser quotidiennement de l'argent ou encore sa manière de travailler sur un ordinateur (comme évoqué précédemment).

Bien qu'il soit délicat de déterminer quels sont les critères qui modélisent un comportement et de ce fait de réaliser un système d'identification fiable, ces nouvelles techniques présentent un certain intérêt par rapport aux systèmes plus classiques d'identification biométrique et autres contrôles d'accès:

- Une composante biométrique (la voix par exemple) est probablement plus imitable qu'un comportement: un enregistrement de très haute qualité numérique peut parfois venir perturber un sytème vocal d'identification, tout comme un masque pourrait imiter un visage dans le cadre de reconnaissance visuelle. En revanche, un comportement reste difficile à identifier, et donc à reproduire, à moins de connaitre ou d'observer l'individu en question pendant une relativement longue période. L'individu lui-même ne connaît pas très bien son comportement dans de nombreuses circonstances (il peut avoir une vague idée de sa façon de conduire ou de travailler, mais n'en connaît certainement pas toutes les caractéristiques).

Nous avons avancé des recherches sur ce sujet encore peu exploré, en tenant compte des lourdes contraintes imposées par la carte (capacité mémoire et puissance de traitement réduites) en vue d'une implémentation de systèmes d'identification comportementale appliquée au domaine cartes à puce.

Quelques travaux extérieurs ont étudié certains de ces aspects:

- Le projet NIDES (Next-generation Intrusion Detection Expert System) mené au SRI (Stanford Research Institute) s'intéresse à la détection d'intrusions sur les ordinateurs en observant et mesurant des caractéristiques liées aux habitudes des utilisateurs de Stations de Travail, telles que le temps de connexion par jour, les horaires de connexion etc...
  La description complète du système se trouve dans *[ANDE93]* et *[JAGA93]*.

- La mise en place par American Express d'un Système Expert dédié à l'automatisation de 90% environ du processus de vérification des transactions bancaires, "The Authorizer's Assistant" (voir *[PIKE87]*), a pris en compte des informa-

tions relatives au "niveau de vie" de leurs clients pour déceler plus précisément les transactions frauduleuses.

- Neuralware a également concu un système de détection de fraude à base de réseaux neuronaux pour les transactions bancaires de la Chase Manhattan *[KEYE92]*.

III.1.2 Le concept de «Radar»

Depuis le début de la carte à puce la plupart des recherches associées à la sécurité ont été axées sur l'amélioration du contrôle d'accès. L'intérêt est bien entendu d'empêcher une personne non autorisée d'accéder aux ressources de la carte. Cependant, un tel système peut faillir quelle que soit son efficacité étant donnés des facteurs de négligence ou de violence pouvant intervenir (un PIN code noté au dos d'une carte de crédit ou dans un agenda, une agression pour prendre possession de la carte et de son mot de passe demandé lors du contrôle d'accès...).

De façon analogique, la sécurité d'un bâtiment ou d'une maison est généralement assurée par des serrures ou verrous sur les portes d'accès. De temps en temps cette sécurité ne suffit pas, c'est pourquoi on utilise alors des radars pour détecter et observer les éventuels intrus.

Nous avons voulu implémenter cette notion de Radar dans la carte. L'objectif est de reconnaître le comportement de «l'utilisateur de la carte» et d'éventuellement refuser le service requis si ce comportement paraît anormal et suspect. A l'opposé d'un verrou (PIN code pour la carte par exemple) qui est indépendant de l'application, le Radar est étroitement lié à l'application.

Le Radar ne cherche pas à remplacer le contrôle d'accès. Il s'ajoute au système pour renforcer le schéma de sécurité déjà présent.

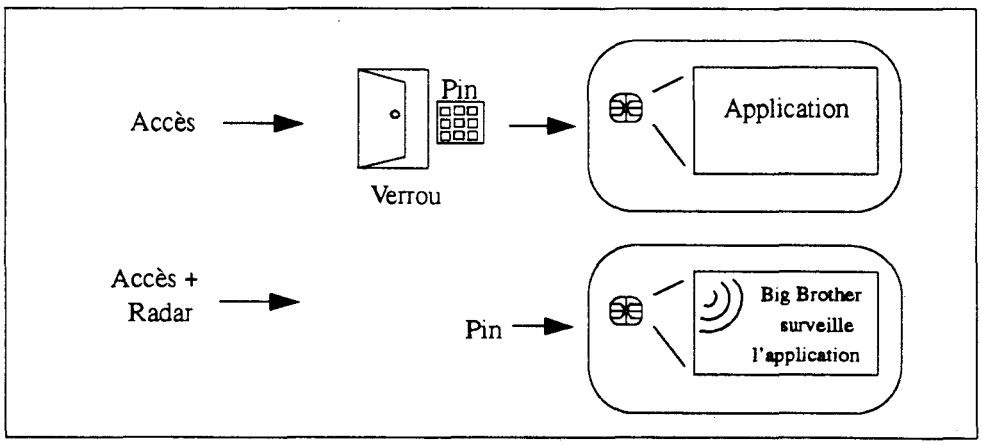


FIGURE 25 Positionnement du Radar dans le schéma de sécurité

L'approche ensembliste

Dans cette approche l'individu fournit des informations qui sont traitées par des prédicats dont le résultat est **vrai** ou **faux**.

Tout comme pour les systèmes biométriques, on peut imaginer soit un système aveugle, c'est-à-dire qui doit identifier au sein d'une base de données un individu particulier parmi un grand nombre, soit une solution avec référence qui doit vérifier l'appartenance d'un Test à sa Boule de Référence.

Soit L une liste d'informations $L=(I_1, I_2, ..., I_n)$ fournie comme échantillon et soit par ailleurs un ensemble de faits connus relatifs à l'individu. $F=(F_1, F_2, ..., F_m)$. Soit enfin un ensemble R de règles liant les faits et les éléments de la liste L et liant des résultats de règles les uns avec les autres.

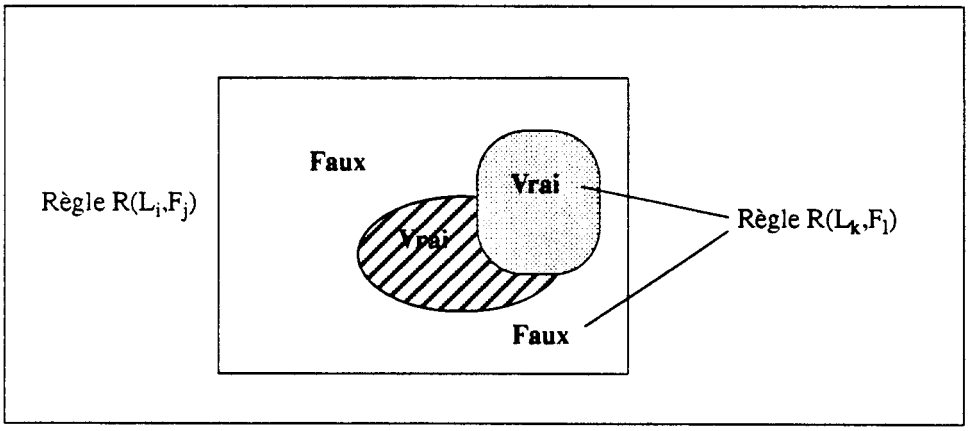


FIGURE 26 Construction de règles sous forme ensembliste

L'écriture des règles peut se faire dans un langage d'intelligence artificielle sous la forme:

SI REGLE-1 ET REGLE-2 = VRAI ALORS...

Dans la partie qui suit nous avons expérimenté un exemple de Radar utilisant cette approche. Il s'agit d'un Radar appliqué aux systèmes de transactions bancaires.

III.2 Description d'un Radar appliqué aux systèmes de transactions par carte de crédit

III.2.1 Mise en place du Radar

Nous avons étudié l'identification comportementale des individus au travers d'un exemple concret consacré aux transactions par carte de crédit *[ALEX94a]*.

Dans notre approche nous avons caractérisé une transaction par 4 éléments:

- **Nature**, choisie parmi les huit suivantes: Cash, Transports, Essence, Nourriture, Logement, Ameublement, Loisirs, et Autres.
- **Montant**: Un nombre réel positif.
- **Date**, explicitée sous la forme Jour/Mois/Année.
- **Lieu**, par exemple un code postal.

A partir de ces données, on peut construire un ensemble de règles pour décrire les habitudes d'un individu utilisant sa carte de crédit comme moyen de paiement. Pour M. Dupont, il s'agirait par exemple des règles suivantes:

1. M. Dupont ne fait quasiment jamais de retrait d'argent (dans un distributeur de billets) dont le montant dépasse 2000 FF.
2. Le montant total de ses retraits d'argent durant la même semaine excède rarement la somme de 2500FF.
3. Ses paiements de type Transports ne vont généralement pas au delà de 4500 FF par semaine.
4. La plupart du temps, M. Dupont effectue ses retraits d'argent liquide dans la ville où il habite (Paris) ou à Marseille (où vit sa famille).
5. Chaque semaine il paie avec sa carte approximativement le même loyer de l'ordre de 2500FF.
6. etc...

Cet ensemble de règles ne couvre pas tous les cas de figures mais peut être considéré comme le noyau de base caractérisant le comportement de M. Dupont. Bien entendu, ces règles ne sont pas toujours vérifiées: il se peut que M. Dupont voyage et soit confronté à des dépenses inhabituelles.

D'après l'observation d'un ensemble de transactions <Nature, Montant, Date, Lieu>, on peut extraire un jeu de règles écrites sous la forme ensembliste que nous avons évoquée dans l'introduction du concept de Radar:

if (*Prédicat*) then - *Transaction normale*

- *Transaction suspecte*

- *Transaction refusée*

Par exemples:

- If (Montant > 2000) then *Transaction suspecte*
- If (Location<>Paris & Location<>Marseille) then *Transaction suspecte*

La base de connaissance résultante, comportant les habitudes de M. Dupont, peut s'assimiler à celle présentée ci-dessous:

Nature	Montant/ Transaction	Montant/ Semaine.	Lieu
Cash	**<2000**	**<2500**	**cf Loc-Table**
Transports	**<3000**	**<4500**	**--**
Essence	**<250**	**<700**	**--**
Nourriture	**<800**	**<1500**	**--**
Logement	**<2500**	**<3000**	**cf Loc-Table**
Ameublement	**--**	**<2000**	**--**
Loisirs	**<1000**	**--**	**cf Loc-Table**
Autres	**<10000**	**--**	**--**

FIGURE 27 Base de connaissances associée à M. Dupont

Dans le tableau donné précédemment, Loc-Table représente une Table décrivant les endroits où M. Dupont effectue ses transactions le plus souvent, comme par exemple:

Indice de fréquence	Lieu
10	**Paris**
3	**Lille**
3	**Marseille**
2	**Lyon**
2	**Nice**
1	**Bordeaux**
1	**Aix**
1	**Brest**

FIGURE 28 Exemple de Loc-Table

L'indice de fréquence s'établit à partir de statistiques d'enregistrement des transactions.

Outre ces deux tables, la carte doit contenir l'historique de toutes les transactions effectuées par l'utilisateur. C'est déja le cas pour les cartes de crédit. Une Table *Historique* peut être de la forme suivante:

Nature	Montant	Date	Lieu
Cash	**200**	**01/03/94**	**Paris**
Logement	**2300**	**01/10/94**	**Paris**
Cash	**500**	**01/24/94**	**Lille**
Loisirs	**300**	**01/26/94**	**Aix**
Autres	**1300**	**01/26/94**	**Aix**
...			

FIGURE 29 Exemple de Table «Historique»

Avant de rajouter une transaction nouvellement acquise dans sa base de données, le système procède à une expertise pour vérifier si elle paraît normale ou frauduleuse. Dans ce dernier cas, plusieurs actions peuvent être conduites à ce stade:

- Lancer une procédure on-line auprès d'un organisme habilité pour vérifier que la carte n'a pas été déclarée perdue ou volée.
- Refuser la transaction
- N'accepter que certaines transactions dites de première nécessité en cas de suspicion forte.

- Accepter la transaction tout en gardant trace de l'alarme générée pour cette transaction (en incrémentant un compteur par exemple) de façon à sanctionner plus sévèrement l'utilisateur dans le cas d'une autre transaction suspecte ultérieurement. Il s'agit en fait d'augmenter la côte d'alerte.

Afin d'établir plus précisément et complètement les relations possibles entre les 4 entités <Nature, Montant, Date, Lieu>, nous avons considéré tous les types de règles, en les faisant intervenir une par une, puis deux par deux et trois par trois, et pour terminer les quatre à la fois.

En prenant les entités une par une, on peut évidemment générer quatre types de règles comme suit:

1. **Nature:** if (Nature) then Transaction refusée.

Exemple: if (Nourriture) then Transaction refusée.

2. **Montant:** if (Montant > 2000) then Transaction refusée.

3. **Date:** if not(Date_Inf < Date < Date_Sup) then Transaction refusée.

4. **Lieu:** if (Lieu <> Lieu_possible_de_la_Table-Loc) then Transaction suspecte.

Prises deux par deux, 6 types de règles peuvent apparaître:

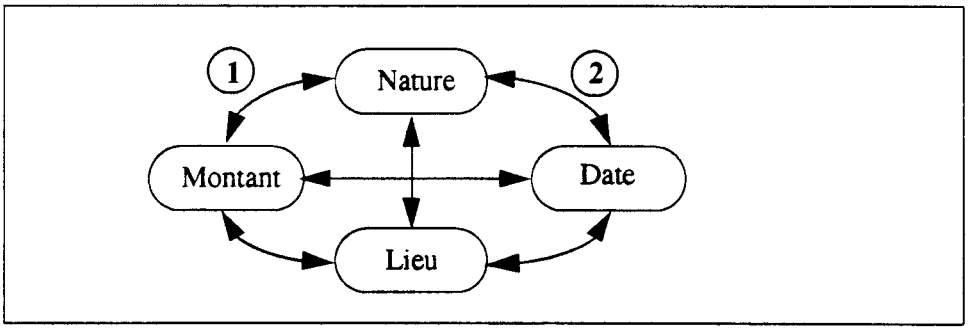


FIGURE 30 Règles possibles utilisant deux éléments

1. **(Nature, Montant):** if (Montant de type Nature > Montant_Seuil) then Transaction suspecte.

Exemple: if (Montant de Transport > 2000) then Transaction suspecte.

2. **(Nature, Date):** if (2 transactions de type Nature entre 2 Dates) then Transaction suspecte.

Exemple: If (2 paiements de Logement le même jour) then Transaction suspecte.

- Les règles **(Nature, Lieu)**, **(Montant, Date)**, **(Montant, Lieu)** et **(Date, Lieu)** sont décrites de la même façon.

Trois par trois, les 4 types de règles possibles s'écrivent d'une façon similaire au modèle suivant:

1. **(Nature, Montant, Date):** if (Montant de type Nature > Montant Seuil entre 2 Dates) then Transaction suspecte.

Exemple: if (Montant total de 2 retraits de Cash le même jour > Montant Seuil) then Transaction suspecte.

Enfin, le type de règle faisant intervenir les 4 éléments simultanément serait par exemple semblable à la règle précédente appliquée pour un certain endroit.

III.2.2 Intérêt de l'intégration du Radar dans la carte à puce

Depuis quelques années certaines institutions bancaires se sont attachées à élaborer des systèmes de détection de fraude utilisant des techniques de l'intelligence artificielle telles que Systèmes Experts *[DZIE89]* ou Réseaux de Neurones *[KEYE92]* dans le but d'automatiser en partie la vérification de leur transactions bancaires. Dans tous les cas l'expertise intervient de façon centralisée sur un serveur.

L'idée du Radar que nous avons décrit précédemment est de faire cette vérification de manière locale et décentralisée, ce qui apporte les avantages suivants:

- les techniques de détection de fraude font appel à des analyses sur des données se rapportant à la vie privée des individus. Dans le cas de détection sur un serveur central, l'individu n'a pas la garantie que les résultats obtenus sur son analyse de comportement ne seront pas utilisés à d'autres fins qui portent une atteinte à sa vie privée. Si la vérification se fait sur la carte qu'il porte, la divulgation de données le concernant reste limitée.

- la vérification peut être immédiate pendant la session d'utilisation de la carte par son porteur alors qu'elle est différée dans le cas d'un système central. Cette caractéristique permet notamment à la carte de devenir active par rapport à l'utilisation qu'on en fait.. Jusqu'à présent les cartes bancaires ne sont sécurisées

pour l'utilisateur que par le PIN code qui contrôle leur accès. Rajouter un détecteur de fraude propre au porteur renforce la sécurité du PIN qui seul s'expose à de plus en plus de fraudes car facilement transmissible.

Intégrer un système de type Radar dans la carte à micro-processeur suppose néanmoins de résoudre les difficultés d'implémentation liées aux lourdes contraintes de l'environnement carte en termes de puissance de calcul (processeurs 8-bits), place mémoire (256 octets de RAM au maximum) et donc de temps de réponse.

III.3 Evaluation du Radar

Cette partie s'intéresse à la mesure du degré de fiabilité d'un Radar en vue de son implantation dans une application de type bancaire.

Nous avons essayé de valider le concept de Radar sur la base de relevés de comptes existants permettant d'utiliser des chiffres authentiques pour une meilleure évaluation. Malheureusement, cette évaluation reste très limitée car nous n'avons pu tester le Radar que sur très peu de relevés de comptes pris dans notre entourage, de telles données étant personnelles et souvent confidentielles.

En effet, nous avons rencontré des experts de la fraude bancaire travaillant dans deux opérateurs financiers de portée internationale qui nous ont confirmé le caractère confidentiel et la non disponibilité des banques de données relatives aux transactions de leur clients.

Afin de simuler un potentiel fraudeur nous avons dans un premier temps mis en place un générateur de transactions aléatoires, l'idée première étant de considérer qu'un fraudeur détenant une carte avec son PIN code pouvait réaliser n'importe quel type de transaction. Bien entendu cette modélisation demeure approximative car en règle générale, un fraudeur n'agit pas de façon aléatoire mais calculée et réfléchie. Néanmoins, l'évaluation donne une idée de l'intérêt du Radar. Elle permet de confronter le caractère habituel des transactions d'un individu avec le côté aléatoire des transactions d'un fraudeur.

Pour enrichir le comportement du fraudeur, nous avons dans un second temps établi un jeu de transactions frauduleuses sur la base d'un sondage effectué auprès de notre entourage. Nous décrivons dans les points suivants les étapes de l'évaluation.

III.3.1 Mise en place de l'évaluation

Pour chaque individu nous avons pris en compte des transactions enregistrées pendant 6 mois et correspondant à des comptes bancaires réels. Une Table de Transactions (ou Historique), donnée en exemple ci-dessous, retrace l'activité de la carte bancaire d'un individu pendant 6 mois de façon à analyser son comportement sur une période suffisamment large:

Nature	Montant	Date	Lieu
Cash	**200 FF**	**02/12/93**	**Lille**
Loisirs	**174 FF**	**02/16/93**	**Lille**
Cash	**400 FF**	**02/17/93**	**Rouen**
etc...			

FIGURE 31 Exemple d'Historique de Transactions sur 6 mois

A partir de ces données récoltées sur 6 mois, nous avons pu établir une base de connaissances retraçant les habitudes de l'utilisateur concerné, comme suit:

Nature	Montant Maximum Autorisé/ Transaction	Montant Maximum Autorisé/ Semaine
Cash	**1000**	**1500**
Transports	**1000**	**1500**
Essence	**300**	**700**
Nourriture	**300**	**500**
Logement	**1000**	**2000**
Ameublement	**800**	**800**
Loisirs	**700**	**1000**
Autres	**500**	**800**

FIGURE 32 Base de connaissances associée à un utilisateur

La table des Lieus les plus fréquentés était formée de 2 éléments:

Priority Number	Location
100	**Lille**
7	**Rouen**

FIGURE 33 Loc-Table

III.3.2 Modélisation du Fraudeur

III.3.2.1 Générateur aléatoire

Cette simulation approximative de fraudeur a consisté à construire un jeu de transactions aléatoires respectant néanmoins certains intervalles:

Critères	Intervalles de choix
Nature	**probabilité identique pour chaque Nature**
Montant	**0 à 3000FF**
Fréquence	**0 à 10 le même jour**
Lieu	**50% pris dans la Loc-Table** **50% Inconnu**

FIGURE 34 Générateur aléatoire de transactions

Certaines améliorations auraient pu être apportées au générateur pour le rendre plus réaliste. Par exemple, il y a davantage de chances pour que le fraudeur effectue des transactions ayant des montants élevés. Nous avons gardé toutefois un modèle très simple de générateur aléatoire, car nous nous sommes très rapidement tournés vers une modélisation plus vraisemblable de fraudeur faisant intervenir des réactions humaines et réfléchies. C'est la modélisation par sondage que nous décrivons ci-après.

III.3.2.2 Modélisation par sondage

Le but du sondage était, pour chaque individu impliqué, de décrire les 10 premières transactions qu'il effectuerait s'il venait de voler une carte de crédit et en détenait le code d'accès. Cette manière de construire un jeu de transactions frauduleuses permet d'enlever le caractère très aléatoire de la solution précédente.

Nous avons de ce fait récolté un certain nombre de stratégies d'attaques et donc des jeux de transactions pouvant simuler le comportement du fraudeur de

façon plus précise que précédemment. Tous ces résultats sont discutés au chapitre IV lors de l'implémentation du Radar sous forme de Système Expert. Une étape supplémentaire consisterait à faire ce sondage sur un plus grand nombre de personnes choisies sur un éventail plus représentatif de la population mettant en valeur les différentes classes sociales.

III.3.2.3 Conclusion

Les résultats quant aux taux de détection correctes ou non de fraudes (évaluation du TVR et TFA décrits dans le chapitre II) dépendent de l'implémentation envisagée. Dans le chapitre IV nous aborderons deux implémentations possibles, l'une par Système Expert utilisant le langage Prolog, l'autre par les Réseaux Neuronaux, et discuteront les résultats obtenus.

L'identification comportementale en est à ses débuts. Elle fait appel à de nouvelles techniques informatiques (intelligence artificielle), psychologiques et sociologiques. Bien que souvent la preuve d'identification demeure plus difficile que les systèmes biométriques du fait de références plus imprécises, elle présente les avantages suivants:

- La simplicité technologique: elle manipule de l'information et n'exige donc pas de capteur.
- L'adaptation dynamique, par intégration dans la Référence de nouveaux échantillons correspondant à l'évolution du comportement.
- Une aptitude à supporter des niveaux de sécurité progressifs et imbriqués.
- Une capacité discriminante extensible par contrôle du nombre de faits et règles pris en compte.
- Une grande difficulté à produire des clones.
- La capacité éventuelle à détecter des comportements anormaux en cas de stress (agression).
- La capacité à traiter des données sécuritaires de type temporel telles que des dates limites, intervalles ou durées.

CHAPITRE IV *La manipulation des données dans la carte*

Introduction

Dans les 2 chapitres précédents nous avons souligné l'intérêt de structurer les données acquises pendant un processus d'identification. Ceci permet de développer des mécanismes globaux pour leur manipulation en automatisant les accès. La structure la plus couramment rencontrée est la séquence. Il peut s'agir d'une séquence d'entiers ou d'autres éléments plus élaborés tels que des points voire même des structures complexes (tables, arbres). A partir de ces observations et expérimentations nous proposons dans une première partie de ce chapitre une implémentation de structure de données dédiée au stockage des informations multimédia dans la carte.

La seconde partie du chapitre s'intéresse aux traitements associés à ces données. Nous avons exploré les deux voies suivantes:

1. Une approche par les langages de haut niveau.

2. Une approche par les réseaux de neurones.

Pour chacune des deux approches, nous avons envisagé trois types de traitements multimédia, qui ont fait l'objet de chapîtres précédents. Il s'agit de:

- **la compression des données**: le but du traitement est la compression **par la carte** d'une photo d'identité utilisant des méthodes non conservatives de la totalité des informations. La première approche concerne le standard de compression JPEG et consiste en la réalisation en langage C d'un algorithme de

compression dédié à la carte utilisant des simplifications de JPEG. La seconde approche étudie une compression par réseaux neuronaux. Auparavant nous aurons résumé les étapes et résultats essentiels à la compréhension de la technologie neuronale.

- **l'identification biométrique**: le traitement que nous avons expérimenté concerne une méthode originale de reconnaissance dynamique de la signature clavier. Dans un premier temps, nous décrivons le système et l'implémentons en langage C. Pour l'approche neuronale, nous utilisons des algorithmes classiques d'apprentissage supervisé et des méthodes d'auto-organisation ou d'apprentissage non supervisé.

- **l'identification comportementale:** nous avons implémenté cet aspect sous la forme de «Radar» appliqué aux systèmes de paiement par carte de crédit (concept introduit au chapître III). L'approche utilisant les langages de haut niveau concerne la réalisation d'un Radar en Prolog. Comme dans les cas précédents, le Radar sera également repris sous forme de réseau neuronal.

Les études détaillées de ces divers traitements débouchent sur l'établissement de primitives communes dédiées au traitement global de données multimédia dans la carte. Enfin, nous réécrirons des exemples applicatifs de traitements multimédia à partir des quelques primitives formulées.

L'organisation de cette partie peut ainsi se résumer par le schéma suivant:

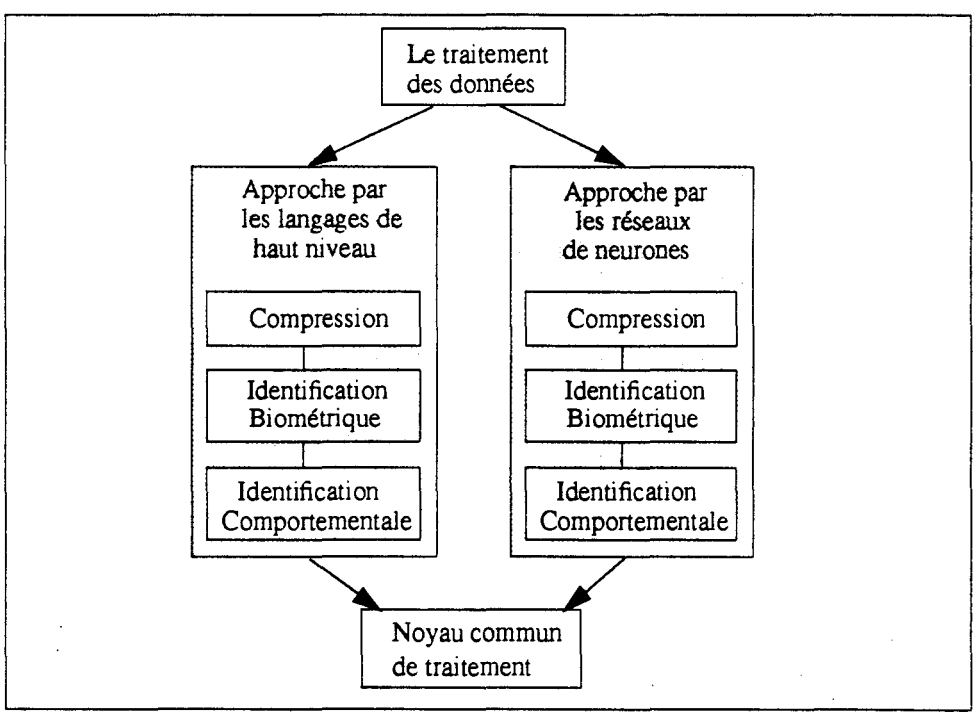


FIGURE 35 Approche envisagée pour le traitement des données

IV.1 La structuration des données adaptée à la carte

IV.1.1 Le manque de structuration des cartes actuelles

Contrairement aux progrès liés à la manipulation de structures de données devenues de plus en plus riches et complexes sur les micro-ordinateurs classiques d'aujourd'hui, la carte à micro-processeur en est restée à une représentation et organisation de ses données en mémoire sous une forme des plus primitives.

En effet, les seules cartes actuelles qui semblent intégrer un début d'organisation des données sous une forme structurée sont les deux suivantes, dont on détaille un exemple:

- La carte MCOS (Multi-applicative Chip Operating System) compatible avec la norme ISO 7816-3/4:
  Conçue par Gemplus Card International, cette carte offre une gestion de la mémoire de bas niveau similaire à MS-DOS, c'est-à-dire sous forme de fichiers de taille variable contenus dans des répertoires dont l'accès est contrôlé par un bloc de sécurité. La structure de la carte MCOS , détaillée dans *[CARO94]*, se limite ainsi à la définition d'une zone mémoire par application.

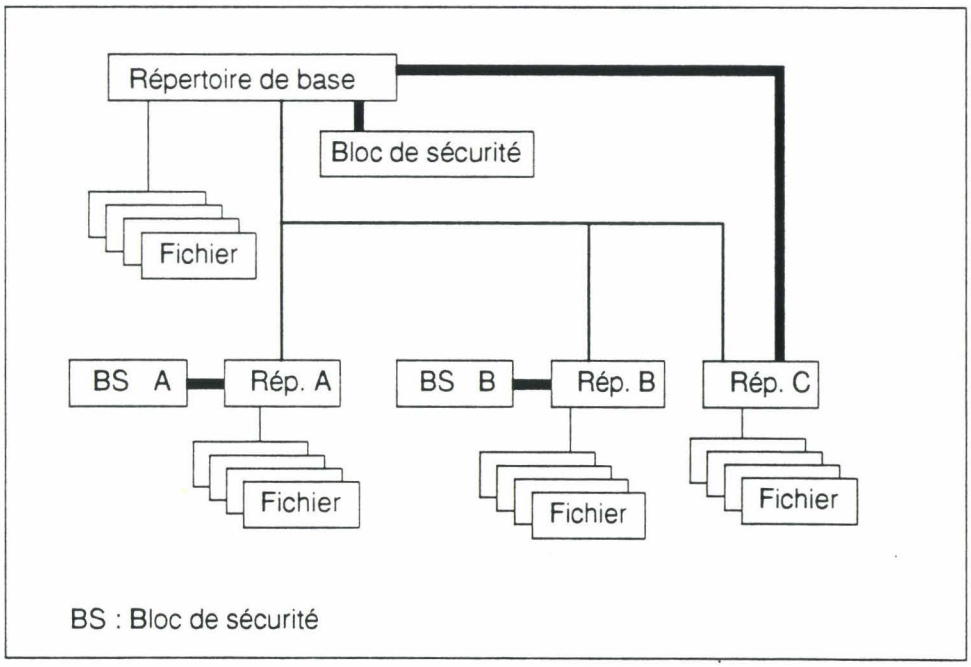


FIGURE 36 Organisation de l'espace utilisateur de la carte MCOS

- La carte CQL (Card Query Language), en cours de normalisation par l'ISO: Cette carte créée en 1992 à RD2P intègre un moteur de gestion de bases de données. Son atout majeur est de pouvoir être interrogée par l'intermédiaire d'un langage de requêtes de bases de données, le langage CQL (Card Query Language), lui même sous ensemble du standard SQL (Structured Query Language). La représentation des données au sein de cette carte s'articule autour d'une unique structure, la Table: il s'agit d'une liste de lignes, une ligne étant une liste de colonnes.

L'explication de l'absence ou de la pauvreté de la structuration des données dans les applications cartes réside principalement dans les très fortes contraintes d'intégration liées à la carte, et notamment:

- La taille de RAM disponible sur les cartes les plus performantes du marché ne dépasse pas **256 octets**. Cette taille, environ 32000 fois inférieure à celle couramment rencontrée sur des plateformes telles que PCs ou Macs (dotés de 8 Méga-octets de RAM), limite énormément le développement de strutures élaborées telles que celles qui servent de support aux applications multimédia classiques. L'encombrement de la mémoire RAM par rapport à l'ensemble d'un processeur encartable est tel que même les prochaines générations de cartes (par exemple celle annoncée dans *[CASC94]*) ne prévoient qu'une sensible évolution de capacité de RAM (pouvant atteindre 512 octets).

- La puissance des processeurs des cartes, aujourd'hui limitée à des architectures 8-bits (cadencées à 3,57 MHz), reste également un frein à la définition d'une structure de données complexe.

- Le temps de réponse est un facteur déterminant dans de nombreuses applications réclamant un traitement presque en temps réel (par exemple le contrôle d'accès ou la monétique).

Ces trois contraintes nous ont conduits à définir une structure de données minimum pour ne pas pénaliser l'occupation en mémoire RAM ni le temps de réponse. Bien que les structures considérées soient relativement élémentaires et classiques, elles apportent à la carte un progrès par rapport à ce qui existe. Après un bref rappel sur quelques notions de base liées aux langages orientés objets (comme le C++), nous définissons dans notre structure de données 3 types d'éléments: les vecteurs, les matrices et les listes.

Etant donné l'espace mémoire restreint de la carte, le terme «stockage de données multimédia» implique de disposer d'outils pour compresser les informations similaires à ceux employés sur les micro-ordinateurs classiques. Nous aborderons également cet aspect en adaptant des techniques déjà utilisées au monde spécifique de la carte.

IV.1.2 Les notions de Classe et d'Objet

Sans vouloir implémenter directement les applications cartes à puce dans des langages objets souvent très consommateurs en place mémoire et puissance de calcul comparativement aux capacités d'une carte à microprocesseur (Smalltalk, Eiffel...), nous avons repris certains concepts de base des langages orientés objets intéressants pour leur souplesse dans la mise en oeuvre et l'évolution d'applications.

D'une manière très simple, une Classe peut être définie par la réunion de 2 éléments:

- Des Données
- Des méthodes pour manipuler ces données

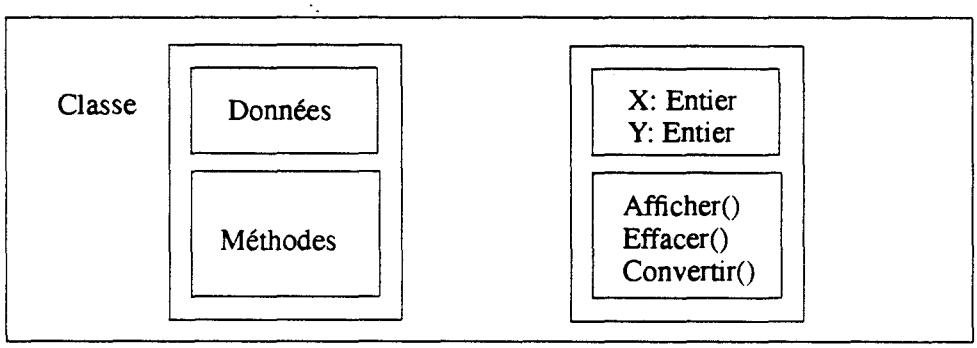


FIGURE 37 Définition d'une Classe - Exemple: la Classe Point (2D)

Le mécanisme d'héritage (voir *[STRO91]*) permet à l'utilisateur de créer de nouvelles classes sur la base de celles définies précédemment. Par exemple, dans le cas de la définition d'une classe «Point», on peut construire une nouvelle Classe «Segment» héritant de «Point», comme suit:

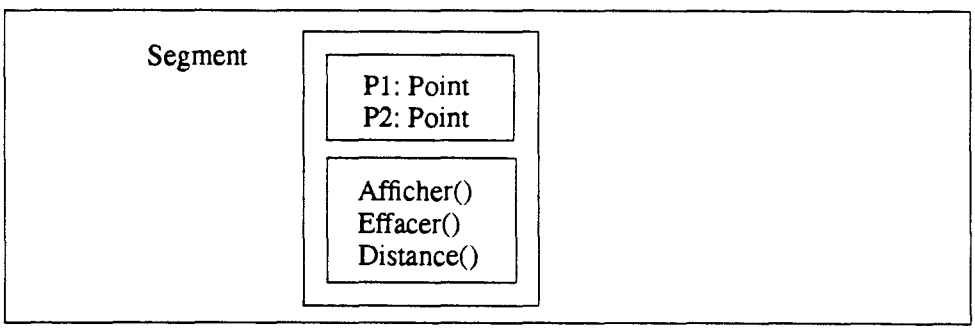


FIGURE 38 Définition d'une classe Segment

Dans ce cas, la méthode Afficher() de la classe Segment par exemple utilise celle de la classe Point

IV.1.3 Organisation des données

Dans notre approche la structuration des données multimédia se fait autour du concept de **Séquence,** qui est définie comme une suite finie et homogène de N-uples, de longueur quelconque:

$S = \{N_1, N_2, ..., N_i, N_{i+1}, ..., N_n\}$

Un N-uple peut se composer d'un ou plusieurs scalaires de type entier, réel ou booléen.

Exemples:

- Son = $\{E_1, E_2, ..., E_i, ..., E_m\}$

avec E_i un échantillon pouvant lui-même s'exprimer sous la forme d'un N-uple comme par exemple: $E_i = (C_{i1}, C_{i2}, ..., C_{i7}, C_{i8})$ pour un analyseur à 8 canaux.

- Empreinte Digitale = $\{P_1, P_2, ..., P_i, ..., P_n\}$

avec par exemple P_i un point remarquable de l'empreinte digitale en coordonnées polaires: $P_i = (M_i$:Réel, A_i:Réel).

Afin de pouvoir gérer dynamiquement les séquences créées en les concaténant , en les coupant ou en leur ajoutant des éléments au fur-et-à-mesure des traitements, nous proposons dans les points suivants 3 implémentations possibles de séquences, qui cherchent à donner le maximum de flexibilité tout en gardant une représentation compacte des informations.

IV.1.3.1 Le type Vecteur

C'est le type de base adopté pour le stockage d'une séquence de taille donnée. Cette représentation est compacte car il s'agit de stocker les objets de la séquence à des adresses mémoires consécutives.

Un vecteur est un objet composé de 2 éléments:

- Un nombre entier définissant la longueur de la séquence
- Un pointeur sur le premier élément de la séquence, l'accès aux autres éléments pouvant être calculée par une fonction simple

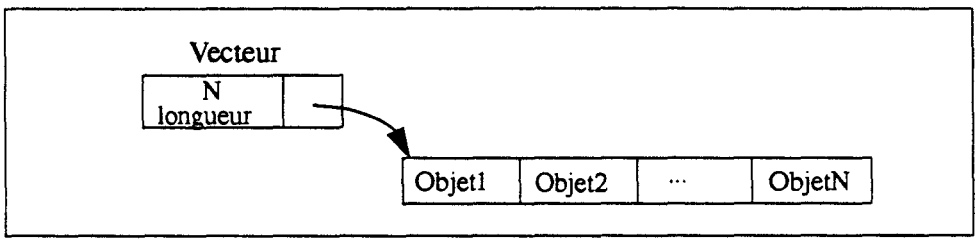


FIGURE 39 Le type Vecteur

En pratique nous avons défini la classe Vecteur en C++ de la manière suivante:

```
template<class T>
class Vector {
T* v; // pointeur sur le premier élément
int sz; // longueur du Vecteur
public:
Vector(int s=1) { v=new T[sz=s];}
Vector& operator=(const Vector&);
T& operator[](int i) {if(i<sz) return v[i];} };
```

L'opérateur [] permet d'accéder à n'importe quel élément de la séquence.

Le template <class T> permet de définir un Vecteur dont le type T des éléments qu'il contient n'est pas connu par avance. On peut donc créer par exemple un Vecteur de Points, voire même un Vecteur de Vecteurs de Points (un Vecteur dont les éléments sont des Vecteurs de Points).

IV.1.3.2 Le type Matrice

Enormément de traitements de données peuvent être réalisés par le calcul matriciel. C'est pourquoi nous avons implémenté le type Matrice. Il a été construit à partir du type Vecteur défini précédemment.

De la même manière que pour les Vecteurs, une Matrice est composée de 2 éléments:

- Un nombre entier définissant la longueur de la séquence de Vecteurs
- Un pointeur sur le premier élément du Vecteur de Vecteurs, l'accès aux autres éléments pouvant être calculé par une fonction simple.

La représentation du type Matrice est donnée sur le schéma suivant:

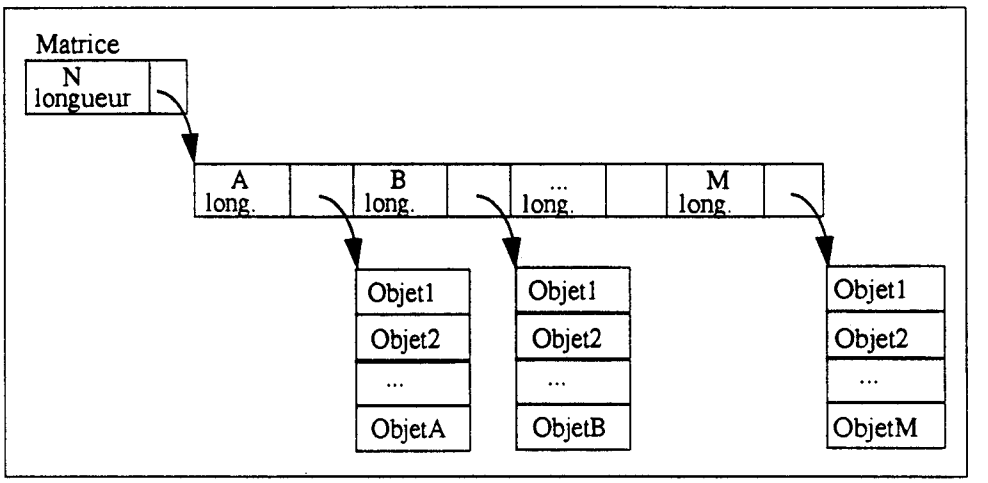


FIGURE 40 Le type Matrice

En C++, une Matrice est définie de la manière suivante:

```
template<class T>
class Matrix {
Vector<T>* v;
int sz; // longueur de la Matrice
public:
Matrix(int s=1) { v=new Vector<T>[sz=s];}
Matrix& operator=(const Matrix&);
Vector<T>& operator[](int i) {if(i<sz) return v[i];}
};
```

Par rapport à la définition classique d'un tableau 2D sous la forme d'une zone mémoire contiguë qui suppose d'avoir à disposition un bloc d'octets consécutifs pour le stockage de l'ensemble du tableau , le type Matrice défini ici permet de séparer les colonnes pour le stockage.

En outre, la plupart du temps, les manipulations de données font intervenir des matrices **creuses**, c'est-à-dire des matrices possédant un bon nombre de zéros. Dans le cas où des zéros consécutifs se situent en fin de colonne, le réglage indépendant de la longueur de chaque colonne permet de diminuer l'espace requis pour le stockage de la matrice.

IV.1.3.3 Le type Liste

De façon similaire au concept classique de gestion de Liste *[PAIR77]*, le type que nous avons implémenté permet de définir des séquences dynamiques d'objets chaînés au prix de quelques octets supplémentaires nécessaires au stockage des pointeurs gérant le chaînage.

Une Liste est définie par les 2 éléments ci-dessous:

- Un pointeur sur le premier élément de la liste
- Un pointeur sur le dernier élément, afin de pouvoir accéder à l'adresse du dernier élément sans avoir à parcourir le chaînage complet. Bien que non nécessaire, cette information de fin de liste est utile particulièrement lorsque l'on concatène deux listes, c'est pourquoi nous l'avons conservée.

Le type Liste est représenté sur la figure suivante:

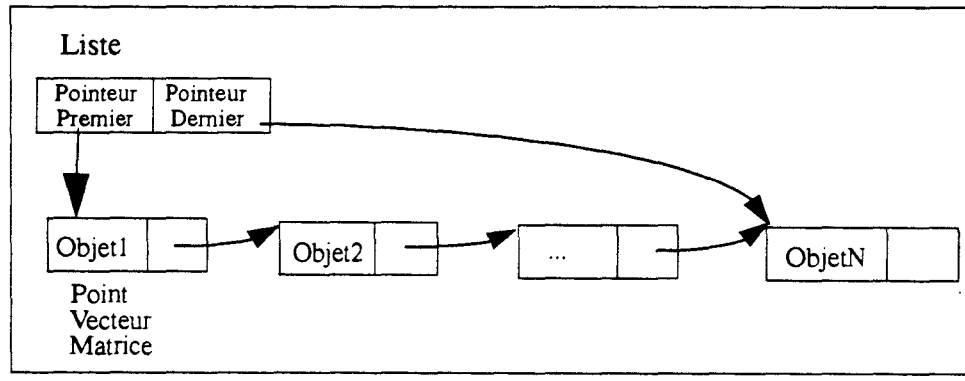


FIGURE 41 Le type Liste

La définition du type Liste en C++ est donnée ci-après:

```
template<class T>
class List {
List_Element<T>* First;
List_Element<T>* Last;
void Clear(void) { First = Last = NULL;}
public:
Liste& operator=(const Liste&);
Liste(void) { Clear(); }
};
```

où un élément de Liste «List_Element» est défini par:

```
template<class T>
class List_Element {
Vector<T>* L;
List_Element* Next;
public:
List_Element(List_Element* E) { Next = E; }
List_Element(Vector<T>* V) { L = V; Next = 0; }
List_Element(Vector<T>& V, List_Element<T> *E) {Next=E;L=&V;}
};
```

Conclusion

Combinés entre eux, les 3 types Vecteur, Matrice et Liste peuvent assurer une bonne souplesse dans la gestion des données. La structuration est essentielle dans la mesure où elle sert de support à toutes les manipulations futures.

IV.1.4 La réduction de la taille des données

IV.1.4.1 Introduction

Dans un environnement tel que la carte à microprocesseur, la place occupée par les données en mémoire est d'une importance capitale. Cela est d'autant plus vrai lorsque l'on souhaite stocker des données telles que des graphiques, des signaux sonores et des images qui nécessitent un grand volume d'informations. Par exemple, une image de résolution 128 x 128 pixels telle qu'une photo d'identité en 256 couleurs ou niveaux de gris (c'est-à-dire 8 bits/pixel) représente 16384 x 8 bits, soit 16K octets.

Etant donné qu'une carte à microprocesseur est dotée d'une mémoire EEPROM de faible capacité (environ 16K octets), on s'aperçoit que l'on est vite limité par la taille mémoire dès que l'on envisage le stockage et traitement de données autres que du texte.

De nombreux logiciels commercialisés s'intéressent à la réduction d'information. Ils peuvent se classer en deux catégories:

- La catégorie conservative de la totalité des informations: on parle alors de ***Compactage.*** La réduction des informations réside alors dans un changement de codage des informations adapté aux données (par exemple la technique du Run-Length-Coding qui consiste à coder les répétitions). Parmi les algorithmes de

compactage de données les plus utilisés, on peut citer les algorithmes de *Huffman, Shannon-Fano* et *Lempel-Ziv-Welch [PLUM 93]*.

- La catégorie non conservative d'informations. Cette technique ne se limite pas à l'élimination de la redondance dans les données mais supprime également les informations non pertinentes. Dans ce cas, une image par exemple obtenue après décompression a été plus ou moins dégradée mais de façon à ce que cette dégradation ne soit que peu ou pas du tout visible. On parle alors de ***Compression.*** Dans les algorithmes standards de ce type on trouve *JPEG (Joint Photographic Expert Group)* pour la compression d'images fixes, *MPEG (Motion Picture Expert Group)* pour la compression d'images animées, *PASC (Precision Adaptive Sub-band Coding)* pour la compression de sons *[GUOJ92]*.

Dans la littérature, on trouve également cette distinction sous la forme **Compression sans perte** (=Compactage) et **Compression avec perte** (=Compression).

Les techniques conservatives d'informations sont souvent plus faciles à mettre en oeuvre et nécessitent moins de calculs que les techniques avec perte d'informations mais le taux de compactage obtenu dépend beaucoup du contenu de l'image et en moyenne ce taux est nettement moins bon.

Dans le seul but de stocker dans une carte à puce des informations graphiques telles que des schémas, courbes, images, on peut imaginer recourir à des outils performants existant sur le marché pour effectuer une compression puis une décompression efficace et rapide de telles données. La carte devient alors un support de stockage d'informations compressées qui sont transférées ou extraites de sa mémoire. Tout traitement sur ces données s'effectue alors à l'extérieur de la carte et ne dispose plus de la sécurité garantie par l'environnement carte à puce.

Dans de nombreux cas de traitement de données (biométrie, reconnaissance de formes, analyse de signaux...), la sécurité impose que ce soit la carte elle-même qui effectue le traitement afin de ne jamais livrer ses références. C'est pourquoi il semble intéressant de pouvoir effectuer des opérations de compression et décompression de données à l'intérieur de la carte.

La technologie carte à microprocesseur impose certaines contraintes sévères en termes d'espace mémoire et de capacité de traitement (processeur 8 bits, mémoire de données de 10 Ko, mémoire RAM de 256 octets, temps de réponse limité). Dans le cadre d'une étude sur les moyens possibles de compression et décompression dans la carte, il s'agit donc de trouver un compromis entre le taux de compression obtenu, le temps de calcul effectif et l'acceptabilité du résultat rendu par une image ayant subi une certaine dégradation.

Dans la section qui suit nous évoquons brièvement certaines techniques de compression de données couramment utilisées pour le stockage et la représentation des informations. Elles correspondent aux systèmes que nous avons étudiés et ne

représentent en aucun cas une liste exhaustive, même si elles résument la majorité des techniques existantes. Une comparaison de plusieurs techniques de compression sans perte est également détaillée dans *[ARPS93]*.

Bien qu'étant des algorithmes effectuant des traitements et donc pouvant faire partie du chapitre IV, nous les voyons ici comme des outils de **stockage** de l'information. Nous étudions alors leur éventuelle intégration dans l'environnement carte à micro-processeur.

IV.1.4.2 Présentation des techniques de compression dédiées au stockage des informations

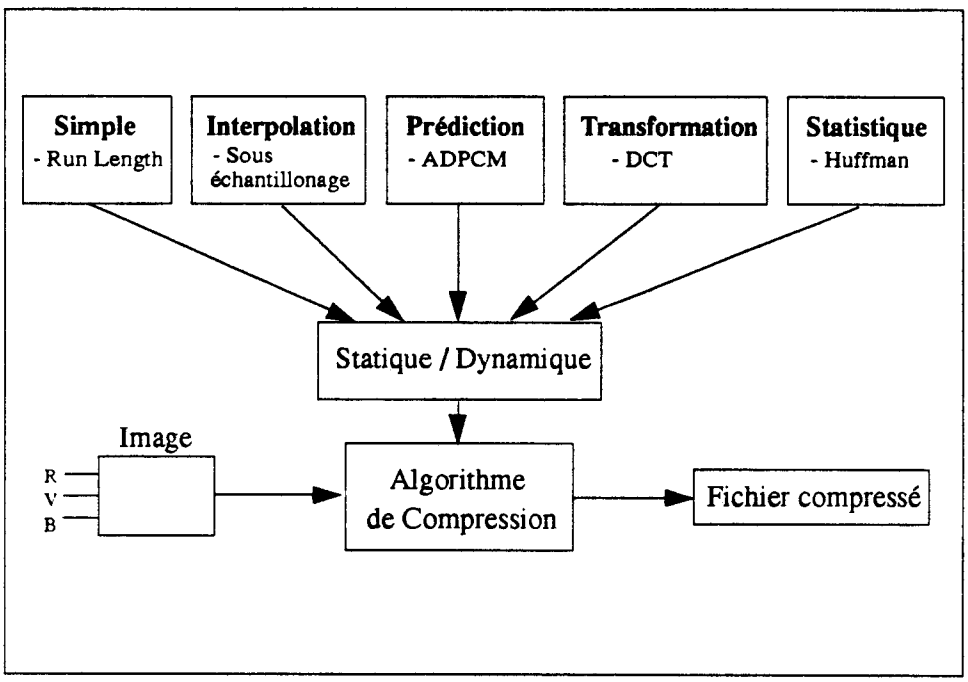


FIGURE 42 Les différentes méthodes de compression de données *[KOEG94]*

IV.1.4.2.1 La compression avec perte

1. La compression de type JPEG

Dans la seconde partie de ce chapitre concernant les manipulations associées aux données stockées dans la carte à puce, nous développons en détail cette technique comme exemple de traitement d'informations embarquées. C'est pourquoi ici nous renvoyons le lecteur à cette seconde partie.

2. La compression fractale

La compression fractale est une technique qui associe une équation (Fractale) à une image *[BARN88a]*. D'une part, la fractale peut être décrite avec quelques règles succinctes. D'autre part, elle contient la plupart ou la totalité de l'information présente dans l'image. Puisque les règles sont décrites sur moins de bits que l'image elle-même, il en résulte une compression des données.

Les très hauts taux de compression sont obtenus au prix de lourds calculs. Cependant, c'est uniquement la phase de compression qui est coûteuse, pas la décompression, car la difficulté calculatoire réside dans la recherche de la fractale associée à l'image que l'on veut compresser. Le «Collage Theorem» est la solution adoptée pour trouver les règles qui encodent une fractale à partir de l'image (voir *[BARN88b]*). Cette technique est stable dans le sens ou de petites erreurs dans la description des règles entraînent seulement de petites erreurs dans la représentation de l'image.

On peut envisager le stockage de fractales dans la carte à puce. Bien entendu la compression doit se faire dans un environnement extérieur. En revanche, on peut imaginer effectuer la décompression dans la carte, qui consiste à utiliser la fonction fractale pour retrouver les points de l'image. L'intérêt majeur réside dans le très fort taux de compression généralement obtenu, bien supérieur aux méthodes plus classiques telles que JPEG.

3. La compression par les réseaux de neurones

Tout comme la compression JPEG, cette méthode est reprise dans la seconde partie comme exemple de traitement possible dans la carte à puce. C'est pourquoi il ne sera pas évoqué ici. Un cas de compression par réseaux neuronaux est traité dans *[AROZ90]*.

IV.1.4.2.2 La compression sans perte

1. Le codage de Huffman

C'est certainement l'une des techniques les plus populaires de compression sans perte. Elle est d'ailleurs présente dans le standard JPEG comme l'une des étapes de la compression. Sous forme statique, elle consiste à attribuer, de façon définitive pour chaque signe différent du message à compresser, un code qui est fonction de la fréquence d'apparition de ce signe par rapport à l'ensemble du message. Il existe une forme dynamique plus efficace qui consiste à changer le codage au fur-et-à-mesure de l'évolution de la fréquence d'apparition de chacun des signes du message. Le détail des deux versions est présent dans *[NELS92]*.

Si la version statique de la compression de Huffman ne demande que peu de calculs et peut facilement être mise en place dans la carte, le procédé dynamique requiert davantage de puissance. Nous pensons qu'il ne se justifie pas dans les cartes actuelles vu le peu d'amélioration qu'il ajoute à la compression statique par rapport au coût qu'il apporte en terme de calcul.

2. Le codage arithmétique

Souvent meilleure que la méthode de Huffman au niveau du taux de compression obtenu, cette technique relativement calculatoire ne paraît pas envisageable sur les cartes à puce actuelles. Ce codage peu utilisé pour le moment est encore récent, c'est pourquoi il n'est pas à exclure dans les applications futures des cartes à micro-processeur. Pour davantage d'informations sur cette approche , se reporter à *[HOWA91]*.

Conclusion

Si le fait de réduire la taille des objets multimédia stockés dans la carte présente l'avantage certain d'économiser de la mémoire, il ne doit pas ralentir de façon trop importante l'application. C'est pourquoi la compression des données ne doit pas être systématique. Il s'agit alors de trouver un compromis entre le niveau de compression obtenu et la puissance de calcul requise pour effectuer cette compression.

Dans le cas de stockage et manipulation d'images ou de sons, la compression avec perte est néanmoins recommandée car, sans des facteurs de compression de l'ordre de 5 ou plus, la mémoire EEPROM de la carte est immédiatement saturée. Dans la partie suivante nous étudierons notamment des techniques permettant d'atteindre ces facteurs de compression.

IV.2 Le traitement des données dans la carte

Introduction

Dans la partie précédente, nous avons établi une structure de données à partir des observations notamment des chapitres II et III en prenant en compte les lourdes contraintes technologiques des cartes.

Cette partie s'intéresse aux traitements permettant de manipuler dans la carte les données stockées en gardant bien à l'esprit le contexte carte à micro-processeur qui oriente les choix d'implémentation en fonction des contraintes qu'il engendre en termes de capacité mémoire et puissance de calcul. On doit donc trouver un compromis entre des applications supportant une certaine complexité et les possibilités réduites de traitement des cartes.

IV.2.1 Approche par les Langages de Haut Niveau

IV.2.1.1 La Compression de Données adaptée à la Carte

IV.2.1.1.1 Rappels sur la compression JPEG

JPEG est une norme de compression applicable dans le domaine des images fixes. Son principe de codage s'effectue en trois étapes: une transformation mathématique basée sur la transformée de Fourier, une étape de quantification et une compression statistique dynamique adaptative *[WALL91]*.

La Tranformation de Fourier effectue une 'compaction d'énergie' d'un signal spatial dans le domaine des fréquences, elle facilite ainsi la réduction des informations non pertinentes. Etant donné que les coefficients de la transformation sont peu corrélés, la redondance est en grande partie éliminée. La compression des données consiste à éliminer les coefficients de faible énergie donc de faible signification dans l'image.

1. La Transformée Cosinus Discrète *(DCT: Discret Cosine Transform)*

Soit un bloc N x N de pixels f(j,k), où N est une puissance de 2 et j et k sont les indices de repérage des lignes et colonnes dans ce bloc. Par la tranformation de Fourier, on obtient un bloc de N x N nombres complexes notés F(u,v).

Soit (u,v) les coordonnées d'un point du bloc obtenu après transformation. L'équation de la transformation de Fourier peut s'écrire de la façon suivante (pour u et v compris entre 0 et N-1):

$$F(u,v) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j,k)\, e^{-\frac{2\pi}{N} i (uj + vk)}$$

La transformée de Fourier d'un bloc de NxN valeurs entières donne un bloc de NxN valeurs complexes c'est-à-dire $2N^2$ réels. Dans de telles conditions, il paraît difficile de réaliser de la compression en doublant la quantité d'information. Grâce aux propriétés de la transformation de Fourier décrites ci-après, on va pouvoir se ramener à un bloc résultat de N^2 valeurs:

- *la Transformée de Fourier d'une fonction paire ne donne que des coefficients réels.* Afin de pouvoir utiliser cette propriété dans le cas du bloc NxN, on peut obtenir une fonction paire en symétrisant ce bloc sur les quatre quadrants du plan (O,x,y), ce qui nous donne une image dont la transformée de Fourier ne possède que des termes réels (ici $4N^2$) comme le montre la figure suivante:

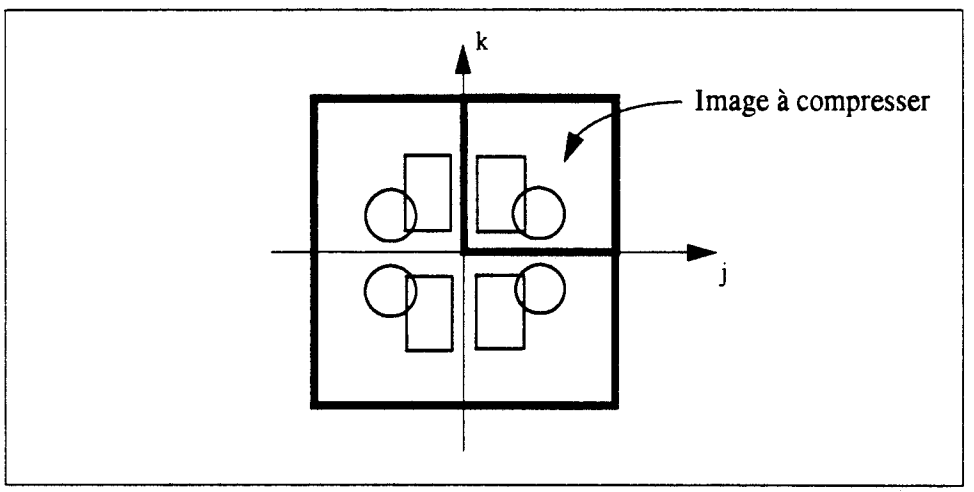


FIGURE 43 Bloc de NxN pixels symétrisé pour obtenir une fonction paire

- d'autre part, *la transformée de Fourier d'une fonction paire est elle-même paire*, donc il suffit de garder uniquement les coefficients du premier quadrant (soit N^2 coefficients).

La fonction f étant paire, on a :

$$f(\varepsilon x, \varepsilon' y) = f(x,y)$$
$$\varepsilon, \varepsilon' \in \{-1.1\}$$

Dans ce cas, la transformée de Fourier s'écrit aussi:

$$F(u,v) = \frac{1}{2N-1} \sum_{j=-N+1}^{N-1} \sum_{k=-N+1}^{N-1} f(j,k)\, e^{-\frac{2\pi}{2N-1}(uj+vk)}$$

ou encore:

$$F(u,v) = \frac{4}{2N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f_s(j,k) \cos\left[\frac{2\pi uj}{2N-1}\right] \cos\left[\frac{2\pi vk}{2N-1}\right]$$

avec f_s symétrisée de f:

$$f_s(j,k) = f(j,k)\ si\ (j \cdot k \neq 0)$$

$$f_s(j,k) = \frac{1}{2} f(j,k)\ si\ (j \wedge k) = 1$$

$$f_s(j,k) = \frac{1}{4} f(j,k)\ si(j,k) = (0,0)$$

Nous soulignons le fait que la DCT appliquée à l'image ne compresse pas l'image. Elle ne fait que changer sa représentation.

A titre d'exemple, considérons la transformée de Fourier du bloc 8x8 suivant, représentant un trait horizontal de largeur 2 pixels (cette image est binaire, les coefficients 1 représentent les pixels allumés):

0 0 0 0 0 0 0 0	4.0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0	0.41 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0	-3.57 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1	-1.0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1	2.445 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0	1.0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0	-1.0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0	-0.28 0 0 0 0 0 0 0
Image de départ	**Coefficients de Fourier**

FIGURE 44 Exemple de Transformation Cosinus d'une image

La Transformée Cosinus Discrète n'est pas appliquée en général directement sur les radiométries de l'image mais sur les trois composantes du système de codage (Y,U,V) qui sont non entrelacées et où l'information est principalement concentrée dans la luminance *[MARS92]*.

2. L'étape de quantification

C'est au cours de cette étape que sont enlevées les informations non pertinentes qui correspondent en général aux hautes fréquences de l'image. Lorsque l'on s'éloigne du premier coefficient F(0,0) correspondant au niveau moyen du bloc NxN (aussi appelé le Cosinus Discret ou le Fondamental), les coefficients transformés correspondent aux fréquences de plus en plus importantes. Ce sont ces valeurs que nous allons éliminer en partie afin de compresser l'image.

C'est pourquoi on va accorder à chacun des coefficients obtenus par la transformée discrète un poids différent. Ceci revient à diviser les coefficients de la matrice obtenue après transformation cosinus discrète par une matrice dite de quantification. JPEG propose pour la Luminance et la Chrominance les matrices de quantification suivantes *[TERR92]*:

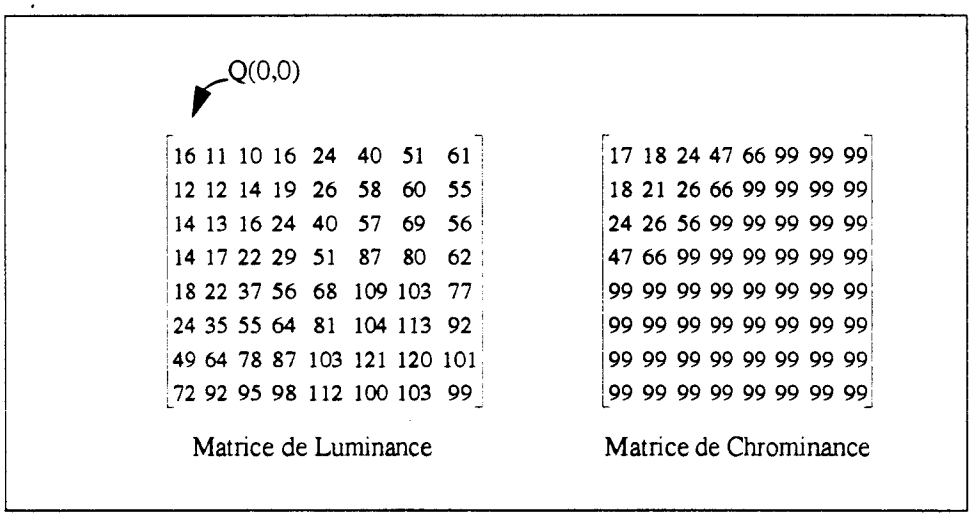


FIGURE 45 Matrices de quantification proposées par JPEG

Les coefficients les plus élévés se situent dans la partie inférieure de la matrice de quantification coupée par sa deuxième diagonale et permettent ainsi de filtrer davantage les hautes fréquences.

3. Le codage statistique

Le codage des coefficients intervenant après la quantification peut se décomposer en deux étapes. D'une part, étant donné qu'après avoir réalisé l'étape

de filtrage, la plupart des coefficients des hautes fréquences tendent vers zéro, il semble intéressant de pouvoir regrouper au maximum ces coefficients nuls, afin d'augmenter l'efficaté du codage. D'autre part, on opère sur les blocs un codage de Huffman afin de compacter séquentiellement le fichier dont la taille a déjà été réduite par l'étape de quantification suivie du regroupement des zéros.

Le codage des zéros

Avant de stocker séquentiellement les coefficients en vue d'un codage, il est judicieux de ranger les coefficients de la matrice de façon à augmenter les plages de zéros consécutifs. JPEG effectue ce rangement selon l'ordre de Cantor (voir la figure ci-dessous) de manière à regrouper les coefficients des hautes fréquences qui, après le filtrage, sont souvent nuls.

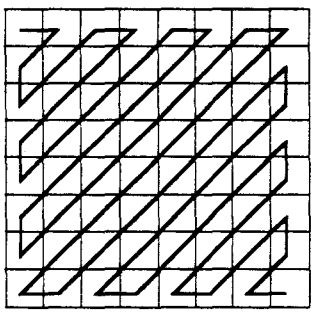


FIGURE 46 Rangement selon l'ordre de Cantor

On utilise alors la méthode du Run-Length-Coding qui consiste à coder le nombre de zéros successifs rencontrés dans un bloc.

Par exemple, 2500000000000 sera codé par 250#10.

Le codage de Huffman

La dernière étape de la compression JPEG consiste à compacter les coefficients (donc sans perte d'information) par un algorithme de Huffman adaptatif *[NELS92]* à partir de tables connues par avance. Ainsi, il est inutile de stocker un en-tête pour chaque bloc, ce qui réduirait le taux de compression.

Les taux de compression obtenus par cette méthode peuvent varier considérablement, tout dépend de la qualité que l'on désire dans la restitution de l'image. On peut néanmoins espérer des taux de compression de l'ordre de 10 à 16 sans pertes trop sensibles à l'oeil.

IV.2.1.1.2 Modifications apportées à JPEG pour son intégration dans la carte

Tout comme dans le cas de l'identification biométrique, dès lors que l'on manipule et utilise des images à des fins de sécurité (par exemple le contrôle d'une photo d'identité), l'algorithme de compression ou décompression ne garantit son résultat que s'il s'opère dans un environnement sécurisé.

C'est pourquoi il est intéressant de le réaliser dans la carte. L'intérêt supplémentaire de cette dernière solution réside dans le fait de ne pas avoir à intégrer dans le terminal extérieur de matériel ou logiciel spécifique à la compression ou décompression.

En considérant les fortes contraintes technologiques imposées par la carte, il semble évident que le type d'algorithme proposé par JPEG doit subir certaines simplifications s'il on veut qu'il s'exécute dans la carte en des délais acceptables. Nous rappelons ci-dessous les contraintes très lourdes liées à l'intégration d'un algorithme tel que JPEG dans la carte:

- la carte ne dispose que d'environ 256 octets de mémoire RAM (pour les cartes les plus puissantes actuellement) et 10 Koctets de ROM.

- les processeurs 8-bits des cartes actuelles rendent difficile l'exécution d'algorithmes calculatoires tels que JPEG qui doivent se faire presque en temps réel pour présenter un intérêt dans une application (par exemple, le contrôle d'identité par photo stockée dans la carte doit être quasiment immédiat). C'est pourquoi le temps de réponse associé à l'algorithme de compression ou décompression doit être de l'ordre de la seconde.

C'est dans cette optique que sont décrites ci-après les adaptations possibles pour la carte à puce. Nous reprenons chaque étape de la compression JPEG en expliquant les améliorations que l'on peut apporter compte tenu des contraintes de la carte. Il s'agit en fait de simplifier les calculs et certaines étapes de l'algorithme classique tout en minimisant les pertes visuelles qui en résultent.

1. Transformée Entière et Simplification Cosinus.

La Transformée Cosinus Discrète décrite précédemment fait intervenir des opérations sur des nombres flottants, utilise de nombreux appels à des fonctions cosinus très gourmandes en temps de calcul. En ne gardant que la partie entière des calculs de cosinus, la formule de Transformée Cosinus ne traite plus que des entiers.

La perte de précision de l'image résultante est suffisamment peu perceptible pour que l'on adopte cette modification aux besoins de la carte.

D'autre part, pour des blocs de 8x8 pixels (N=8), le terme

$$\cos\left[\frac{2\pi uj}{2N-1}\right] = \cos\left[\frac{2\pi}{15}\cdot uj\right]$$

ne peut prendre que 15 valeurs distinctes étant donné que la fonction cosinus est 2pi périodique.

Ces 15 valeurs sont les cosinus que l'on obtient pour les multiples de 2pi/15, comme le montre la figure suivante:

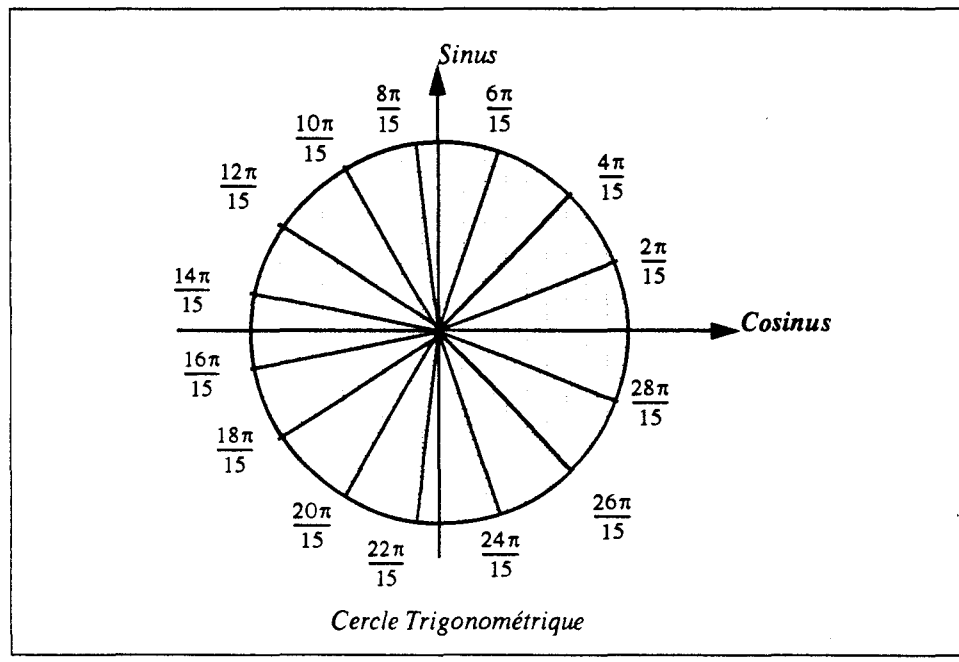


FIGURE 47 Cosinus intervenant dans la Transformée d'un bloc de 8x8 pixels

Il est alors aisé pour l'implémentattion d'un nouvel algorithme de remplacer tout le calcul des cosinus par une table de 15 valeurs entières. L'accès aux différentes valeurs de cosinus sera calculé de manière relativement simple en fonction des valeurs de u et j, ou de v et k.

On remarque que les arcs multiples de 2pi/15 sont symétriques par rapport à l'axe des abscisses; il n'y a donc en fait que 8 valeurs de cosinus différentes à stocker. Mais on gardera la table de cosinus de 15 valeurs pour ne pas alourdir le calcul de repérage au sein de cette table. La table des cosinus est alors la suivante:

arc	$\frac{0\pi}{15}$	$\frac{2\pi}{15}$	$\frac{4\pi}{15}$	$\frac{6\pi}{15}$	$\frac{8\pi}{15}$	$\frac{10\pi}{15}$	$\frac{12\pi}{15}$	$\frac{14\pi}{15}$	$\frac{16\pi}{15}$	$\frac{18\pi}{15}$	$\frac{20\pi}{15}$	$\frac{22\pi}{15}$	$\frac{24\pi}{15}$	$\frac{26\pi}{15}$	$\frac{28\pi}{15}$
cos x10	10	9	7	3	-1	-5	-8	-10	-10	-8	-5	-1	3	7	9

FIGURE 48 Table des cosinus entiers utilisés dans la Transformée DCT

Même si des processeurs 32 bits pour cartes à microprocesseur commencent à voir le jour *[PEYR94]*, la plupart des cartes actuelles travaillent sur des mots de 8 bits. La réduction des coefficients de la DCT à des entiers codés sur 8 bits signés (donc prenant des valeurs de -128 à +127) permettrait un gain de temps conséquent dans les opérations.

Pour cela, il suffit soit de filtrer davantage les coefficients de la DCT afin d'obtenir des valeurs plus faibles mais dans ce cas la dégradation de l'image est non négligeable, soit de réduire par exemple proportionnellement les valeurs entières de la table des cosinus. Dans ce dernier cas, on s'aperçoit que la dégradation de la précision que nous avons obtenue est nettement plus faible. Des tests ont été effectués avec les tables de cosinus ci-dessous:

arc	$\frac{0\pi}{15}$	$\frac{2\pi}{15}$	$\frac{4\pi}{15}$	$\frac{6\pi}{15}$	$\frac{8\pi}{15}$	$\frac{10\pi}{15}$	$\frac{12\pi}{15}$	$\frac{14\pi}{15}$	$\frac{16\pi}{15}$	$\frac{18\pi}{15}$	$\frac{20\pi}{15}$	$\frac{22\pi}{15}$	$\frac{24\pi}{15}$	$\frac{26\pi}{15}$	$\frac{28\pi}{15}$
cos x10 réduit à 60%	6	5	4	2	-1	-3	-5	-6	-6	-5	-3	-1	2	4	5
cos x10 à 30%	3	3	2	1	0	-1	-2	-3	-3	-2	-1	0	1	2	3
cos x10 à 10%	1	1	1	0	0	-1	-1	-1	-1	-1	-1	0	0	1	1

FIGURE 49 Réduction des cosinus pour avoir des coefficients DCT sur 8 bits.

Le tableau précédent indique qu'en réduisant les cosinus à 10% de leur valeur entière de départ, on arrive au cas limite ou seuls des 0 et des 1 sont utilisés. Dans ce cas, les coefficients DCT peuvent toujours être compris entre -128 et 127 sans que la qualité de l'image ne soit affectée de façon trop importante (le niveau des coefficients DCT maximal et minimal est dans tous les cas ajustable en opérant un facteur multiplicatif inférieur à 1 à la matrice de quantification).

2. La quantification dynamique

Dans le cadre de la compression d'images possédant certaines zones plus uniformes que d'autres, ne serait-il pas intéressant d'établir le degré de filtrage en fonction du contenu de l'image? Nous avons mené des recherches dans ce sens.

S'il on prend par exemple le cas d'une photo d'identité, le fond est souvent plus uniforme que les traits du visage. De plus, il n'est doté que d'un faible intérêt par rapport à la reconnaissance du visage. De ces considérations nous avons décidé d'attribuer un coefficient multiplicatif aux termes de quantification. Ce coefficient doit varier en fonction du nombre de hautes fréquences contenues dans un bloc. Ainsi, on filtrera davantage un bloc possédant peu de hautes fréquences.

Cette opération consiste à créer une variable COMPTEUR qui évalue le nombre de moyennes ou hautes fréquences d'un bloc dépassant un certain SEUIL. Le coefficient multiplicatif de quantification est alors une fonction de la variable COMPTEUR. Les résultats en terme de gain de compression sont étudiés dans la partie *EVALUATION* plus loin dans ce chapitre. On n'affectera pas de quantification dynamique au coefficient fondamental F(0,0) de la DCT représentant le niveau moyen du bloc car dans ce cas nous avons observé des différences nettes de niveaux de gris d'un bloc à l'autre qui altèrent la qualité de l'image de façon significative.

3. Ordre de rangement avant codage

L'ordre de Cantor utilisé par JPEG (*voir les rappels sur JPEG*) a l'avantage d'optimiser le regroupement de zéros consécutifs lorque l'on passe d'une représentation matricielle (tableau à 2 dimensions) à une représentation à 1 dimension (rangement séquentiel des données).

Néanmoins, il nécessite la mise en oeuvre d'un petit algorithme de passage de coordonnées (j,k) à un indice de repérage i (fichier séquentiel) qui demande un certain nombre d'opérations et qui alourdit la simplicité de l'algorithme de compression.

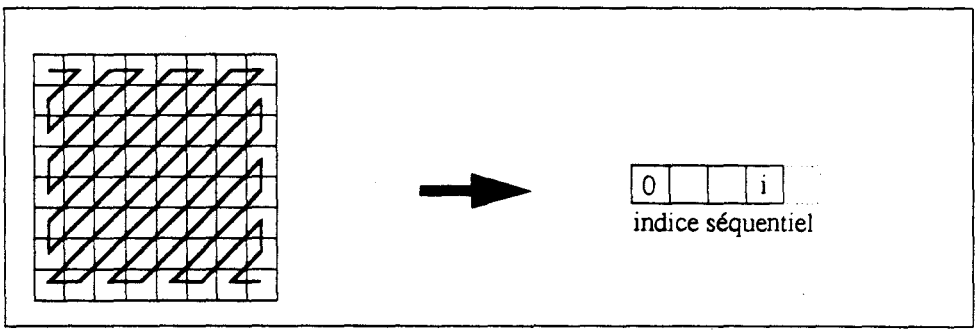


FIGURE 50 Passage de 2 dimensions à 1 dimension

Une manière de l'implémenter consiste à dire qu'une ligne diagonale de l'ordre de Cantor représente les éléments tels que (j+k) = Constante = N°de diagonale. L'indice i d'un élément de coordonnées (j,k) s'obtient donc en calculant la somme S des longueurs des diagonales précédentes plus la longueur de la diagonale en cours (qui est soit j soit k suivant le sens de parcours dans la diagonale où se trouve l'élément (j,k)). S est en fait la somme des (j+k-1) premiers termes d'une suite arithmétique de raison 1. On a donc:

$$S = \frac{(j+k-1)(j+k-2)}{2}$$

Ce calcul est valable uniquement pour la première moitié de la matrice (donc jusqu'à j+k=7). Pour l'autre moitié, c'est quasiment le même calcul mais en commençant le parcours par la fin de la matrice (j+k=14), comme le retrace l'algorithme suivant:

```
Passage_2D_1D (j,k)
{ int l,m,pair;
for(j=0;j<=7;j++)
for(k=0;k<=7;k++)      {
                       if (j+k>=8)   { l=7-j; m=7-k; }
                       else          { l=j;   m=k;   }
                       if ((j+k)%2) pair=1; else pair=-1;
                       i=((l+m)*(l+m+2)+pair*(m-l))/2;
                       if(j+k>=8) i=63-i;
                       return( i ); }}
```

Afin de simplifier l'algorithme de passage de 2D à 1D, nous avons utilisé le rangement selon l'ordre de Morton *[CHAS91]* , comme suit:

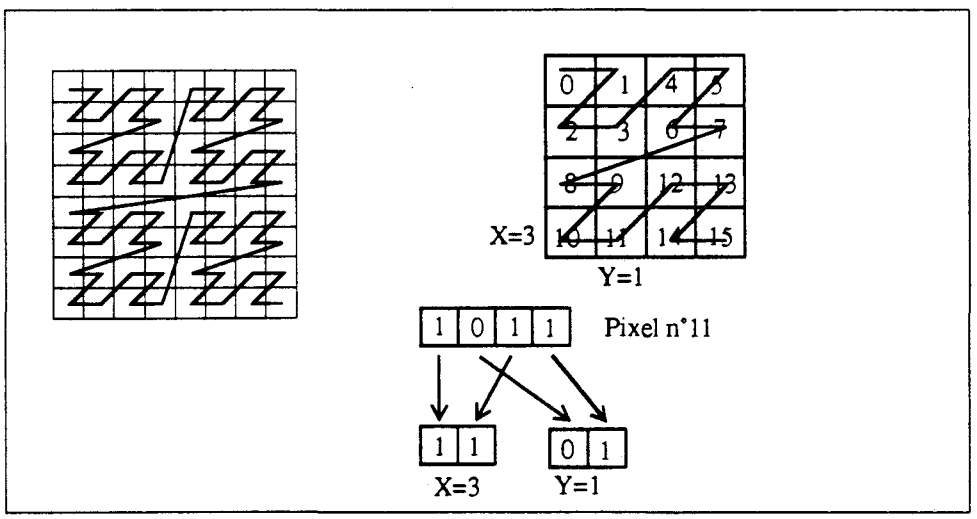


FIGURE 51 Passage 2D à 1D par l'ordre de Morton

En effet, pour passer de l'indice i aux coordonnées (X,Y) d'un élément de tableau 2D, il suffit de prendre un bit sur deux du codage de i en binaire pour obtenir les X, et les bits restants pour obtenir les Y.

L'opération inverse consiste à effectuer simplement quelques décalages comme le montre la ligne algorithmique suivante:

```
#define MORTON(j,k)
((j&100)<<3)|(((k&100)|(j&10))<<2)|(((k&10)|(j&1))<<1)|(k&1)
```

Dans ce cas on a légèrement perdu en optimisation du regroupement des zéros, mais on a réduit sensiblement la taille de l'algorithme (en EEPROM mais aussi en RAM) et surtout le temps de calcul (les décalages se font en un seul temps d'horloge alors que les multiplications en consomment plusieurs (10 environ)).

4. Le codage de Huffman

Le codage de Huffman dynamique nécessite la création d'un arbre binaire c'est à dire un espace mémoire de taille importante en comparaison de la capacité en RAM d'une carte à microprocesseur (256 octets).

C'est pourquoi dès le départ on abandonne l'idée de mettre en place un codage dynamique. La transformation DCT puis le filtrage font apparaître une majorité de 0, et les autres coefficients majoritaires se situent autour de 0. La répartition des fréquences d'apparition de coefficients de même valeur donne, en moyenne, une courbe de la forme suivante:

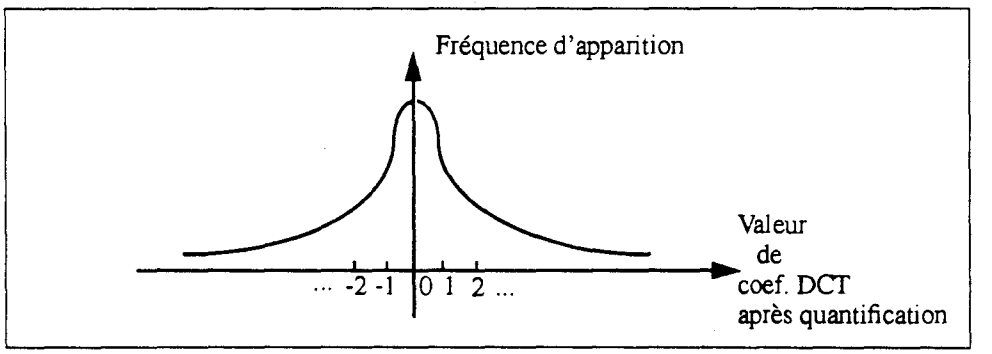


FIGURE 52 Répartition des coefficients DCT

On peut donc établir une table de référence des codes de Huffman et la stocker en ROM. La carte n'a plus qu'à effectuer un travail séquentiel de lecture/comparaison des bits à coder ou décoder. Si l'on ne veut pas avoir à stocker toute la table de référence, soit 256 codes de Huffman, on peut très bien avoir seulement une table pour les toutes premières valeurs (0,1,-1,2,-2,...,10 par exemple) et ensuite effectuer une numérotation logique car dans une image donnée les fréquen-

ces d'apparition de coefficients supérieurs à un seuil de l'ordre de 8 ou 10 sont faibles et quasiment identiques.

5. Antialiasing

Afin d'améliorer la qualité de restitution de l'image, nous avons recouru à certaines méthodes d'antialiasing. Celles-ci permettent d'atténuer les contrastes et pour le type de compression que nous effectuons elles s'avèrent particulièrement efficaces tout en étant simples à mettre en oeuvre. Il appartient à l'utilisateur de définir si ces techniques doivent se faire à l'intérieur ou l'extérieur de la carte.

Une méthode très simple consiste à considérer chaque pixel de l'image et lui attribuer un poids par rapport à ses voisins en 8-connectivité. On utilise par exemple la matrice suivante:

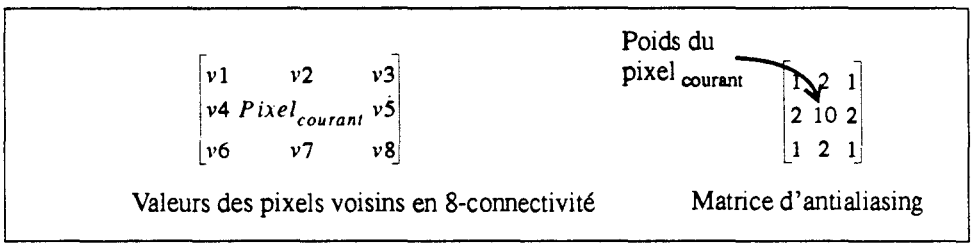


FIGURE 53 Exemple d'antialiasing

Les voisins les plus éloignés du pixel auront un poids plus faible dans le calcul de la moyenne. Celle-ci consiste à réaliser l'opération suivante:

$$P_{courant} = \frac{1}{\sum poids}\left[10P_{courant} + 2\,(v2 + v4 + v5 + v7) + v1 + v3 + v6 + v8\right]$$

Lors de l'évaluation de la compression dans la carte à mémoire, l'anti-aliasing est effectué à l'extérieur de la carte. Nous l'avons volontairement mis à l'extérieur car il rallonge le temps de calcul et l'on considère que cette étape ne se justifie pas à l'intérieur de la puce.

IV.2.1.1.3 Evaluation de la compression dans la carte

1. Généralités

La compression de graphiques et d'images destinée à la carte a pour objectif d'augmenter la capacité d'accueil en terme de place de stockage, ceci afin de pouvoir réaliser des opérations de type manipulations d'objets graphiques. Mais l'émetteur d'une application qui souhaiterait avoir dans la carte des images afin de

les utiliser pour des traitements divers peut très bien effectuer l'opération de compression à l'extérieur de la carte. Seule la décompression de l'image semble devoir se faire dans la carte (pour garantir la sécurité, aspect évoqué dans la partie qui précède).

Prenons l'exemple d'un permis de conduire: la photo d'identité pourrait être compressée à l'extérieur puis chargée dans la mémoire puisqu'à ce moment-là c'est l'émetteur de permis de conduire qui garantit la sécurité et non la carte (elle n'est pas encore totalement personnalisée). C'est seulement lors d'un contrôle d'authentification du porteur par rapport à sa photo d'identité que la carte elle-même doit décompresser la photo (pour que la comparaison s'effectue à l'intérieur), garantissant ainsi la sécurité de l'opération.

C'est pourquoi dans l'implémentation de la compression (ou plutôt décompression) dans la carte, il n'y a pas besoin d'étape de quantification. Seules sont nécessaires les étapes suivantes:

- Décompression de Huffman statique par blocs
- Lecture du bloc obtenu avec décompression de la partie Run-Length-Coding de façon à obtenir un bloc de 64 coefficients (ceux de la DCT)
- Remplissage d'une matrice 8x8 selon l'ordre de Morton par la conversion de données séquentielles en données indicées par des coordonnées (j,k).
- Transformation Cosinus Discrète Inverse
- Envoi des données de la carte vers l'extérieur sous la forme d'un fichier image (de type bitmap par exemple)

Dans un premier temps, nous allons évaluer l'apport de la quantification dynamique par rapport à la quantification statique dans le système de compression. Puis nous simulerons l'algorithme de décompression effectué par la carte grâce à un outil d'évaluation d'architectures (le projet OCEAN *[CARO93]*).

2. La quantification dynamique

Les résultats ci-dessous ne permettent pas de déterminer avec exactitude le gain de la quantification dynamique par rapport à la quantification statique car les tests n'ont été effectués que sur un nombre réduit de photos d'identité. De plus, l'évaluation de la qualité de l'image restituée est difficile car elle fait intervenir un jugement personnel et peut donc varier d'une personne à l'autre. Néanmoins, elle donne une idée de ce que l'on peut attendre de la quantification dynamique.

L'exemple évoqué ci-dessous retrace la quantification effectuée sur les coefficients DCT d'une photo d'identité de format 128x128x256 niveaux de gris (cette image non compressée occupe donc 16K octets).

La variable SEUIL_QUANTIF du tableau suivant est un paramètre utilisé dans l'algorithme. Dès lors qu'une 'haute fréquence' dépasse cette valeur seuil, un COMPTEUR est incrémenté. Ce dernier intervient comme coefficient multiplicatif dans le calcul du facteur Quantification_dynamique rajouté à la matrice de quantification fournie par JPEG.

Le fichier obtenu par transformée DCT puis quantification statique a pour taille 4307 octets. Après codage de Huffman, il ne fait plus que 1902 octets. Ces deux valeurs vont servir de référence pour le calcul des gains de compression. A titre d'information, l'image de départ (16384 octets) a été réduite à 1902 octets, c'est à dire d'un facteur 8,6.

Les résultats de compression utilisant la quantification dynamique sont résumés dans le tableau suivant:

SEUIL QUANTIF	Taille fichier DCT+$Q_{dyn.}$ (en octets)	Gain par rapport à DCT+Q_{stat} (en %)	Taille fichier après Huffman (en octets)	Gain par rapport à DCT+Q_{stat}+ Huffman (en %)	Facteur de compression total par rapport à $image_{départ}$
0	4214	2.2	1857	2.4	8.8
30	4069	5.5	1811	4.8	9.04
60	3807	11.6	1722	9.5	9.5
100	3499	18.8	1589	16.5	10.3
130	3335	22.6	1532	19.5	10.7
150	3176	26.3	1458	23.3	11.23
170	3045	29.3	1406	26.1	11.65
200	2815	34.6	1287	32.3	12.73
220	2683	37.7	1219	35.9	13.44
240	2565	40.4	1169	38.5	14
260	2411	44.0	1115	41.4	14.7
280	2288	46.9	1059	44.3	15.47
300	2161	49.8	995	47.7	16.47
320	2089	51.5	961	49.5	17.05
400	1755	59.3	806	57.6	20.3
600	1089	74.7	492	74.1	33.3

FIGURE 54 Tableau des résultats de la quantification dynamique

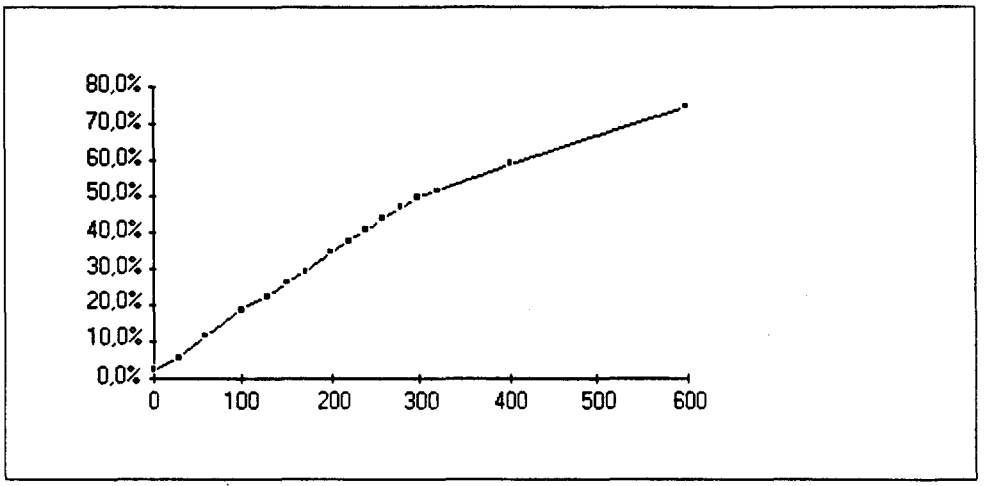


FIGURE 55 Courbe du gain de compression obtenu par DCT+Quantif. dynamique par rapport à DCT+Quantif. statique.

Dans les résultats, la dégradation n'est pas trop importante jusqu'à QUANTIF_SEUIL = 170 environ. Au delà, il apparaît des défauts nettement visibles qui souvent ne sont pas acceptables.

On peut néanmoins affirmer que la quantification dynamique peut apporter 10 à 20% de compression supplémentaire sans affecter outre mesure le rendu de l'image.

3. Evaluation de l'algorithme de décompression dans la carte

Cette évaluation a été réalisée par le simulateur *OCEAN (Outil de Conception et d'Evaluation d'Architectures Nouvelles)*. Cet outil, créé à RD2P, a pour objectif la conception et l'évaluation d'architectures RISC (Reduced Instuction Set Computer) adaptées aux futures cartes à micro-processeur *[CARO93]*. Il nous a permis d'évaluer notre algorithme de compression d'image dans l'environnement carte en simulant une architecture encartable d'un processeur RISC. L'architecture du processeur carte servant à l'évaluation est détaillée dans *[CARO94]*. L'algorithme a lui été écrit en langage C-CARD qui est un langage C proche du C ANSI mais spécifique au monde de la carte (voir *[GRIM91]*).

L'image considérée est toujours celle utilisée précédemment (128x128x256).

Une première simulation sur ce processeur encartable cadencé à 5 Mhz a donné un temps de 9.89 sec pour exécuter la phase de décompression de l'image. En étudiant les différents résultats et statistiques fournis par OCEAN, certaines optimisations peuvent être appliquées à l'algorithme de base:

- on peut remplacer les nombreuses multiplications par 0, 1 et -1 apparues lors des lectures de cosinus par 3 tests 'if'.

- après cette amélioration, une nouvelle simulation (donnant un temps total de 8.46 sec) a montré que les seules multiplications restantes sont des multiplications par 8. On peut donc les remplacer par des décalages de 3 à gauche.
  1 multiplication = 10 cycles d'horloge environ.
  1 décalage = 1 cycle d'horloge.

- L'architecture peut être améliorée (elle est alors un peu plus complexe) en séparant le bus d'instructions et le bus de données d'une part, et en rajoutant un multiplexeur de lecture des registres d'autre part *[CARO93]*.

Une dernière simulation nous a donnés le tableau de résultats suivant:

	à 5 Mhz (en sec)	à 20 Mhz (en sec)	à 50 Mhz (en sec)
Nb cycles d'horloge: 10080989	8.06	2.0	0.81
Nb cycles d'horloge (avec bus d'instruction et bus de données séparés): 9948176	7.95	1.98	0.79
Nb cycles d'horloge (comme précédemment et avec multiplexeur de lecture de registre): 4924911	3.94	0.98	0.39

FIGURE 56 Simulation de la décompression sur *OCEAN*

En ce qui concerne la place occupée en RAM, la simulation effectuée sur une compression ne faisant intervenir que des entiers codés sur 4 octets a établi une occupation de 154 octets (pour 256 possibles au total). En réduisant la taille de codage des entiers et en mettant la table statique des 15 cosinus dans la ROM au lieu de la RAM on arrive rapidement à descendre en dessous de 100 octets.

IV.2.1.1.4 Résultats visuels

Les photos qui suivent donnent une idée de la qualité de restitution après décompression d'une image par la carte à puce. La qualité réelle de l'image est meilleure que celle affichée ci-dessous car à l'écran elle est affichée en 256 niveaux de gris alors qu'ici elle n'est imprimée qu'en noir et blanc.

Les deux premières figures montrent deux exemples visuels d'images qui ont été compressées puis décompressées par la carte (du moins par la simulation de carte de la machine OCEAN). Dans ces exemples nous avons adopté toutes les modifications expliquées précédemment. A chaque fois nous montrons l'image originale et celle qui résulte de la compression puis décompression.

FIGURE 57 Image originale et image après décompression par la carte

En particulier, la photo originale ci-dessous est celle que nous avons prise comme exemple dans la partie d'évaluation de la quantification dynamique. La photo qui se situe à sa droite est celle obtenue par décompression en ayant utilisé la quantification dynamique correspondant à la variable SEUIL_QUANTIF=100. Si l'on reprend la ligne correspondante du tableau des résultats de quantification dynamique, on a:

SEUIL QUANTIF	Taille fichier DCT+$Q_{dyn.}$ (en octets)	Gain par rapport à DCT+Q_{stat} (en %)	Taille fichier après Huffman (en octets)	Gain par rapport à DCT+Q_{stat}+ Huffman (en %)	Facteur de compression total par rapport à $image_{départ}$
100	3499	18.8	1589	16.5	10.3

Le fichier avant décompression, que l'on stocke dans la carte, a pour taille 1589 octets, l'image originale occupant 16384 octets.

FIGURE 58 Autre photo décompressée par la carte

La photo de gauche de la figure qui suit correspond à l'image décompressée montrée ci-dessus que l'on a légèrement améliorée par la méthode d'anti-aliasing évoquée dans ce document. Celle de droite correspond au résultat obtenu si l'on retire uniquement l'étape de quantification dynamique, gardant la quantification statique de JPEG. Le fichier compressé occupe alors dans la carte 1902 octets, contre 1589 pour la quantification dynamique, sans différence sensible dans le résultat visuel du visage.

FIGURE 59 Image décompressée puis antialiasing (gauche) et Image compressée puis décompressée sans quantification dynamique (droite)

IV.2.1.1.5 Conclusion

Les résultats obtenus par la compression à base de DCT semblent encourageants étant donné qu'il est déjà envisagé de mettre des microprocesseurs RISC (du type de celui simulé sur OCEAN) dans la carte à puce.

De plus, ce principe de compression possède de bonnes propriétés vis-à-vis des fortes contraintes technologiques de la carte:

- Adaptation relativement simple pour la compression de signaux sonores puisqu'il s'agit en fait du cas d'une image mais traité à une seule dimension.

- L'ajout de couleurs dans les images au lieu de niveaux de gris n'augmente presque pas la taille du fichier compressé. C'est seulement l'algorithme qui est un petit peu modifié car il doit s'effectuer à la fois sur la luminance et la chrominance. Dans le pire des cas on peut estimer qu'il faudrait 2 fois plus de temps pour la compression.

- La compression et la décompression se font par le calcul de la même formule de DCT (cf la formule de Transformée Inverse dans les rappels de JPEG).

- On peut régler le taux de compression en fonction de la place dont on dispose. Cette caractéristique est très importante dans le cadre d'un environnement carte à puce où les octets sont comptés.

IV.2.1.2 La Reconnaisance Comportementale de la Signature Clavier

Nous avons choisi ce système d'identification biométrique pour expérimenter une implémentation utilisant les langages de haut niveau dans la carte étant donné son originalité. Ce système, que nous allons introduire et détailler dans cette partie, répond à la plupart des exigences évoquées dans le chapitre II concernant l'intégration possible d'un système d'identification biométrique dans la carte à puce. Les principaux avantages de ce système résident dans les points suivants:

- La Référence est de petite taille
- L'algorithme ne demande que peu de calculs
- Les étapes de constitution de la Référence et d'utilisation du système sont très faciles et rapides
- Le système possède un bon pouvoir de discrimination
- Il n'y a pas besoin de capteur biométrique particulier comme par exemple un micro ou un scanner, l'acquisition des données se faisant sur un clavier en général déja présent. Cette dernière caractéristique en fait un système très compétitif au niveau coût de revient.

IV.2.1.2.1 Description de la reconnaissance comportementale du clavier

L'idée de cette identification comportementale est partie de l'hypothèse suivante: «Lorsqu'il frappe sur un clavier une suite 'aléatoire' de chiffres un individu obéit à des lois de corrélation qui sont caractéristiques et identifiables».

Comme tout autre système d'identification biométrique classique, nous avons établi deux phases pour la mise en oeuvre de cette technique:

1. La constitution de la Référence

On demande à un individu de taper une suite de p chiffres (dans notre expérimentation p=1000) d'une manière qu'il croit aléatoire, en ayant posé ses deux mains sur le clavier de façon à attribuer une touche et une seule à chaque doigt. Pour des raisons de facilité et d'aisance sur le clavier, nous n'avons retenu dans un premier temps que 8 touches au lieu de 10, c'est-à-dire sans les pouces.

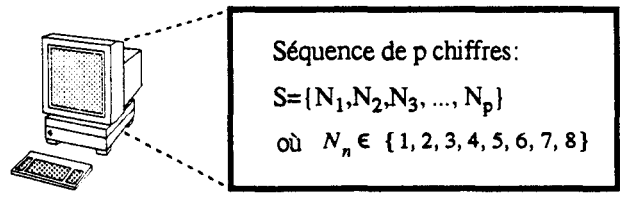


FIGURE 60 Données acquises pour la construction d'une Référence

A la différence des systèmes de dynamique du clavier évoqués dans le chapitre II et tenant compte du temps de frappe entre 2 touches, ce système ne demande à l'utilisateur aucune compétence particulière de frappe sur un clavier, l'acquisition des données se faisant très rapidement et sans réfléchir.

L'enregistrement d'une séquence de 1000 touches ne réclame qu'une minute ou deux tout au plus. Cette séquence peut être vue comme l'acquisition de 10 séquences de 100 chiffres, tout comme dans la majorité des autres systèmes biométrique, ceci pour établir une référence moyenne plus stable que si l'on avait seulement enregistré 100 touches.

Dans notre version utilisant le langage C la phase de vérification consiste à calculer 2 distances. La première est basée sur la fréquence d'apparition de chacune des touches, comme le montre l'exemple suivant:

Touche	1	2	3	4	5	6	7	8
Fréquence	7	12	14	8	24	17	8	10

FIGURE 61 Fréquence d'apparition de chacune des 8 touches

Si Fr_i représente la fréquence d'apparition de la touche i pour la Référence R et Ft_i est la fréquence d'apparition de la touche i pour le modèle de Test T, une distance euclidienne entre les deux représentations peut être donnée par:

$$D_1(r,t) = \sqrt{\sum_{i=1}^{8} (Fr_i - Ft_i)^2}$$

FIGURE 62 Distance entre la Référence et le Test - Cas à 1 dimension.

La seconde distance correspond aux fréquences d'apparition de deux touches successives, autrement dit de paires (N_i,N_{i+1}), comme sur l'exemple suivant:

Ni\Ni+1	1	2	3	4	5	6	7	8
1	**0**	**48**	**9**	**10**	**31**	**21**	**12**	**14**
2	**15**	**0**	**36**	**9**	**16**	**20**	**7**	**7**
3	**8**	**19**	**0**	**17**	**21**	**22**	**5**	**8**
4	**40**	**13**	**8**	**1**	**65**	**37**	**12**	**13**
5	**39**	**14**	**35**	**81**	**2**	**11**	**3**	**8**
6	**23**	**11**	**8**	**40**	**22**	**0**	**17**	**8**
7	**7**	**3**	**0**	**11**	**25**	**12**	**0**	**9**
8	**12**	**2**	**5**	**20**	**11**	**6**	**11**	**0**

FIGURE 63 Fréquence d'apparition de paires (N_i,N_{i+1}) pour une Référence

De la même façon, la distance euclidienne associée peut être:

$$D_2(r,t) = \sqrt{\sum_{i=1}^{8}\sum_{j=1}^{8} (Fr_{(i,j)} - Ft_{(i,j)})^2}$$

FIGURE 64 Distance entre la Référence et le Test - Cas à 2 dimensions

IV.2.1.2.2 Evaluation du système

Nous avons essayé de valider la fiabilité du système au travers de tests sur un ensemble restreint d'individus (13).

Si l'on se réfère aux notions introduites dans la présentation des systèmes biométriques du chapitre II, notre identification porte sur l'évaluation d'une distance entre le Test et la Référence dans l'espace d'identification. Ici il s'agit du calcul des 2 distances évoquées précédemment.

Si l'on veut que pour un individu A le Taux de Faux Acceptés soit minimum tout en ayant un Taux de Vrais Rejetés proche de zéro, il faut que les Tests fournis par des personnes autres que A soient en dehors de la boule de Référence associée à A tout en ayant les Tests de A dans cette boule de Référence:

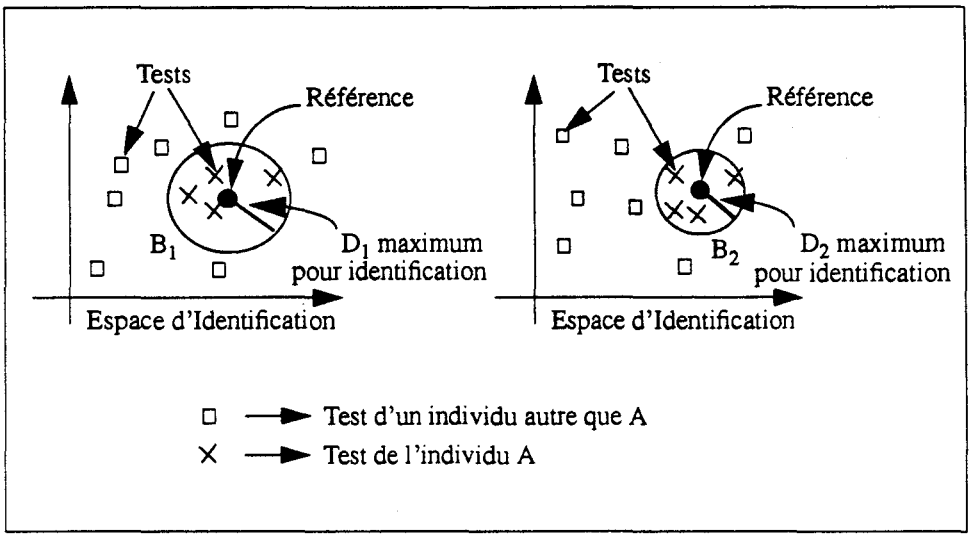


FIGURE 65 Signature Clavier vérifiée sur la mesure de 2 distances

Pour chaque individu nous avons enregistré 15 séquences de 100 chiffres. Les 10 premières ont servi pour la constitution du profil de Référence de cet individu, les 5 autres permettant à l'individu de tester son identification par rapport à cette Référence nouvellement acquise. Enfin, les 15 séquences ont également servi de Tests d'identification par rapport aux Références des autres candidats (correspondant aux tentatives d'intrusion frauduleuse).

Nous avons validé une identification uniquement lorsqu'un Test se situe dans la Boule de Référence B_1 pour la distance D_1 **ET** dans la Boule de Référence B_2 pour la distance D_2.

Un seul individu parmi les 13 a réussi à se faire passer pour un autre en utilisant l'une de ses séquences. Bien que ceci corresponde à une tentative de fraude réussie, cet individu ne sait pas comment il a réussi à s'introduire et donc est incapable de reproduire systématiquement sa fraude. Ce dernier avantage apporte à la reconnaissance comportementale du clavier un atout supplémentaire par rapport à de nombreux systèmes biométriques (dans le cas de la reconnaissance du visage, un masque permettant de frauder lors de l'identification sera valable de façon illimitée, de même qu'une voix enregistrée).

IV.2.1.3 Le Radar basé sur les Systèmes Experts

IV.2.1.3.1 Organisation de l'expertise

En reprenant une définition évoquée dans *[DELA87]*, nous pouvons établir la structure fondamentale d'un Système Expert comme la combinaison de trois éléments:

- une **Base de Connaissances**, le plus souvent divisée en assertions (ou Faits) et en Règles qui, par exemple, sont de la forme *SI condition ALORS conclusion.*

- un ensemble d'algorithmes de manipulation de la base de connaissances aussi appelé **Moteur d'Inférences** et qui rend le système apte à «raisonner» à partir des faits et des règles, pour inférer de nouveaux faits.

- un certain nombre d'**interfaces** permettant à l'expert de constituer, de modifier, de compléter la base de connaissances et aux utilisateurs de l'interroger.

Les règles et faits de la base de connaissance que nous avons construite sont ceux décrits au chapitre III introduisant le concept du Radar. Nous avons récolté l'avis de plusieurs experts issus d'opérateurs financiers de portée mondiale pour la constitution de la base de connaissance. Par souci de simplicité vis-à-vis de la faible capacité de traitement de la carte, nous n'avons utilisé dans l'expertise que les paramètres qui caractérisent la transaction elle-même. Dans le cadre d'un développement plus complet, on pourrait comme l'ont souligné les experts introduire des paramètres personnels se rapportant à l'individu, tels que son age, sa catégorie socio-professionnelle ou son revenu mensuel, engendrant davantage de règles.

Nous avons alors construit un moteur d'inférence selon un mode de chaînage avant en utilisant le langage PROLOG *[STER86]*.

IV.2.1.3.2 Mise en place du Radar en Prolog

Afin de construire facilement des règles de type «if (predicat) then ...» nous nous sommes intéressés au langage de programmation logique PROLOG . Sans vouloir détailler les nombreuses possibilités offertes par ce langage introduit au début des années 70, voici brièvement quelques unes de ses caractéristiques:

- Prolog est **logique**: un programme peut être vu comme un ensemble d'axiomes décrivant un problème.

- Prolog utilise une **véritable notion de variable**: ces variables permettent à l'utilisateur de créer et manipuler des objets non complètement spécifiés.

- Prolog permet d'écrire de façon concise des morceaux de programme à utilisations multiples.

- Prolog est **non-déterministe**, ce qui signifie que plusieurs solutions peuvent être trouvées pour une même «fonction», et Prolog effectue la recherche de ces valeurs d'une manière similaire au parcours d'une arborescence.

- Enfin, Prolog est **dynamique**. Il supporte ainsi les manipulations de bases de données, le concept de listes comme défini dans LISP, le parcours d'arbres etc...

IV.2.1.3.3 Evaluation du Radar utilisant Prolog

Nous avons effectué des tests sur le Radar pour évaluer sa capacité à reconnaître les comportements. Les échantillons de test du comportement de l'individu sont alors les nouvelles transactions qu'il effectue à la suite de celles prélevées pendant 6 mois qui ont servi à constituer la base de connaissances. Les échantillons de test correspondant à des fraudes sont dans un premier temps ceux créés par le générateur aléatoire de transactions frauduleuses en respectant tout de même certains intervalles comme expliqué à la fin du chapitre III. Les tests ont concerné 10 potentiels fraudeurs.

Le tableau suivant indique les résultats de cette simulation par rapport à un exemple de carte bancaire et donc de la modélisation d'un comportement d'utilisateur pendant une période de 6 mois. Les chiffres correspondent au nombre de transactions effectuées par chacun des 10 fraudeurs avant que la carte ne soit bloquée suite à une décision du Radar. Les seuils de la base de connaissance de l'utilisateur et les niveaux de sécurité ont été réglés de façon à ce que lui ne voit jamais sa carte bloquée par aucune de ses transactions durant 6 mois.

Fraudeur	Nombre de fraudes avant bloquage de la carte
Belin	**3**
Dubois	**4**
Dupont	**7**
Durand	**7**
Leconte	**5**
Legrand	**6**
Martin	**3**
Perin	**3**
Petit	**6**
Picard	**10**

FIGURE 66 Test de Radar sur des fraudeurs aléatoires

Nous avons alors utilisé comme seconde évaluation 10 nouveaux jeux de transactions frauduleuses collectées d'après le sondage résumé à la fin du chapitre III. Ces jeux de transactions frauduleuses correspondent à des scénarios d'attaques typiques retrouvés fréquemment dans le sondage. Cette manière de procéder pour constituer un jeu de transactions frauduleuses résulte de la non disponibilité de telles données récoltées par les organismes bancaires.

Le nombre de transactions réalisées par chacun des fraudeurs a abouti au tableau suivant:

Fraudeur	Nombre de fraudes	Cause de l'arrêt
Belin	**4**	**Carte bloquée**
Dubois	**1**	**Fraudeur**
Dupont	**3**	**Carte bloquée**
Durand	**7**	**Fraudeur**
Leconte	**12**	**Carte bloquée**
Legrand	**6**	**Carte bloquée**
Martin	**3**	**Carte bloquée**
Perin	**10**	**Carte bloquée**
Petit	**4**	**Carte bloquée**
Picard	**10**	**Carte bloquée**

FIGURE 67 Test de Radar sur des fraudeurs

De ces tests découlent certaines remarques sur l'attitude des différents fraudeurs et l'approche qu'ils ont utilisée pour mener à bien leur fraude:

- La stratégie de Mr Dubois consiste à n'effectuer qu'une transaction de montant élevé et à se débarasser de la carte. Le Radar est impuissant face à cette approche, mais dans ce cas la fraude totale est limitée au montant maximum autorisé pour une transaction.

- De la même façon, Mr Durand s'est arrêté après plusieurs transactions de différentes natures, d'un montant moyen.

- Les cas de fraudeurs dont le nombre de fraudes est élevé avant blocage de la carte correspondent en règle générale à ceux qui répartissent leurs transactions sur une longue durée. Sous cette dernière condition, le fraudeur augmente ses chances de se faire piéger car le propriétaire de la carte aura eu tout le temps de déclarer sa carte volée ou perdue.

- Les autres catégories se rapprochent davantage d'un comportement de fraudeur «typique» qui consiste à effectuer de nombreuses transactions sur une durée très courte.

L'approche Radar par système expert paraît efficace pour arrêter la majeure partie des attaques typiques ou du moins pour limiter les montants de la fraude.

IV.2.1.3.4 Le Radar en Prolog embarqué sur la carte à puce

Un Radar de type évoqué précédemment à bord d'une carte à puce possède les propriétés essentielles suivantes:

- il évite la connexion on-line avec un serveur centralisé de détection de fraudes (semblable à celui d'American Express *[DZIE89]*)

- il assure au propriétaire de la carte la confidentialité de «l'espionnage».

Un noyau d'interprêteur Prolog développé dans *[ROUS90]* nous a permis d'envisager l'intégration du Radar dans une carte à puce. Il semblerait que l'on puisse se satisfaire de 256 octets de RAM, moyennant des tailles de piles et de buffers relativement restreintes. Dans cette implémentation la taille de code évaluée pour le stockage de l'interprêteur représente environ 4 K octets.

Les limites du Radar sous forme de système expert

Bien que le Radar ait prouvé une relative efficacité sur un ensemble restreint d'expérimentations, il souffre de certaines faiblesses que nous allons évoquer ci-après:

- Du nombre réduit de règles pouvant être implémentées dans la carte à puce résulte une expertise limitée. Augmenter le nombre de paramètres associés à une transaction a tendance à augmenter rapidement le nombre de règles. Par exemple, pour les 4 entités <Nature, Montant, Date, Lieu>, nous avions 15 types de règles possibles (4 pour une seule entité, 6 pour 2 entités simultanées, 4 pour trois entités et 1 pour les quatre en même temps).
  Avoir 5 paramètres au lieu de 4 produirait jusqu'à 31 types de règles. Le nombre N de règles possibles appliquées sur un ensemble de n paramètres est en fait donné dans la relation suivante:

$$N = \sum_{p=1}^{n} \frac{n!}{p! \times (n-p)!}$$

- Si l'on considère une durée assez importante (de l'ordre de 6 mois), il se peut que le comportement d'un individu change alors que les règles du système expert envisagé n'ont pas évolué. Le fait de ne pas avoir d'évolution dynamique du système expert débouche sur un affaiblissement du Radar en fonction du temps. La résolution de ce problème demande une augmentation sensible de la complexité afin de gérer dynamiquement la base de connaissances relative à un individu. Pour la carte, il semble difficile d'ajouter cette étape sans augmenter de façon significative la taille de l'algorithme et la consommation de RAM.

- Une fois que le système expert a été mis en place, tout changement dans les paramètres décrivant la transaction implique des modifications importantes dans le système expert. Cet inconvénient a été rencontré par American Express dans son projet d'automatisation de vérification de transactions bancaires à l'aide d'un système expert *[PIKE87]*. Après avoir construit et mis en service son «Authorizer Assistant», il fut difficile de changer sa politique d'acquisition de carte pour un utilisateur. Un exemple correspondait au désir d'American Express de diminuer les informations demandées à ses clients lors de la création d'une carte.

Au vu des contraintes de place mémoire et de temps de calcul requis pour l'implémentation du Radar sous forme de système expert, nous nous sommes intéressés à d'autres approches, et notamment une basée sur les réseaux de neurones. Nous allons détailler ce type d'implémentation dans la partie qui suit.

IV.2.2 Approche par les Réseaux de Neurones

Intégrer la technologie neuronale aux applications cartes à puce suppose de prendre en compte 2 paramètres importants:

- La taille mémoire réduite disponible pour le stockage du réseau neuronal (en EEPROM).
- La puissance de calcul faible des processeurs encartés d'aujourd'hui (8 bits), bien que de nouvelles générations de cartes plus puissantes soient annoncées *[PEYR94]* (architectures RISC 32-bits).

IV.2.2.1 Introduction aux Réseaux de Neurones

La technologie des réseaux neuronaux est basée sur la modélisation de la structure neuronale du cerveau sur laquelle de nombreux scientifiques poursuivent d'intensives recherches afin d'en comprendre sa structure biologique et comportementale. Le cerveau humain est constitué de plus de 100 milliard de **neurones** interconnectés qui véhiculent des signaux électro-chimiques de diverses intensités.

Des nombreuses expérimentations biologiques ont résultés des modélisations informatiques de diverses complexités. Nous allons rappeler dans les parties qui suivent les principaux éléments nécessaires à la compréhension de la technologie neuronale. Pour une introduction plus détaillée sur les réseaux de neurones, se référer à *[LIPP87]* ou *[DAVA93]*.

IV.2.2.1.1 Le Neurone simple

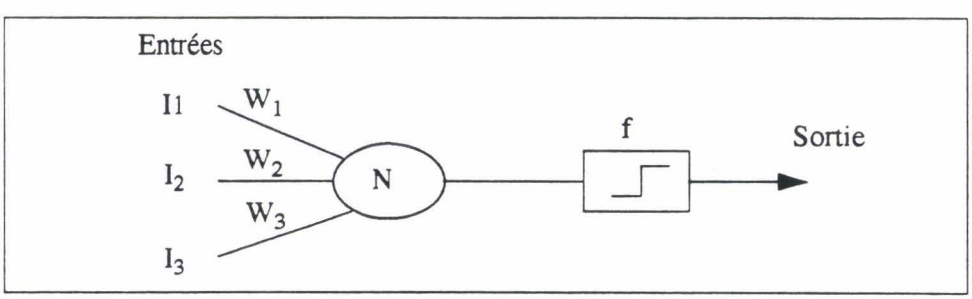


FIGURE 68 Un neurone simple à 3 entrées

Un réseau neuronale peut être vu comme une sorte de boîte noire possédant des entrées et sorties et réalisant une tâche particulière. Pour un ensemble (I_1, I_2, ..., I_N) d'entrées appliquées à un **neurone simple** (ou unité de calcul élémentaire),

la sortie est calculée par sommation de ces entrées pondérées par des **coefficients** (W_1,W_2, ..., W_N) (aussi appelés «**poids**») qui représentent la contribution de chaque entrée. La valeur résultante est alors comparée à une valeur **seuil**. Si ce seuil est supérieur à la somme pondérée, une valeur 0 par exemple est attribuée à la sortie du neurone, sinon une valeur de 1 représente la sortie du neurone. Le procédé peut se résumer de la manière suivante:

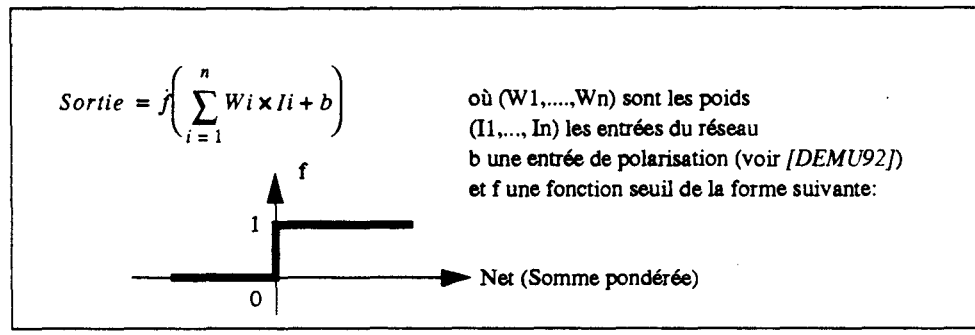


FIGURE 69 Formulation mathématique du calcul de la sortie d'un neurone

Un neurone à deux entrées peut par exemple permettre de classifier des données 2D fournies en entrée en deux classes linéairement séparables comme le montre graphiquement la figure suivante:

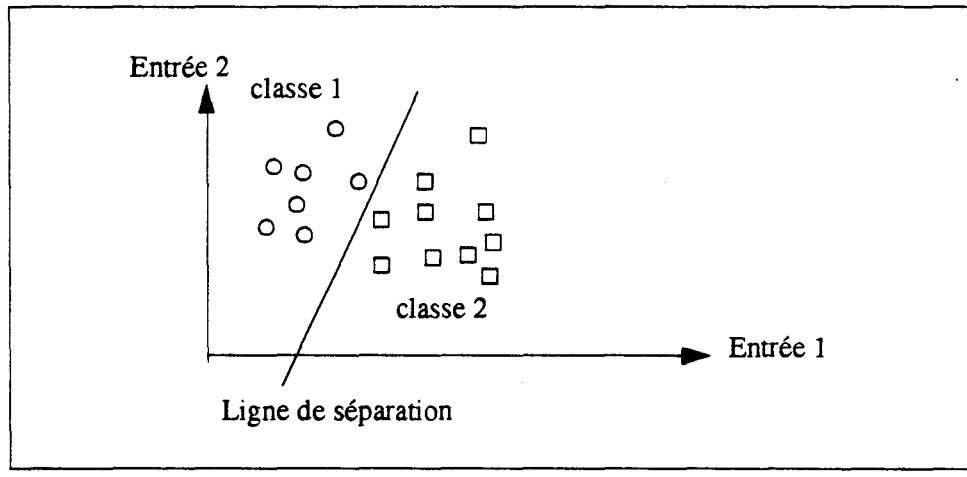


FIGURE 70 Problème à 2-dimensions linéairement séparable

Suivant la complexité de modélisation du réseau et afin de se rapprocher des observations biologiques, d'autres fonctions seuils sont couramment utilisées: il s'agit des fonctions **sigmoïdales** dont un exemple est fourni ci-après. Il faut alors redéfinir les entrées car la sortie n'est plus binaire.

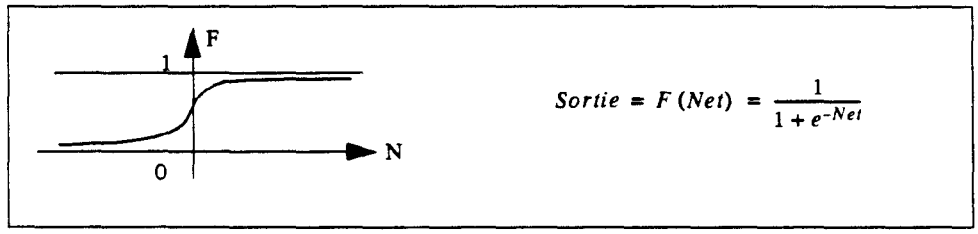


FIGURE 71 Fonction sigmoïdale

Un réseau de neurones **multicouches** est comme son nom l'indique organisé en plusieurs couches successives de 1 ou plusieurs neurones interconnectés, ayant une couche d'entrée, une couche de sortie et 1 ou plusieurs couches intermédiaires, également appelées **couches cachées**. Un exemple typique de réseaux multicouches est donné sur la figure suivante:

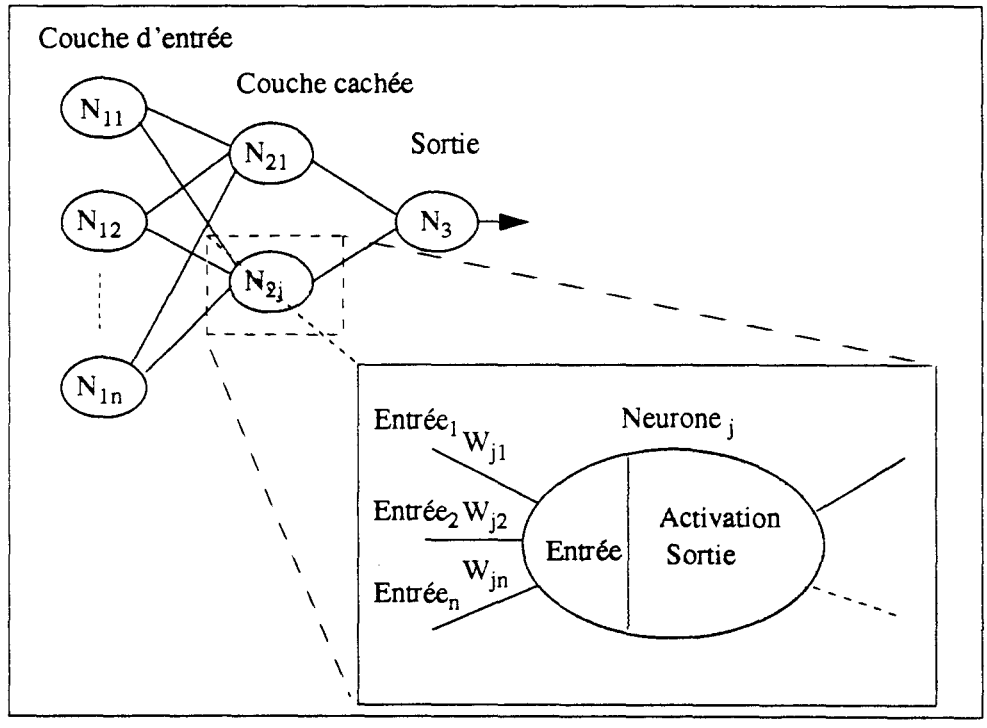


FIGURE 72 Exemple de réseau neuronal

Les réseaux de neurones sont caractérisés par plusieurs étapes:

- Une **phase d'apprentissage** qui vise à établir les valeurs des poids des connexions de telle sorte que le réseau soit capable de réagir correctement à un problème particulier. Nous allons résumer dans la partie qui suit diverses techniques d'apprentissage couramment utilisées.

- Une **phase de test** qui consiste à observer comment le réseau réagit à l'apport en entrée de nouvelles données jamais apprises.

- Une **phase d'exploitation** ou d'utilisation.

IV.2.2.1.2 Les algorithmes d'apprentissage

Les techniques d'apprentissage des réseaux de neurones peuvent se décomposer en deux grandes familles: les apprentissages **supervisé** et **non supervisé**.

1. L'apprentissage supervisé

Il consiste à présenter en entrée du réseau un certain nombre d'exemples et à comparer la sortie donnée par le réseau avec le résultat souhaité. De cette comparaison résulte un ajustement des poids des connexions selon une règle d'apprentissage de sorte que lors d'une présentation future de ces exemples la sortie donnée par le réseau corresponde davantage au résultat escompté. La phase d'apprentissage s'achève lorsque la différence entre le résultat calculé en sortie et celui attendu (ou erreur en sortie) est passée en deça d'un seuil fixé ou que le réseau classifie correctement les exemples.

Parmi les algorithmes d'apprentissage supervisé les plus populaires, on peut trouver ceux du «Perceptron» (voir *[ROSE61]* et *[BLOCK]* pour une description détaillée) et surtout «L'algorithme de Rétro-Propagation» (voir *[RUME86]*).

Le «Perceptron» a longtemps été utilisé pour sa simplicité puis abandonné (du moins dans sa forme la plus simple) pour son incapacité à résoudre certains problèmes de classification de vecteurs non linéairement séparables. C'est le cas par exemple de l'implémentation de la fonction XOR (OU exclusif) expliquée dans *[MINS69]*.

L'algorithme de «Rétropropagation» est certainement à l'heure actuelle le plus utilisé dans les applications des réseaux de neurones. L'idée de la règle d'apprentissage est d'ajuster les poids des connexions du réseau en propageant à l'envers du réseau l'erreur de sortie selon une descente de gradient. L'erreur à minimiser s'écrit sous la forme:

$$E_p = \frac{1}{2} \times \sum (O_{pi} - T_{pi})^2$$ où O_{pi} est la sortie du neurone i sur la couche de sortie du réseau

T_{pi} est la ième valeur de sortie attendue pour l'exemple p

FIGURE 73 Erreur en sortie du réseau

La règle d'apprentissage correspondante s'écrit de la façon suivante:

$$\Delta W_{ij} = \alpha\left(\frac{\partial E_p}{\partial W_{ij}}\right) = \eta \times \delta Lpi \times O_{Lpi}$$

E_p est l'erreur explicitée précédemment

ΔW_{ij} est la variation à affecter au poids Wij correspondant à l'ajustement de l'exemple p

η est le taux d'apprentissage

δLpi est l'erreur en sortie de l'unité i sur la couche L pour l'exemple p

FIGURE 74 La règle d'apprentissage de la rétro-propagation

Bien que l'algorithme de Rétro-propagation ait prouvé son efficacité dans de nombreuses applications, quelques difficultés sont souvent rencontrées dans la mise en place d'un tel algorithme, ceci en raison des points suivants:

- La fonction d'erreur d'un réseau de neurones peut avoir plusieurs minima. L'apprentissage est ainsi susceptible de s'arrêter dans un minimum local qui n'est pas la meilleure solution du problème car un minimum global donnerait une meilleure minimisation de l'erreur.

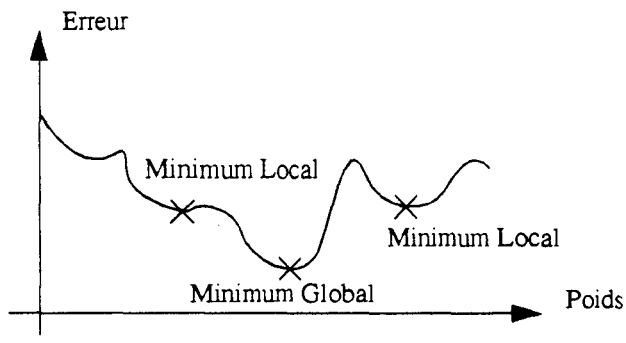


FIGURE 75 Exemple de courbe d'erreur d'un réseau neuronal

- Trouver le taux d'apprentissage approprié à un réseau non-linéaire reste une tâche difficile. D'une part, une valeur trop forte peut conduire à un apprentissage oscillant car dans ce cas, même si les minimas locaux peuvent être évités en passant par dessus, le système est susceptible de tourner autour d'une minimum global sans jamais l'atteindre. D'autre part, un taux trop faible peut conduire à un temps d'apprentissage incroyablement long.

- Les réseaux de neurones sont sensibles au nombre de neurones de leurs couches cachées. Trop peu de neurones empêchent le réseau de répondre correctement à un problème, alors que trop de neurones vont bien classifier les exemples mais peut être mal généraliser dans la phase d'utilisation et utiliser les ressources de calcul plus que nécessaire.

- Trouver les valeurs initiales adéquates pour les poids des neurones n'est pas toujours simple et peut avoir une grande influence sur le temps d'apprentissage.

Quelques nouvelles méthodes sont apparues récemment pour couvrir certaines de ces lacunes et sont applicables sur l'algorithme de base de la Rétro-propagation *[VOGL88] [NGUY90]*. Nous les utiliserons pour améliorer l'efficacité et la compacité des réseaux que nous allons détailler en vue d'une implantation dans la carte à micro-processeur.

2. L' apprentissage non supervisé

Dans la section précédente, nous avons vu que le mécanisme de rétro-propagation avait besoin d'un certain nombre d'exemples donnant la sortie attendue par rapport à une entrée particulière, de sorte que le réseau puisse attribuer des coefficients aux poids des neurones pour répondre à un problème spécifique..

Les procédés d'apprentissage non supervisés n'ont pas besoin qu'un «Professeur» leur fournisse de tels exemples. Les neurones apprennent à reconnaître et regrouper dans de mêmes classes des données présentant des similitudes en entrée. De tels réseaux détectent les corrélations dans leurs entrées et adaptent leurs réponses futures en conséquence. Le travail de Kohonen sur ce sujet a débouché sur plusieurs modèles très intéressants basés sur des résultats d'expérimentations biologiques *[KOHO87]*. Deux remarques sont à la base de ces travaux:

- Chaque cellule du réseau neuronal semble répondre à un stimulus spécifique. Cela implique pour la modélisation informatique de définir un réseau où chaque neurone d'une couche interne doit répondre à une entrée particulière.

- Il existe un mécanisme d'interaction latérale entre neurones voisins, qui peut être modélisé par la fonction suivante:

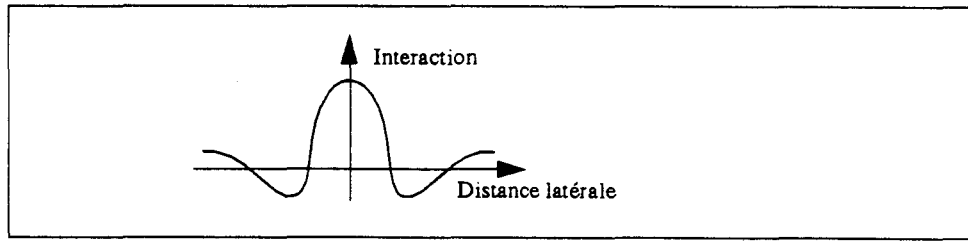


FIGURE 76 Interaction latérale entre neurones.

Pour une description complète de la règle d'apprentissage de Kohonen, se référer à *[KOHO87]* ou *[DAVA93]*.

Nous utiliserons ces résulats et algorithmes d'apprentissage pour optimiser des systèmes d'identification biométrique, notamment sur le procédé de reconnaissance de la signature comportementale du clavier.

Les Cartes Auto-Organisatrices de Kohonen basées sur l'apprentissage non supervisé ont déjà été appliquées à des problèmes de reconnaissance de forme et d'identification biométrique, tels que ceux décrits dans *[GROS91]* et *[SHEP94]*.

Conclusion

Nous avons vu dans cette introduction que les applications utilisant des réseaux de neurones étaient divisées en 2 phases:

- Une phase d'apprentissage pour établir les poids du réseau de façon à répondre correctement à un jeu d'exemples.
- Une phase d'interrogation correspondant à l'utilisation du réseau.

De là découlent plusieurs options pour intégrer des réseaux de neurones dans les applications cartes à puce. Une première consiste à ne mettre dans la carte que le processus d'utilisation du réseau, beaucoup moins gourmand en calculs que la phase d'apprentissage. Une seconde stratégie vise à intégrer l'ensemble dans la carte. Cette dernière option apporte à la fois des avantages et inconvénients que nous soulignerons quand elle sera envisagée dans les implémentations qui suivent.

La technologie neuronale n'est apparue que récemment. Etant donné que les fondements mathématiques associés et procédures de design d'applications n'ont toujours pas été fermement établis, la mise en place d'applications est encore basée principalement sur des expérimentations. Nous allons détailler dans les parties suivantes des implémentations possibles des applications envisagées précédemment (utilisant les langages de haut niveau) sous forme de réseaux de neurones.

Bien que de nombreuses configurations de réseaux et méthodes d'apprentissage soient disponibles, nous essayerons de justifier nos intuitions, en vue d'apporter au monde de la carte les outils nécessaires au développement d'applications cartes utilisant les réseaux de neurones.

IV.2.2.2 La Compression de Données par les Réseaux Neuronaux

Si pour le moment le standard JPEG est très largement utilisé dans les applications multimédia, d'autres méthodes de compression non conservatives de l'information sont à l'étude et notamment certaines basées sur la technologie neuronale.

Sans avoir construit notre propre version neuronale de compression en vue d'une implémentation dans la carte à puce comme nous avons fait pour JPEG dans la partie précédente , nous avons étudié plusieurs systèmes neuronaux de compression comme exemple de traitement multimédia associé à la carte et utilisant pour noyau commun des réseaux de neurones.

IV.2.2.2.1 Le principe de la compression neuronale d'une image

L'idée consiste à réduire la dimensionnalité de l'image à compresser en utilisant un réseau neuronal multi-couches. La couche cachée du réseau présentant un nombre de neurones inférieur à celui de la couche d'entrée ou de sortie, il en résulte une compression des données, comme le montre par exemple la figure suivante:

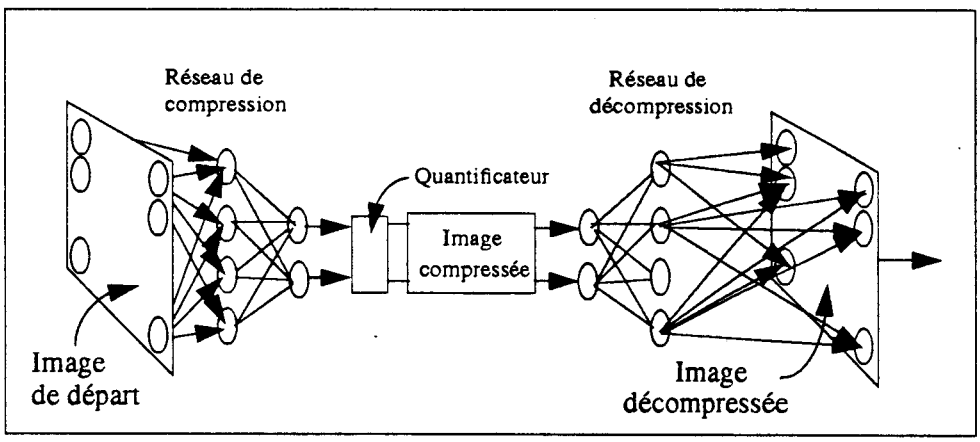


FIGURE 77 La compression utilisant les réseaux de neurones

Généralement on utilise l'algorithme de rétropropagation pour la phase d'apprentissage qui consiste à fournir en entrée du réseau un certain nombre d'images à compresser *[AROZ90]*. La règle apprise par le premier étage du réseau représente une transformation ayant pour but de supprimer la redondance des informations présentes dans l'image. Le second étage, celui de décompression, a pour objectif la transformation inverse nécessaire à la reconstruction de l'image. Un étape de quantification effectuée dans la couche cachée du réseau permet de réduire de façon sensible la taille des données. Ces différentes étapes de compression interviennent dans *[SONE89]*.

Ce type de compression reste d'ailleurs assez proche de la compression JPEG dans la mesure ou la transformation initiale de l'image au travers du réseau de neurones peut être vue comme la Transformée Cosinus Discrète appliquée dans l'algorithme JPEG. Les 2 méthodes comportent ensuite toutes les deux une phase de quantification de données avant d'obtenir l'image finale compressée.

IV.2.2.2.2 Une approche de compression de données utilisant les Cartes Auto-Organisatrices de Kohonen

Dans cette approche qui connaît à l'heure actuelle une attention soutenue, c'est l'algorithme d'apprentissage non supervisé de Kohonen (décrit succinctement dans l'introduction sur les réseaux de neurones) qui est utilisé pour créer un dictionnaire de blocs de référence (par exemple des blocs de 3x3 pixels). La compression consiste alors à remplacer chaque bloc de l'image par l'index du bloc du dictionnaire qui s'en rapproche le plus. Un exemple de compression d'une image de télévision figure dans *[BURE91]* et permet de coder l'image source à l'aide de 256 blocs qui ont été déterminés par l'algorithme de Kohonen.

Le détail théorique de la constitution d'un dictionnaire à partir de la règle d'apprentissage de Kohonen est repris dans *[NASR88]*, où il est également comparé avec d'autres méthodes.

Conclusion

Sans avoir effectué de développement spécifique autour de ce mode de compression, nous avons succinctement décrit des méthodes de compression neuronale afin de montrer que, en plus des puissantes capacités de traitement dans le domaine de la biométrie ou de l'identification comportementale (comme nous allons le voir par la suite), les réseaux de neurones sont capables de compresser des informations de type images ou sons. Cette particularité est intéressante car elle permet , en utilisant de mêmes types d'algorithmes (Rétro-propagation, Cartes de Kohonen), d'effectuer à la fois des manipulations dédiées à l'identification ou plus généralement à la reconnaissance de formes d'une part et le stockage d'objets multimédia dans la carte d'autre part, limitant ainsi le nombre de fonctions de traitement disponibles à quelques primitives communes.

IV.2.2.3 La Signature Comportementale du Clavier utilisant des techniques d'Auto-Organisation et d'apprentissage supervisé

IV.2.2.3.1 La Signature Clavier utilisant l'algorithme de rétro-propagation

Nous avons expérimenté la phase de Vérification de la Signature Comportementale du Clavier détaillée précédemment en introduisant la technologie neuronale. Etant donné que l'algorithme de base de Rétro-propagation est marqué par plusieurs inconvénients (temps d'apprentissage trop long dépendant des valeurs initiales *[KOLE91]*, minima locaux etc...), nous avons implémenté les variantes de l'algorithme proposées par *[VOGL88]* et *[NGUY90]*.

Le réseau que nous avons utilisé est constitué de 64 entrées (+1 de polarisation) possédant une couche cachée de 5 neurones. Il a été entraîné par l'algorithme de rétropropagation sur un jeu de 10 exemples. Les 64 entrées correspondent dans le cas présent aux fréquences d'apparition de paires (N_i, N_{i+1}) (apparitions de 2 touches successives).

La première couche comprend donc 5 neurones dont le seuil est une fonction sigmoïdale, et la seconde est faite d'un neurone linéaire. Le schéma suivant représente le réseau global considéré:

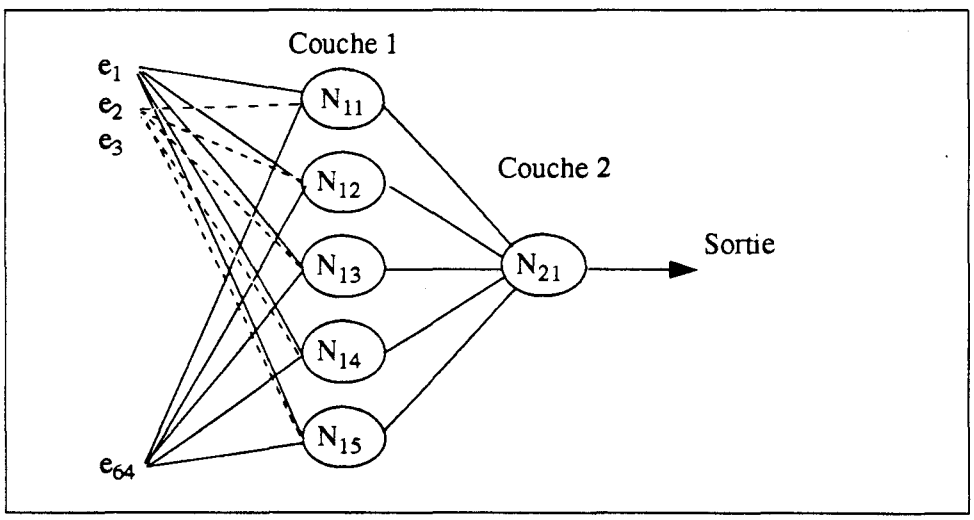


FIGURE 78 Réseau de reconnaissance de la Signature Clavier

Le réseau à été initialisé avec la méthode de Nguyen-Widrow *[NGUY90]*. Le Taux d'apprentissage adaptatif qui cherche à être le plus fort possible tout en gardant des pas d'apprentissage raisonnables accélère la convergence de l'apprentissage en évitant les brutales augmentations de l'erreur. La technique du Moment pour éviter d'être piégé dans un minimum local a aussi été intégrée.

Nous avons décrit le réseau et l'algorithme d'apprentissage en utilisant le langage de programmation de Matlab *[DEMU92] [NAZA92]*. Un résultat typique d'apprentissage pour un individu ayant fourni sa Référence (10 séquences de 100 chiffres) est donné ci-après:

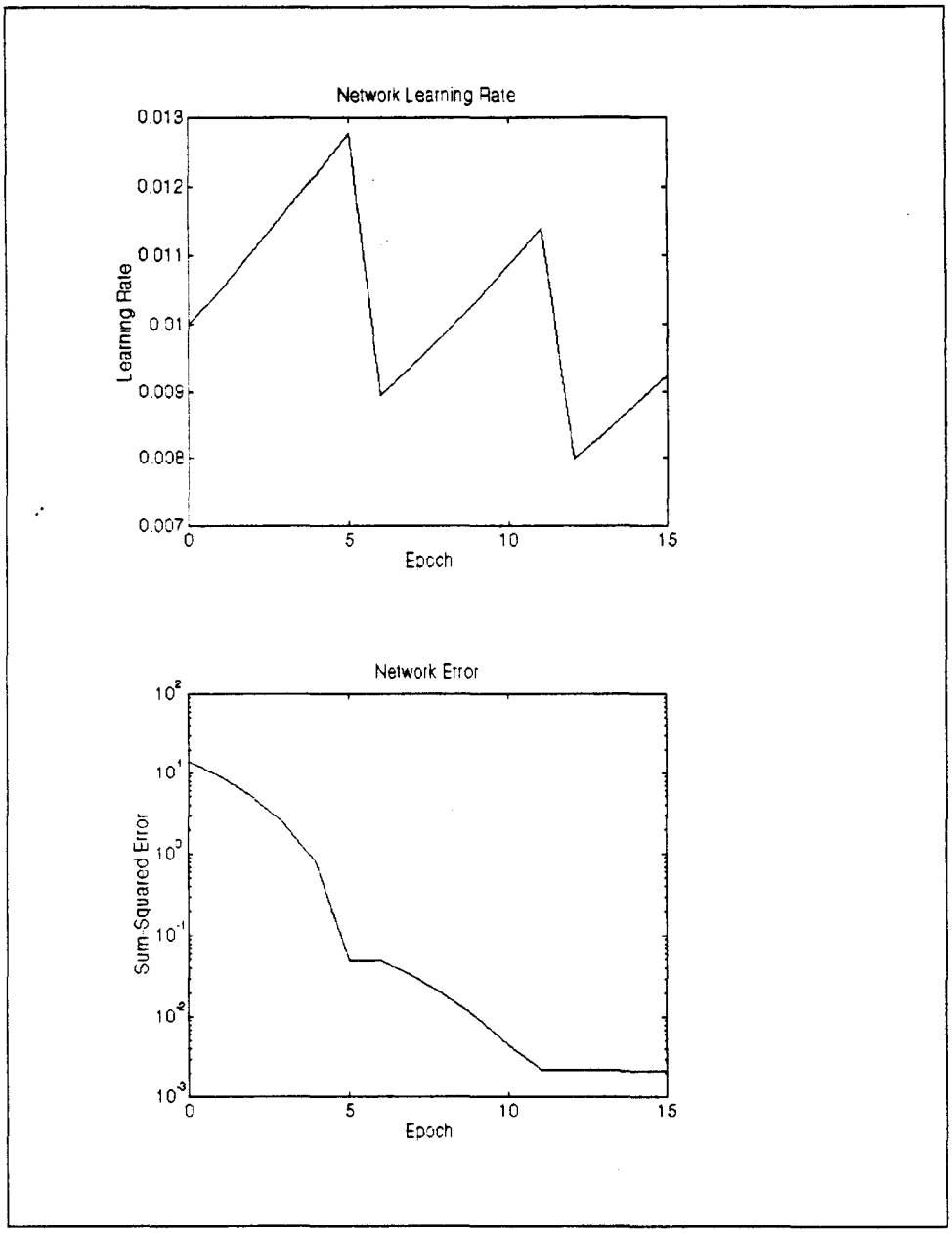


FIGURE 79 Résultats de l'apprentissage pour un individu

Le jeu de 10 exemples est présenté plusieurs fois jusqu'à ce que l'erreur du réseau soit inférieure à un certain seuil. Ici 11 présentations des exemples (aussi appelées **Epochs**) ont été suffisantes pour rendre l'erreur inférieure à 0.0023. L'erreur commise par chacun des exemples à la fin de l'apprentissage si l'on s'arrête à une erreur globale de 0.0023 est alors la suivante:

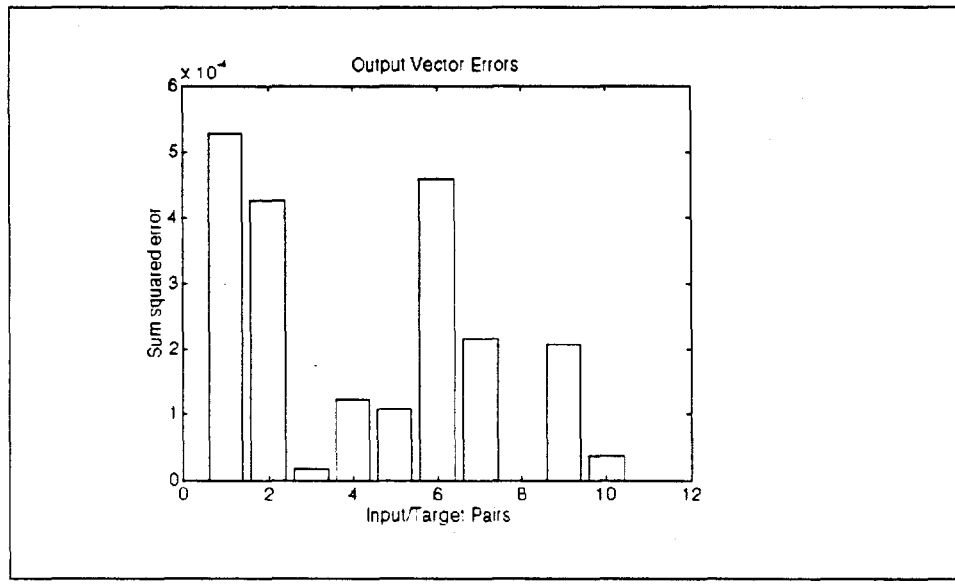


FIGURE 80 Erreur commise par chaque exemple à la fin de l'apprentissage

La phase d'apprentissage a duré environ 2 secondes sur un micro-processeur de type 80486 cadencé à 50 MHz. Des réseaux entraînés de ce type ont été testés à la fois sur les utilisateurs propriétaires des Références et d'autres inconnus afin de mesurer les TVR et TFA associés au système (voir rappels du chapitre II). Aucun des tests n'a vu de vrai utilisateur échouer à l'identification ni de fraudeur réussir à s'introduire. Cependant, ces tests ont été limités à un ensemble restreint de personnes et devraient être effectués sur une population plus vaste et plus représentative.

Pour le moment de telles bases de données n'existent pas comme pour des systèmes d'identification plus classique, tels que l'identification de la voix ou des empreintes digitales. De ce fait il est difficile d'établir une comparaison avec tout autre procédé d'identification biométrique disponible à l'heure actuelle sur le marché.

Evaluation de la taille de la Référence

Nous avons calculé l'espace requis pour le stockage du réseau en mémoire. Etant donné que l'on veut pouvoir stocker la Référence dans une carte à puce, il est essentiel que celui-ci reste de taille modeste.

Dans notre approche nous avons utilisé 65 entrées (64+1 de polarisation) connectées à 5 neurones dans la première couche. En admettant que les coefficients des poids soient des nombres réels stockés sur 4 octets, la première couche doit disposer en mémoire de 65*5*4 = 1300 octets. La seconde couche peut être décrite de la même façon (5 neurones+1 polarisation reliés à 1 neurone linéaire) par 6*1*4 = 24 octets, donnant ainsi une taille totale de 1324 octets.

Puisque 1.27 Koctets est relativement important si l'on veut implémenter la phase de vérification dans la carte, nous avons utilisé dans ce qui suit les cartes Auto-Organisatrices de Kohonen, afin d'extraire des 64 entrées seulement celles qui sont le plus significatives pour le processus d'identification (certaines paires (N_i,N_{i+1}) sont plus invariantes que d'autres). Dans ce sens, l'apport de Kohonen fait acte de pré-traitement sur le système. Il permet de réduire la dimensionnalité du problème *[KOHO87]*.

IV.2.2.3.2 L'apport des Cartes Auto-Organisatrices

Dans cette approche, on considère que chaque entrée est connectée à tous les neurones de la couche suivante, laquelle se compose d'un ensemble de neurones répartis sous la forme d'une carte, comme le montre le schéma suivant:

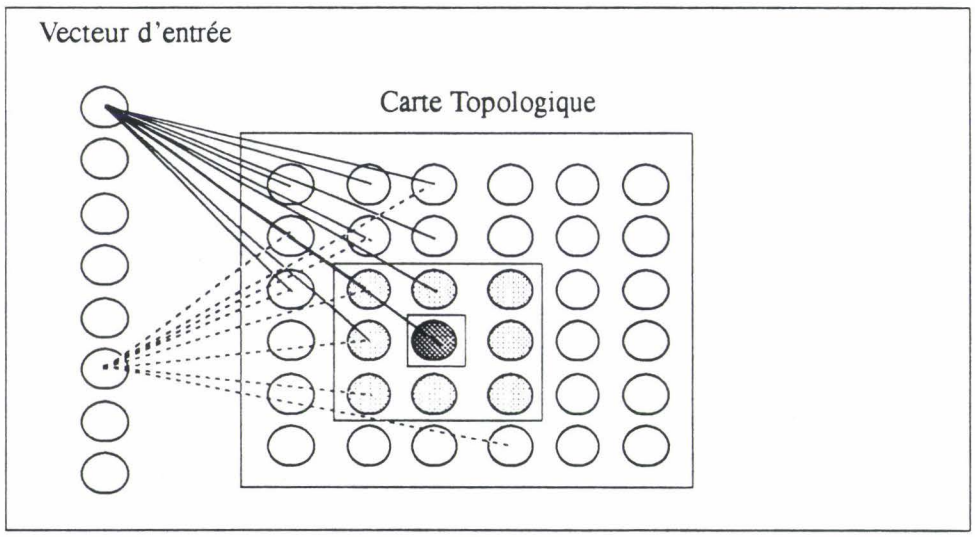


FIGURE 81 Carte Auto-Organisatrice de Kohonen

Chaque neurone de la carte possède le même nombre de voisins (sur le schéma nous avons étalé la carte, normalement l'extrémité droite de la carte est reliée à l'extrémité gauche, et le haut au bas).

Le mécanisme d'auto-adaptation comporte alors deux étapes:

- La sélection du neurone qui doit correspondre à un type de signal d'entrée donné. Pour cela on va comparer le vecteur d'entrée aux vecteurs des poids des connexions de chacun des neurones de la carte avec la couche d'entrée, comme suit:

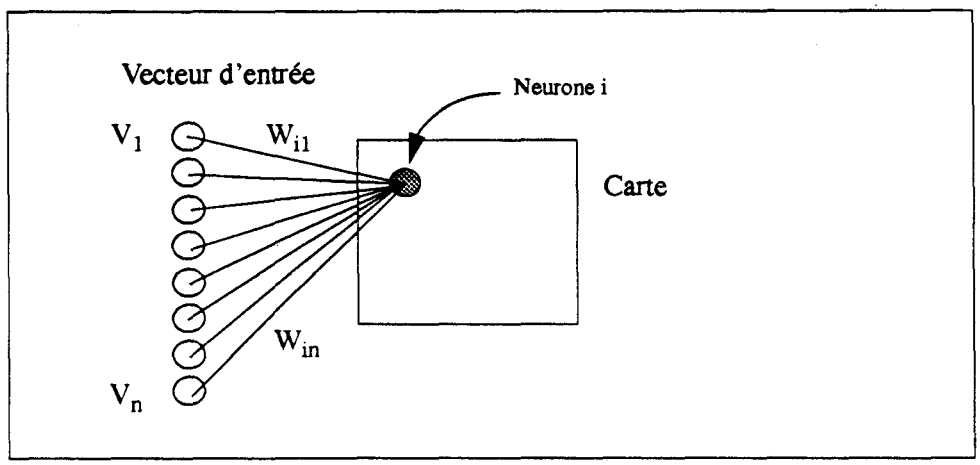


FIGURE 82 Sélection du Neurone i tel que Wi ressemble le plus possible à V

- Le renforcement de cette adéquation pour ce neurone et ses voisins. Consulter *[KOHO87]* pour le détail de la règle d'apprentissage.

Si l'on en revient au problème de reconnaissance de la signature comportementale du clavier qui nous intéresse dans cette partie, les cartes de Kohonen vont permettre de ne retenir parmi les corrélations des paires (N_i,N_{i+1}) fournies en entrée du réseau uniquement celles qui sont pertinentes pour l'identification.

IV.2.2.3.3 Améliorations du procédé de reconnaissance de la signature clavier

Sans avoir effectué de réalisation pratique et de tests spécifiques à ces améliorations, les remarques qui suivent représentent des suggestions quant à l'optimisation de la fiabilité du système:

- On peut imaginer le traitement en temps réel de la reconnaissance étant donné la simplicité des données et des calculs. Le comptage des fréquences d'apparition de chaque touche, de paires, voire de triplets, peut facilement s'effectuer à la volée au fur-et-à-mesure de l'acquisition de la séquence.
  Ainsi à partir d'un nombre de données minimum (une séquence de 100 chiffres par exemple), le système pourrait émettre un score après passage dans le réseau

de neurones. Si à cet instant ce score correspond suffisamment à celui de la Référence, l'acquisition des données n'est plus nécessaire. Si ce score diffère considérablement de la Référence, là aussi la phase de reconnaissance est interrompue et rejète l'individu. En revanche, dans le cas d'un litige où le système n'est pas tout à fait sûr de bien reconnaître l'individu, il peut continuer à enregistrer quelques données supplémentaires correspondant à une chance supplémentaire donnée à l'utilisateur pour s'identifier.

- Dans notre approche nous nous sommes limités à la saisie de données provenant de 8 touches du clavier choisies préalablement. Une idée intéressante serait de laisser à l'utilisateur le choix d'une position qu'il juge confortable pour taper sa séquence. La seule condition à respecter est alors de ne pas changer le positionnement de ses doigts une fois que la saisie des données commence. Ce confort d'utilisation suppose pour le système de savoir au départ quelle touche correspond à quel doigt. Ce petit traitement supplémentaire peut très bien se faire en dehors de la carte. Le reste du système de reconnaissance n'est pas du tout affecté par cette modification.

IV.2.2.3.4 Optimisation par de nouveaux algorithmes

Etant donnés les problèmes rencontrés dans l'utilisation de l'algorithme de rétro-propagation (convergence parfois lente, minima locaux, valeurs initiales, taille du réseau (voir introduction sur la Rétro-propagation dans ce chapitre)), certains chercheurs se sont intéressés à l'optimisation de ce type d'algorithmes supervisés soit par modification de la règle d'apprentissage soit par changement radical dans l'approche de l'apprentissage en termes d'architecture et de traitement.

Pour la première catégorie, nous avons retenu l'algorithme «Quickprop» *[FAHL88]*, qui permet en général d'améliorer d'un facteur 1,5 la vitesse d'apprentissage (voir l'étude comparative de *[NAZA92]*.)

Dans l'environnement cartes à puce, la taille du réseau a une importance capitale à la fois pour le stockage des coefficients et pour le traitement qui s'y rapporte. La Rétropropagation classique suppose que l'on connaisse par avance la taille du réseau adaptée au problème traité, conduisant souvent à des apprentissages difficiles par manque ou par excès de connexions.

C'est pourquoi notre regard s'est porté sur de nouvelles architectures de réseaux et en particulier celle de «Cascade Correlation» élaborée par Fahlman *[FAHL90]*. Au lieu d'ajuster les poids des connexions d'un réseau de topologie fixe, «Cascade-Correlation» commence par un réseau de taille minimale (une seule unité cachée), puis rajoute au fur-et-à-mesure de l'apprentissage de nouvelles unités suivant les besoins. Ce nouveau type d'architecture possède plusieurs avantages par rapport aux algorithmes existants:

- L'apprentissage est très rapide
- Le réseau détermine lui-même sa propre topologie
- il n'y a pas besoin de rétro-propager dans les connexions du réseau le signal d'erreur obtenu en sortie.

L'algorithme de «Cascade-Correlation» combine deux idées maîtresses:

1. **Une architecture en cascade,** dans laquelle les unités cachées sont ajoutées une par une et ne changent pas après avoir été construites.

2. **Un procédé d'apprentissage efficace et original,** qui crée chaque nouvelle unité cachée en corrélation avec l'erreur en sortie que l'on doit éliminer.

L'architecture est illustrée sur le schéma suivant:

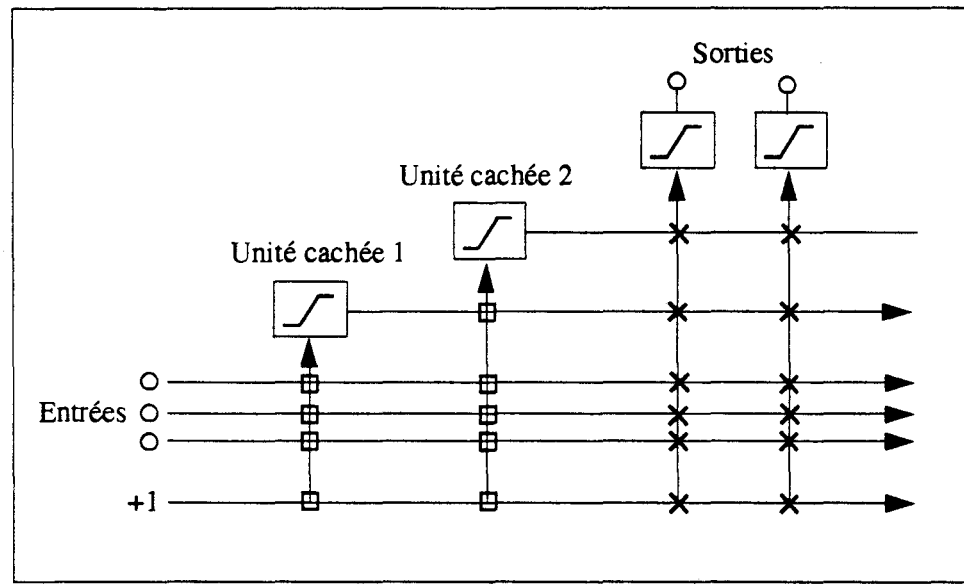


FIGURE 83 Architecture «Cascade-Correlation», après l'ajout de 2 unités cachées

Le détail de l'algorithme d'apprentissage associé est fourni dans *[FAHL90]*. Nous pensons qu'un algorithme de ce type serait très approprié aux applications cartes, spécialement si l'on envisage d'effectuer l' apprentissage en interne de la carte. Une étude des propriétés des architectures cascadées telles que celle décrite ci-dessus figure dans *[LITT93]* et révèle une bonne capacité à la généralisation sur des données jamais encore apprises. En outre, l'algorithme de «Cascade-Correlation» peut être utilisé avec des coefficients dont la précision numérique est limitée,

ce qui contribue à réduire la taille nécessaire au stockage du réseau ainsi que les traitements sans pour autant y perdre de façon significative dans les résultats d'utilisation du réseau: une étude sur la précision numérique est détaillée dans *[HOEH91]*.

IV.2.2.4 Une implémentation de Radar dans la carte à l'aide de Réseaux Neuronaux

Nous avons étudié et réalisé une implémentation de Radar à base de réseaux neuronaux, utilisant un apprentissage supervisé de type Rétro-propagation. Nous évoquons dans un deuxième temps d'autres orientations possibles.

IV.2.2.4.1 Un Radar basé sur l'algorithme de Rétro-propagation

Les entrées sont la description des transactions elle-mêmes à l'aide des paramètres <Nature, Montant, Date, Lieu>. Afin de donner au réseau un paramètre significatif relatif aux dates, nous fournissons en entrée du réseau (comme pré-traitement) le *Montant/Semaine* et le *Nombre de transactions/Semaine* se rapportant à la *Nature* de transaction considérée au lieu de la *Date* elle-même.

Des *Montants* ou *Nombres de Transactions* sont des nombres réels donc comparables. En revanche, on ne peut comparer des *Natures* différentes telles que *Cash* ou *Transports*. C'est pourquoi nous ne pouvons pas décrire directement une *Nature* de transaction à l'aide d'une seule entrée.

Nous avons donc appliqué une technique de codage binaire souvent utilisée pour répondre à ce type de problème en affectant à chaque *Nature* possible une entrée booléenne correspondante. Le même problème concerne l'implémentation du *Lieu* de transaction.

Nature et Lieu sont ainsi codés en entrée du réseau sous la forme de 8 entrées correspondant aux 8 natures et 2 entrées correspondant à la Loc-Table, comme suit:

Nature	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
Cash	1	0	0	0	0	0	0	0
Transports	0	1	0	0	0	0	0	0
Essence	0	0	1	0	0	0	0	0
Nourriture	0	0	0	1	0	0	0	0
Logement	0	0	0	0	1	0	0	0
Ameublement	0	0	0	0	0	1	0	0
Loisirs	0	0	0	0	0	0	1	0
Autres	0	0	0	0	0	0	0	1

Lieu	e_{12}	e_{13}
Dans Loc-Table	1	0
Lieu inconnu	0	1

FIGURE 84 Codage binaire associée aux paramètres *Nature* et *Lieu*

Le jeu d'exemples pour l'apprentissage a été composé à partir d'un mélange de transactions effectuées par le propriétaire de la carte de crédit durant 6 mois et de transactions frauduleuses constituées par sondage, comme expliqué dans le chapitre III.

Le réseau, possédant au total 13 entrées (+1 de polarisation), est composé de deux couches, l'une ayant 5 neurones à seuil sigmoïdal, la suivante 1 neurone linéaire. Il est initialisé avec la méthode de Nguyen-Widrow *[NGUY90]*, et utilise les variantes accélératrices de la Rétro-propagation expliquées pour la signature comportementale du clavier (section précédente).

Le réseau se présente de la manière suivante:

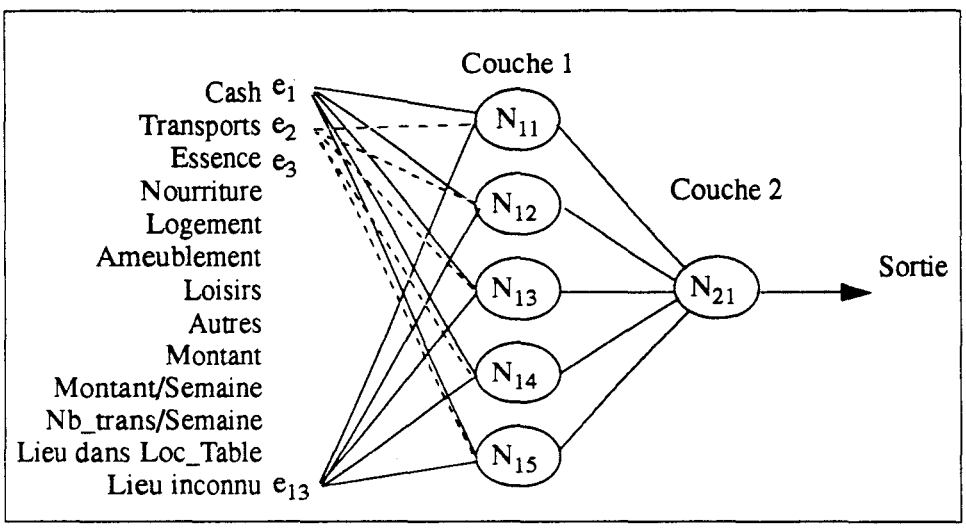


FIGURE 85 Exemple de Radar sous forme de réseau de neurones

Nous avons d'abord donné comme condition d'arrêt de l'apprentissage une erreur en sortie du réseau inférieure à une valeur particulière (0.02 par exemple). Nous avons finalement réalisé que la phase d'apprentissage pouvait s'interrompre lorsque l'erreur en sortie pour chacun des exemples était inférieure à 0.5, laquelle correspond à la différence entre le résultat donné par un exemple en sortie du réseau et le résultat souhaité. Cette condition est suffisante pour classifier correctement les exemples , en disant qu'une sortie supérieure à 0.5 sera interprêtée 1 et inférieure à 0.5 sera interprêtée 0, comme expliqué dans *[FAHL88]*.

Les courbes de la page suivante décrivent les résultats de l'apprentissage du réseau pour un cas particulier d'individu sur la base de 28 vecteurs possédant chacun 13 entrées.

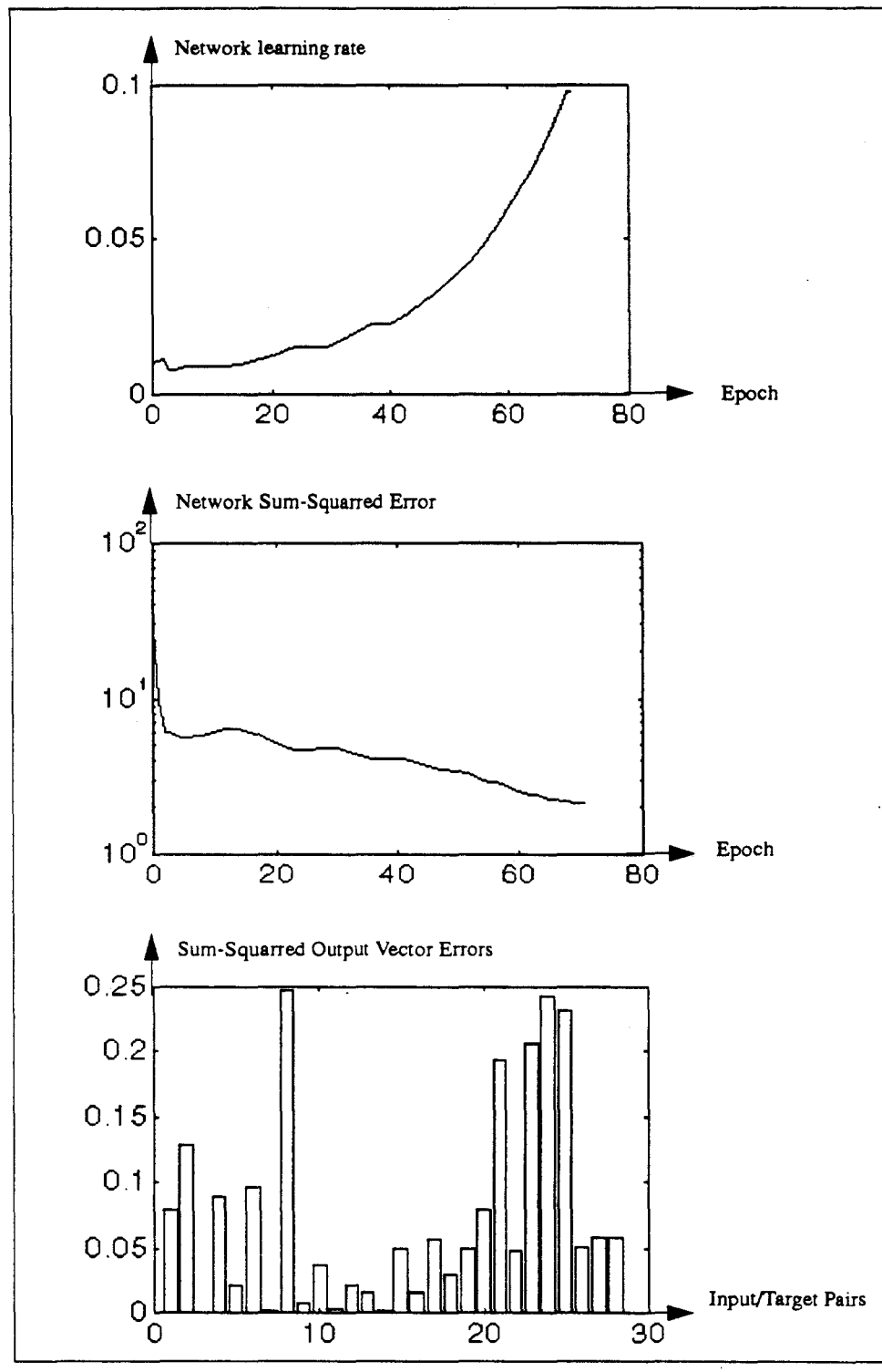


FIGURE 86 Résultats de l'apprentissage supervisé d'un Radar

Il a fallu 71 itérations des exemples pour classifier correctement les données, c'est-à-dire environ 3 secondes sur un micro-processeur 80486 cadencé à 50 MHz.

Nous avons testé ce réseau nouvellement entraîné sur de nouvelles données encore jamais apprises, à la fois des transactions effectuées par le propriétaire de la carte et d'autres de type frauduleux, donnant un taux de détections correctes de l'ordre de 90%. Cette évaluation toutefois n'a concerné qu'un ensemble restreint de données afin de donner une idée de son intérêt pour des applications de ce type, mais devrait être conduite sur un éventail d'exemples beaucoup plus large pour donner des chiffres plus précis et fiables quant aux performances.

Nous avons estimé la taille mémoire requise par ce Radar neuronal en vue de son intégration dans une carte à puce. Le procédé encarté ne concerne pas l'apprentissage dans un premier temps (trop couteux pour une carte actuelle). La taille mémoire nécessaire au stockage du réseau représente 13 x 5 valeurs réelles des poids des neurones de la première couche + 5 de polarisation et 5 poids +1 polarisation pour la seconde couche. Au total, en codant chaque valeur réelle sur 4 octets, le réseau occupe 304 octets en mémoire EEPROM.

Pour évaluer la taille de l'algorithme d'utilisation de ce réseau, nous nous sommes référés à des études effectuées dans le projet européen ESPRIT-CASCADE EP8670, introduit dans *[PEYR94]* et *[CASC94]*. Nous avons estimé qu'un Radar tel que décrit précédemment et appliqué sur un processeur encartable de type celui de Cascade consommerait seulement 300 octets de ROM et moins de 30 octets de RAM, une réponse du réseau associée à une transaction se faisant environ en 5000 cycles d'horloge. Ces chiffres en font un système peu consommateur en ressources (voir les tailles mémoires et puissances de calcul typiques des cartes à puce dans le chapitre I).

IV.2.3 Les primitives de traitement des données

A partir des études et réalisations effectuées dans les deux premières sections de ce chapitre, nous allons synthétiser dans cette partie un certain nombre de primitives servant de dénominateur commun au traitement des données multimédia envisagées. Le but de ces traitements internes ou externes vise des objectifs précis qui sont:

- La reconnaissance
- L'identification
- La sélection de type associatif

- La production d'alertes
- La recherche de points singuliers
- La compression et la décompression de séquences

Les premières commandes qui vont être présentées ne relèvent pas d'une application précise. Elles constituent les primitives essentielles à la manipulation des objets de type séquences. Elles sont peu nombreuses et peuvent se combiner avec d'autres pour constituer de véritables réponses à des applications.

IV.2.3.1 La manipulation des séquences

IV.2.3.1.1 La gestion des Listes

On se propose de limiter l'accès aux éléments de type Liste (définis dans la structuration des données de la partie précédente) à trois opérateurs de base qui sont:

- DEtach_Begin (DEB): cette fonction permet de détacher le premier élément d'une séquence.

Par exemple, si on a une Séquence de 3 N-uples définie par S1={a,b,c} et que l'on applique la fonction DEB à S1, on obtient:

S1.DEB()={a} et S1={b,c}.

- DEtach_End (DEE): de la même façon que précédemment, DEE sert à retirer le dernier élément de la Liste.

En ayant S1={a,b,c}, on obtient après application de la fonction DEE:

S1.DEE()={c} et S1={a,b}.

- APPend (APP): Il s'agit de la concaténation de 2 Listes.

Dans le cas ou S1={a,b,c} et S2={x,y}, l'opération de concaténation S1.APP(S2) donne:

S1={a,b,c,x,y} et S2={}.

Ces trois opérateurs de base permettent de réaliser un certain nombre de manipulations sur des séquences de type Liste. Des exemples applicatifs figurent dans les parties suivantes.

L'implémentation de ces opérateurs en C++ concerne la gestion des pointeurs définis dans la structuration des données. Elle permet d'éviter de déplacer ou recopier les données elle-même contenues dans les séquences de Vecteurs ou Matrices et ainsi limite les opérations de copie souvent consommatrices de temps. Ce dernier point est important dans un environnement de faible puissance comme une carte à puce. Il a d'ailleurs largement contribué à notre choix de représentation de données.

IV.2.3.1.2 Les opérateurs de calcul

Afin de réaliser des opérations globales sur les valeurs contenues dans les Vecteurs et Matrices, nous avons redéfini des opérateurs tels que l'addition, la soustraction, la multiplication ou la valeur absolue de Vecteurs ou de Matrices.

Une opération doit s'étendre à chacune des valeurs des N-uples quel qu'en soit le type. Elle comporte donc dans tous les cas la définition explicite de l'opération pour chaque élément du N-uple.

IV.2.3.2 Exemples applicatifs

IV.2.3.2.1 La dynamique de la signature manuscrite

Cet exemple a été étudié dans le chapitre II. Il s'agit d'extraire de la signature des paramètres dynamiques tels que la vitesse d'exécution ou l'accélération à certains endroits et non pas les caractéristiques graphiques du dessin de la signature.

Si l'on reprend la manipulation de séquences avec les primitives décrites précédemment, on peut facilement calculer les séquences de vitesse et d'accélération. Soit $S1=\{(x_1,y_1), (x_2,y_2), ...,(x_n,y_n)\}$ une séquence de N points de coordonnées (x,y). La séquence de vitesse peut être obtenue par l'enchaînement des primitives suivantes:

S2 = S1	S1 est copiée dans une nouvelle séquence S2
S1.DEB()	S1 possède alors N-1 éléments de 2 à N
S2.DEE()	S2 possède alors N-1 éléments de 1 à N-1
S_v = S1-S2	maintenant S1 représente la séquence des vitesses

Renouveler ces étapes conduit directement à la séquence d'accélération S_a.

Programme de comparaison

Le programme suivant donne une solution rudimentaire de la comparaison de deux signatures. REF représente la séquence de N points (X,Y) acquise pour la constitution de la Référence (VREF étant la vitesse et AREF l'accélération associées) et TEST représente la séquence de N points acquis pendant la phase de Test (avec VTEST et ATEST pour la vitesse et l'accélération respectivement).

```
List<integer> REF, VREF, AREF, TEST, VTEST, ATEST
/* Calcul de la séquence de Vitesse VREF de la Référence */
VREF = REF
VREF.DEB()
REF.DEE()
VREF = ABS(VREF-REF)

/* Calcul de la séquence d'accélération AREF de la Référence */
AREF = VREF
AREF.DEB()
VREF.DEE()
AREF = ABS(AREF-VREF)
```

Les calculs précédents peuvent se faire hors la carte, les deux séquences VREF et AREF seront stockées une fois pour toutes dans la carte.

La suite du programme est elle exécutée par la carte après que celle-ci ait fait l'acquisition de la séquence TEST.

```
/* Calcul de la séquence de Vitesse VTEST de la Référence */
VTEST = TEST
VTEST.DEB()
TEST.DEE()
VTEST = ABS(VTEST-TEST)

/* Calcul de la séquence d'accélération ATEST de la Référence */
ATEST = VTEST
ATEST.DEB()
VTEST.DEE()
ATEST = ABS(ATEST-VTEST)
```

*/*Comparaison par différence entre les deux séquences de vitesse et d'accélération*/*
VTEST = VTEST - VREF
ATEST = ATEST - AREF

IV.2.3.2.2 Un exemple de réseau neuronal

Le réseau suivant est un cas d'école très rudimentaire. Il s'agit d'un neurone simple ayant 4 entrées. Nous avons manipulé les objets définissant le neurone pour effectuer l'apprentissage de ce réseau d'une part en utilisant la règle du Perceptron et d'autre part avec la règle de Rétro-propagation.

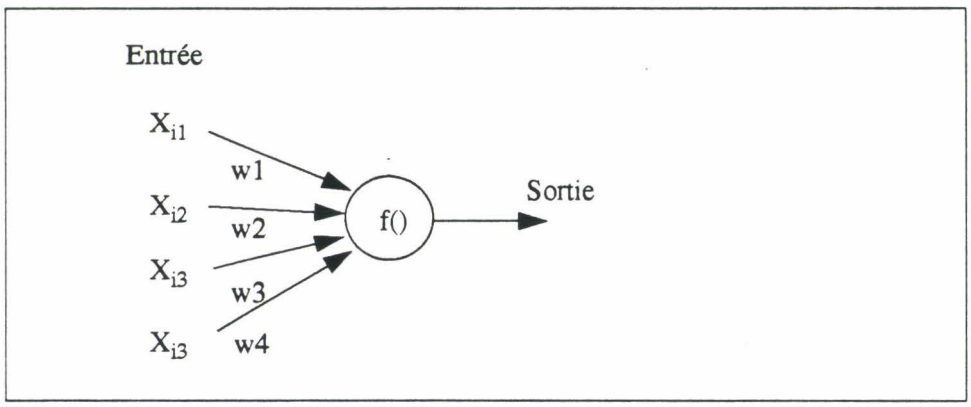


FIGURE 87 Neurone à seuil à 4 entrées

Pour cela nous avons défini 3 vecteurs de valeurs réelles qui serviront d'exemples d'apprentissage. Ces 3 exemples seront fournis successivement en entrée du réseau de façon répétitive pendant tout l'apprentissage. L'ensemble des 3 sorties désirées correspondant aux 3 vecteurs exemples est également construit sous la forme d'un objet de taille fixe, c'est à dire un Vecteur (d).

On peut alors construire au fur-et-à-mesure de l'apprentissage une séquence de vecteurs W qui sont les poids que l'on ajuste à chaque passage de l'un des exemples, jusqu'à ce que le réseau converge (s'il converge) c'est à dire qu'il donne pour chaque exemple X_i la sortie désirée d_i.

La figure suivante résume les exemples utilisés dans la simulation d'apprentissage ainsi que les règles d'apprentissage que l'on écrit directement sous la forme des opérateurs de manipulation que nous avons définis pour les calculs sur les objets de type Vecteurs, Matrices ou Listes.

$$X_1 = \begin{bmatrix} 1 \\ -2 \\ 0 \\ -1 \end{bmatrix} \quad X_2 = \begin{bmatrix} 0 \\ i5 \\ -05 \\ -1 \end{bmatrix} \quad X_3 = \begin{bmatrix} -1 \\ 1 \\ 05 \\ -1 \end{bmatrix}$$

Sortie désirée:

$d = \begin{bmatrix} -1 & -1 & 1 \end{bmatrix}$

Vecteur Poids initial:

$W = \begin{bmatrix} 1 & -1 & 0 & 05 \end{bmatrix}$

La règle d'apprentissage du Perceptron:

W[i+1] = W[i] + (d[i]-(W[i]*X[i]).sgn())*X[i];

La règle de Rétro-propagation:

W[i+] = W[i] + (d[i]-f)*(1-f*f)*V[i]

où f=(W[i]*V[i]).sgm()

FIGURE 88 Apprentissage d'un réseau de neurones

Conclusion

L'utilisation des séquences gérant les réseaux de neurones que nous avons expérimentés porte principalement sur des éléments de taille fixe tels que les Vecteurs ou Matrices. Il serait intéressant de décrire à l'aide des quelques primitives que nous avons construites des apprentissages de réseaux plus dynamiques, ceux qui voient leur nombre de couches et de connexions augmenter au fur-et-à-mesure de l'apprentissage. Ce dernier type de réseau, à topologie variable, permettrait davantage de tester la souplesse apportée par la gestion rudimentaire des Listes.

La manipulation des données multimédia que nous avons effectuée au travers de la définition de quelques primitives a été réalisée en langage C++ sur un environnement PC donc extérieur à la carte, ceci à des fins de simulation.

L'intégration directe de ce jeu de primitives dans la carte supposerait d'avoir un compilateur de langage C++ adéquate. S'il existe déjà un compilateur de langage C dédié à la carte à micro-processeur, C-CARD *[GRIM91]*, une version C++ n'est pour le moment pas disponible. En outre, le but ici étant davantage la définition de primitives utiles à la manipulation des données plutôt que l'intégration de ce noyau par un langage particulier, nous n'avons pas recherché le langage le plus compact en terme de nombres d'octets. Pour un transfert industriel on pourrait réécrire les primitives dans d'autres langages alliant souplesse et optimisation.

Conclusion

Ce doctorat s'est particulièrement intéressé à des approches nouvelles de la sécurité au sein de l'environnement cartes à puce, utilisant notamment des techniques du multimédia au travers de la biométrie ou de la compression et plus généralement des techniques de l'intelligence artificielle.

Dans un cadre plus large, on peut envisager de façon similaire la manipulation de données multimédia pour des traitements plus généraux de reconnaissance de formes , d'analyse de courbes, schémas ou plans dans des domaines aussi variés que la santé, les loisirs ou la culture.

Bien que les nouvelles techniques que nous avons expérimentées en matière de sécurité (notamment le «Radar» et la reconnaissance de la signature comportementale du clavier) soient séduisantes par les nombreux avantages qu'elles apportent, des évaluations plus poussées seraient nécessaires pour valider leur ouverture à des applications destinées à un large public. En effet, nous n'avons pu tester ces systèmes que sur un ensemble réduit de personnes, de larges bases de données regroupant les informations nécessaires à l'évaluation n'étant pas disponibles (contrairement à des systèmes plus classiques de reconnaissance de la voix ou des empreintes digitales qui possèdent des banques de données énormes).

Dans le cas de l'identification comportementale, une ouverture intéressante concernant la détection de fraude serait d'appliquer un système de type «Radar» aux nouveaux systèmes de commerce électronique basés autour des «autoroutes de l'information» (Internet). Nous avons l'intention d'étudier cette aspect dans des recherches futures, et principalement autour du projet Netbill *[NETB95]* en cours de développement et qui a pour but la construction d'un serveur de paiement de transactions sur Internet.

Table des Figures

Références Bibliographiques

[ABAD90] Martin Abadi, "Authentication and Delegation with Smart Cards", DECS 67 Technical Report, Digital Equipment Corporation Systems Research Center, Palo Alto, CA, USA, 1990.

[ACMM93] *Proceedings of the ACM International Conference on Multimedia*, Anaheim, CA, USA, August 1-6, 1993.

[ALEX93] Thomas Alexandre, «La Compression de données dans la Carte à Microprocesseur», Rapport LIFL AS-142 (CNRS URA 369), Université des Sciences et Technologies de Lille, Décembre 1993.

[ALEX94a] Thomas Alexandre, Vincent Cordonnier, «The Radar Concept», in *Proceedings of the CardTech/SecurTech'94 International Conference*, Arlington, Virginia, USA, April 10-13, 1994, pp 87-98.

[ALEX94b] Thomas Alexandre, Vincent Cordonnier, «An Object-Oriented Approach for implementing Biometrics in Smart Cards», in *Proceedings of the CardTech/SecurTech'94 International Conference*, Arlington, Virginia, USA, April 10-13, 1994, pp 149-160.

[ALEX94c] Thomas Alexandre, «The Radar Concept using Neural Networks», in *Proceedings of the First Smart Card Research and Advanced Application Conference CARDIS'94*, Lille, France, October 24-26, 1994.

[ALEX95] Thomas Alexandre, Patrick Trane, «Detecting Intrusions in Smart Card Applications using Expert Systems and Neural Networks», to appear *in Proceedings of the 11th International Information Security Conference*, Cape Town, South Africa, May 9-12, 1995.

[AMMA86] M. Ammar, Y. Yoshida, and T. Fukumura, "A new effective approach for off-line verification of signatures by using pressure features", in *Proceedings of the Eighth International Conference on Pattern Recognition* (Paris, France, October 27-31, 1986), IEEE Publ. 86CH2342-4, 566-569.

[AMMA88] M. Ammar, Y. Yoshida, and T. Fukumura, "Description of signature images and its application to their classification", in *Proceedings of the Ninth International Conference on Pattern Recognition* (Rome, Italy, November 14-17, 1988), Computer Society Press, Washington, DC, 1988, pp 23-26.

[ANDE93] D. Anderson, T. F. Lunt, H. S. Javitz, A. Tamaru, A. Valdes, «Detecting Unusual Program Behavior Using the NIDES Statistical Component», Safeguard Final Report, SRI International, Menlo Park, California, December 1993.

[AROZ90] M. Arozullah, A. Namphol, "A data compression system using neural network based architecture", in *Proceedings of the International Joint Conference on Neural Networks*, San Diego, *CA*, USA, June 17-21, 1990.

[ARPS93] Ronald B. Arps, Thomas K. Truong, "Comparison of international standards for lossless still image compression", RJ 9639 Technical Report, I.B.M. Research Division, Yorktown Heights, N.Y., 1993.

[BARN88a] Michael F. Barnsley, *Fractals everywhere*, Boston: Academic Press, 1988.

[BARN88b] Michael F. Barnsley, Alan D. Sloan, "Fractal Image Compression", in *Proceedings of the Scientific Data Compression Workshop*, NASA Conference Publication 3025, Snowbird, Utah, May 3-5, 1988.

[BARN93] Michael F. Barnsley, Lyman P. Hurd, *Fractal Image Compression*, AK Peters, 1993.

[BATE92] Joseph Bates, «Virtual Reality, Art and Entertainment», Presence: The Journal of Teleoperators and Virtual Environments», *The MIT Press*, Vol.1, No.1, pp 133-138.

[BAUE92] Hans-Ulrich Bauer, Klaus Pawelzik, Theo Geisel, "A Topographic Product for the Optimization of Self-Organizing Feature Maps", in *Lippmann, Moody, Touretzky (eds), Advances in Neural Information Processing Systems 4, Morgan Kaufmann*, San Mateo, CA, 1992.

[BENG90] Yoshua Bengio, "Global optimization of a Neural Network: Hidden Markov Model hybrid", SOCS 90-22 Technical Report, Mc Gill University, Montreal, Quebec, Canada, 1990.

[BENN92] Younès Bennani, «La Reconnaissance Automatique du Locuteur: Formulation et Etat de l'art», Laboratoire de Recherches en Informatique, Centre d'Orsay, Université de Paris Sud, février 1992.

[BENT86] Jon Louis Bentley, Daniel D. Sleator, Robert E. Tarjan, Victor K.Wei, «A locally Adaptive Data Compression Scheme», *Communications of the ACM*, Vol.29, No.4, April 1986.

[BEST94] Peter Best, «MIATA: Machine Independent Audit Trail Analysis», Information Security Research Centre, School of Data Communications, Queensland University of Technology, Australia, July 1994.

[BLOCK] H.D.Block, «The Perceptron: a model for brain functionning», *Reviews of Modern Physics 34*, pp 123-135.

[BOUA92] Hazem Bouattour, Françoise Fogelman Soulié, Emmanuel Viennet, «Neural Nets for Human Face Recognition», Laboratoire de Recherches en Informatique, Centre d'Orsay, Université de Paris Sud, février 1992.

[BOWE93] Tamme D. Bowen , Kelly M. Hall, "Towards a Better Understanding of Dylan", Research Report LAL-93-01, Laboratory for Applied Logic, University of Idaho, May 1993.

Sur Internet:ftp://ftp.cambridge.apple.com:/pub/dylan/.

[BRAT90] Ivan Bratko, *Prolog Programming for Artificial Intelligence*, second edition, Addison-Wesley, 1990.

[BRIG88] R. Bright, *Smart Cards, Principles, Practice, Applications*, Ellis Horwood Limited, 1988.

[BURE91] Gilles Burel, «Une nouvelle approche pour les réseaux de neurones: la représentation scalaire distribuée», Recherches, Traitement du Signal, Vol.10, No.1, novembre 1991, pp 41-51.

[CARD93] «Neural Networks: The way forward ?», *Cards International*, Issue 99, December 9, 1993.

[CARO93] Olivier CARON, Vincent CORDONNIER, Georges GRIMONPREZ, "OCEAN: A Hardware and software tool for design of future smart cards", *in Proceedings of EUROMICRO 93.*

[CARO94] Olivier CARON, «Méthodologies de conception et d'évaluation d'architectures R.I.S.C. adaptées aux futures cartes à micro-processeur», Thèse de Doctorat, LIFL, URA CNRS 369, USTL, Lille, 1994.

[CART94] Bob Carter, «The Present and Future State of Biometric Technology», in *Proceedings of the CardTech/SecurTech'94 International Conference*, Arlington,Virginia, USA, April 10-13, 1994, pp 401-415.

[CASC94] CASCADE Consortium, «Biometrics», Deliverable 1 of the ESPRIT - CASCADE European Project EP8670: Technology Assessment, May 4, 1994.

[CHAS91] Jean Marc CHASSERY, Annick MONTANVERT, *Géométrie Discrète en analyse d'images*, Hermès, 1991.

[CRIN90] J.Mc Crindle, *Smart Cards*, IFS Publications, Springer-Verlag, 1990.

[CULI93] Karel Culik, Simant Dube, «Efficient Compression of Wavelet Coefficients for Smooth and Fractal-like Data», in *Proceedings of the 10th Annual Symposium on Theoretical Aspects of Computer Science (STACS'93)*, Springer-Verlag, February 1993.

[DANI88] P. E. Danielsson and Q. Z. Ye, "Rotation-invariant operators applied to enhancement of fingerprints", *Ninth International Conference on Pattern Recognition* (Rome, Italy, November 14-17, 1988), *Computer Society Press*, Washington, DC, 1988, pp 329-333.

[DAVA93] Eric Davalo, Patrick Naïm, *Des Réseaux de Neurones*, Eyrolles, Paris, 2nd edition, 1993, 232 pages.

[DEDE93] Christopher J. Dede, «The Future of Multimedia: bridging to Virtual Worlds», in *Multimedia for learning: development, application, evaluation*, Diane M. Gayeski ed., N.J.: Educational Technology Publications, Englewood, 1993.

[DEHA92] Gregory R. De Haan, Omer N. Egecioglu, "Links between self-organizing feature maps and weighted vector quantization", TRCS-92-1 Technical Report, University of California, Santa Barbara, 1992.

[DEKK93] Anthony H. Dekker, "Optimal colour quantization using Kohonen Neural Networks", TR- 10/93 Technical Report, Dept. of Information Systems and Computer Science, National University of Singapore, Kent Ridge, Singapore, October 1993.

[DELA87] Jean-Paul Delahaye, *Systèmes Experts: organisation et programmation des bases de connaissance en calcul propositionnel*, Editions Eyrolles, 1987.

[DEMU92] H. Demuth, M. Beale, Neural Network Toolbox for Use with Matlab, The MathWorks Inc., 1992.

[DENN87] D. E. Denning, «An intrusion-detection model», 1987.

[DESA91] Virginia R. De Sa, Dana H. Ballard, "Top-down teaching enables non-trivial clustering via competitive learning", TR-402 Technical Report, University of Rochester, N.Y., USA, 1991.

[DIGI92] Digital Systems Research Center, «On-line Data Compression in a Log-structured File System», Palo Alto, CA, USA, April 1992.

[DIGI93] Digital Signatures, Inc., 9050 Red Branch Road, Columbia, MD 21045-2174, 1993.

[DIMI94] Dimitri A. Dimitroyannis, "Virtual Classroom: A Case Study", in *Proceedings of the First International Conference on the World Wide Web*, CERN, Geneva, Switzerland, May 25-27, 1994.

Sur Internet (WWW): http://www1.cern.ch/WWW94/Welcome.html.

[DOMA94] Domain Dynamics Limited, «TESPAR: Time Encoded Signal Processing and Recognition», Cascade Project EP 8670, March 7, 1994.

[DUVA89] S. Duval, et al., "Use of fingerprint as identity verification", in *Proceedings of the Fifth IFIP International Conference on Computer Security*, Gold Coast, Queensland, Australia, *ed. by William J. Caelli*, May 19-21, 1988.

[DYLA94] "Dylan: An Object Oriented Dynamic Language", Interim Reference Manual, Apple Computer Eastern Research and Technology, Apple Computer Inc., June 3, 1994.

Sur Internet:ftp://ftp.cambridge.apple.com:/pub/dylan/.

[DZIE89] J. M. Dzierzanowski, K. R. Chrisman, G. J. Mac Kinnon, P. Klahr, "The Authorizer's Assistant: a knowledge-based credit authorization system for American Express", in *Proceedings of the First Annual Conference on Innovative Applications of Artificial Intelligence*, Stanford, CA, USA, March 28-30, 1989.

[ESSA94] Irfan Essa, Alex Pentland, «A Vision System for Observing and Extracting Facial Action Parameters», in *Proceedings of IEEE Computer Vision and Pattern Recognition Conference*, Seattle, WA, pp 76-83, June 1994.

[EYED93] EyeDentify, Inc, 11931 Industriplex Blvd, Suite 300, Baton Rouge, LA 70809, USA, 1993.

[FAHL88] Scott E. Fahlman, «Faster-Learning Variations on Back-Propagation: An Empirical Study», in *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann, 1988.

[FAHL90] Scott E. Fahlman, Christian Lebiere, "The Cascade-Correlation Learning Architecture", CMU-CS-90-100 Technical Report, Carnegie Mellon University, Pittsburgh, USA, 1990.

Sur Internet (WWW): http://gs71.sp.cs.cmu.edu:1392:/

[FAHL91] Scott E. Fahlman, "The Recurrent Cascade-Correlation Architecture", in *Lippmann, Moody, Touretzky (eds), Advances in Neural Information Processing Systems 3, Morgan Kaufmann*, San Mateo, CA, 1991.

Sur Internet (WWW): http://gs71.sp.cs.cmu.edu:1392:/

[FLOC92] P. Flocchini, F. Gardin, G. Mauri, M. P. Pensini, and P. Stofella, "Combining image processing operators and neural networks in a face recognition system", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 6, 1992, pp 447-467.

[FREE61] H. Freeman, «On the encoding of arbitrary geometric configuations», *Computer Methods in Image Analysis, Aggrawal & al. ed., IEEE Press*, 1977. Reprinted from *IRE Trans. Electron. Compu.*, Vol.10, pp260-268, June 1961.

[FUHR94] Borko Fuhrt, "Multimedia Systems: an Overview", *IEEE on Multimedia*, pp 47-58, Spring 1994.

[FUNA89] K. Funahashi, "On the Approximate Realization of Continuous Mapping By Neural Networks", *Neural Networks*, Vol.2, No.3, 1989.

[GAYE93] Diane M. Gayeski, *Multimedia for learning: development, application, evaluation*, Educational Technology Publications, Englewood, N.J., USA, 1993.

[GELE91] E. Gelenbe, *Neural Networks: Advances and Applications*, Elsevier Science Publishers, 1991.

[GORD92] E. Gordons, G. Grimonprez, «A card as element of a distributed database», IFIP WG8.4 Workshop, Ottawa, 1992.

[GRAC91] A. E. Grace and M. Spann, "A comparison between Fourier-Mellin descriptors and moment based features for invariant object recognition using neural networks", *Pattern Recognition Letters*, Vol. 12, 1991, pp 635-643.

[GRAN72] G. H. Grandlund, «Fourier preprocessing for handprint character recognition», *IEEE transactions on Computers*, Vol.21, No.2, pp 195-201, 1972.

[GRIM91] G. Grimonprez, P. Paradinas, «A new approach in code development: C-CARD and COSSACK», in *Proceedings of the CardTech'91 International Conference*, Washington DC, 1991.

[GRIM92] G. Grimonprez, «Etude et réalisation d'une carte à microprocesseur intégrée aux systèmes de gestion de bases de données», Mémoire d'habilitation, Février 1992.

[GROS91] Stephen Grossberg, Gail A. Carpenter, *Pattern Recognition by self-organizing Neural Networks*, The MIT Press, Cambridge, MA, USA, 1991.

[GUIL92] L.Guillou, J.J.Quisquater, M.Ugon, *The smartcard: A Standardised Security Device Dedicated to Public Cryptology*, Ed. G.Simmons: Contempory Cryptology, *IEEE-Press*, 1992.

[GUOJ92] Lu Guojun, "Advances in digital image compression techniques", TRA2/92 Technical Report, Dept. of Information Systems and Computer Science, National University of Singapore, Kent Ridge, Singapore, February 1992.

[HAYK88] Martha E. Haykin, Robert B. J. Warnar, *Smart Card Technology: new methods for computer access control*, Security Technology Group, Institute for Computer Science and Technology, NIST, Gaithersburg, MD, USA, 1988.

[HELD91] Gérard HELD, *Data Compression, Techniques and Applications, Hardware and Software considerations*, Wiley, 1991.

[HILL88] David R. Hill, *Experiments in computational matrix algebra*, Random House, New York, 1988.

[HOEH91] Markus Hoehfeld, Scott E. Fahlman, "Learning with limited numerical precision using the cascade-correlation algorithm", CS-91-130 Technical Report , School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1991.

Sur Internet (WWW): http://gs71.sp.cs.cmu.edu:1392:/

[HOFF94] Helene Hoffman, «Network-compatible Multimedia for Medical Training: MedPics», Project Report, *IEEE Multimedia*, Fall 1994, pp 71-73.

[HOLD89] Ronald M. Holdaway, "Enhancing Supervised Learning Algorithms Via Self-Organization", in *Proceedings of the International Joint Conference on Neural Networks*, Washington D.C., June 18-22, 1989, Vol.II, pp 523-530.

[HOLM91] James P. Holmes, Larry J. Wright, Russell L. Maxwell, «A Perfomance Evaluation of Biometric Identification Devices», SAND91-0276 Technical Report, Systems Engineering Division, Sandia National Laboratories, Albuquerque, NM 87185.

[HOWA91] Paul G. Howard, Jeffrey S. Vitter, "Analysis of arithmetic coding for data compression", CS-91-03 Technical Report, Brown University, Providence, R.I., January 1991.

[HTML] HyperText Markup format Language. Sur Internet (World Wide Web): http://info.cern.ch/hypertext/WWW/MarkUp/HTMLPlus/htmlplus_1.html.

[HUNK94] Martin Hunke, «Locating and Tracking of Human Faces with Neural Networks», CMU-CS-94-155 Technical Report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1994.

Sur Internet (WWW): http://gs71.sp.cs.cmu.edu:1392:/

[IDEN93] Identicator Technology, 851 Traeger Avenue, Suite 310, San Bruno, CA 94066, USA, 1993.

[ISO94] ISO/IEC 7816-4, «Inter-industry commands for interchange», 1994.

[JAGA93] R. Jagannathan, A. Tamaru, F. Gildham, D. Anderson, C. Jalali, C.Dodd, H. S. Javitz, A. Valdes, T. F. Lunt and P. G. Neumann, *«Next Generation Intrusion-Detection Expert System (NIDES): Software Design Specifications»*, Technical Report, Computer Science Laboratory, SRI International, Menlo Park, California, March 1993.

[JENK77] Ian C. Jenkins, "The computerized acquisition, compression and retrieval of electrocardiograms", Ph.D. Thesis, Carnegie Mellon University Dept. of Biomedical Engineering, Pittsburgh, PA, 1977.

[KAME93] M. S. Kamel, H. C. Shen, A. K. C. Wong, R. I. Campeanu, «System for the recognition of human faces», *IBM Systems Journal*, Vol.32, No.2, 1993.

[KANG90] J. A. Kangas, T. K. Kohonen, J. T. Laaksonen, "Variants of Self-Organizing Maps", *IEEE Transactions on Neural Networks 1*, p 93, 1990.

[KAYA91] Masahiro Kayama, "Methodology for constructing optimal multi-layered Neural Networks", Technical Report CSD-910068, University of California, Los Angeles, CA, USA, 1991.

[KEYE92] J. Keyes, "Winning back investors' confidence", *Information Strategy*, Vol.8, No.2, Winter 1992.

[KIMU89] Takayuki Dan Kimura, "Back-Propagation with integer arithmetic", Technical Report WUCS-89-25, Washington University Department of Computer Science, Saint Louis, Mo, 1989.

[KJEL91] L. Kjelldahl, *MULTIMEDIA: systems, interaction and applications*, L. Kjelldahl editions, First Eurographics Workshop, Stockholm, Sweden, April 18-19, 1991.

[KLEM94] Anders Klemets, "The Design and Implementation of a Media on Demand System for WWW", in *Proceedings of the First International Conference on the World Wide Web*, CERN, Geneva, Switzerland, May

25-27, 1994.

Sur Internet (WWW): http://www1.cern.ch/WWW94/Welcome.html.

[KOEG94] John F. Koegel Buford, *Multimedia Systems*, Addison-Wesley, 1994.

[KOHO87] Teuvo Kohonen, *Self-Organization and Associative Memory*, 2nd edition, Berlin: Springer-Verlag, 1987.

[KOHO91] T. Kohonen, K. Mäkisara, O. Simula, J. Kangas, «Artificial Neural Networks», in *Proceedings of the 1991 International Conference on Artificial Neural Networks*, Espoo, Finland, 24-28 June, 1991.

[KOLE91] John F. Kolen, Jordan B. Pollack, "Back-Propagation is Sensitive to Initial Conditions", in *Lippmann, Moody, Touretzky (eds), Advances in Neural Information Processing Systems 3, Morgan Kaufmann*, San Mateo, CA, 1991.

[LAMA84] F. Lamarche and R. Plamondon, "Segmentation and feature extraction of handwritten signature patterns", *in Proceedings of the Seventh International Conference on Pattern Recognition* (Montreal, Canada, July 30-August 2, 1984), IEEE Publ. 84 CH2046-1, 756-759.

[LANG88] Glen G. Langdon, "Further developments in lossless gray-scale image compression", RJ-6426 Research Report, I.B.M. Research Division, Yorktown Heights, N.Y., 1988.

[LAVE94] M. G. Lavenant, J. A. Kruper, "The Phoenix Project: Distributed Hypermedia Authoring", in *Proceedings of the First International Conference on the World Wide Web*, CERN, Geneva, Switzerland, May 25-27, 1994.

Sur Internet (WWW): http://www1.cern.ch/WWW94/Welcome.html.

[LEE94] Woobin Lee, Yongmin Kim, Robert J. Gove and Christopher J. Read, "MediaStation 5000: Integrating Video and Audio", *IEEE on Multimedia*, Summer 1994.

[LEGA91] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications", *Communications of the ACM*, Vol.34, No.4, pp 45-68, April 1991.

[LIPP87] R. P. Lippmann, "An introduction to computing with Neural Networks", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol.2, No.4, pp 4-22, April 1987.

[LIPP91] S. B. Lippman, *A C++ Primer*, Addison-Wesley, 1991.

[LITT93] E. Littmann, H. Ritter, "Generalization Abilities of Cascade Network Architectures", in *Lippmann, Moody, Touretzky (eds), Advances in Neural Information Processing Systems 5, Morgan Kaufmann*, San Mateo,

CA, 1993.

[LORE84] G. Lorette, "On-line handwritten signature recognition based on data analysis and clustering", *in Proceedings of the Seventh International Conference on Pattern Recognition* (Montreal, Canada, July 30- August 2, 1984), IEEE Publ. 84CH2046-1, 1284-1287.

[LUCA90] S. M. Lucas, R. I. Damper, "Signature verification with a syntactic neural net", in *Proceedings of the International Joint Conference on Neural Networks*, San Diego, CA, June 17-21, 1990, Vol.I, pp 373-378.

[LUNT93] Teresa F. Lunt, «Detecting Intruders in Computer Systems», in *Proceedings of the 1993 Conference on Auditing and Computer Technology, 1993.*

[MANJ92] B. S. Manjunath, R. Chellappa, and C. von der Malsburg, "A feature based approach to face recognition", Proceedings CVPR '92 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition), Champaign, IL, June 15-18, 1992 (IEEE Computer Society Press), pp 373-378.

[MARS92] Xavier MARSAULT, *Compression et cryptage en informatique*, Hermès, 1992.

[MART91] T. Martinez, K.Schulten, "A 'Neural-Gas' Network Learns Topologies", in *Proceedings of the ICANN 91Conference, ed. Kohonen et al.*, Helsinki, I-397, 1991.

[MASC94] Michael Mascha, Gary Seaman, "Interactive Education: Transitioning CD ROMs to the Web", in *Proceedings of the First International Conference on the World Wide Web*, CERN, Geneva, Switzerland, May 25-27, 1994.

Sur Internet (WWW): http://www1.cern.ch/WWW94/Welcome.html.

[MAST93] Timothy Masters, *Practical neural network recipes in C++*, Boston: Academic Press, 1993.

[MINS69] M. Minsky, S. Papert, *«Perceptrons»*, The MIT Press, 1969.

[MJOL85] Eric Mjolness, "Neural Networks, Pattern Recognition, and Fingerprint Hallucination", Technical Report 5198:85, Dept. of Computer Science, California Institute of Technology, 1985.

[NAIK90] Jayant M. Naik, «Speaker Verification: A Tutorial», *IEEE Communications Magazine*, January 1990.

[NASR88] Nasser M. Nasrabadi, Yushu Feng, «Vector Quantization of Images Based Upon the Kohonen Self-Organizing Feature Maps», in *Procee-*

dings of the IEEE International Conference on Neural Networks, Vol.1, San Diego, July 24-27, 1988, pp 101-108.

[NAZA92] Jamshid Nazari, Okan K. Ersoy, "Implementation of back-propagation Neural Networks with Matlab", EE 92-39 Technical Report, School of Electrical Engineering, Purdue University, West Lafayette, Ind., 1992.

[NELS92] M. NELSON, *The Data Compression Book, Featuring Fast, Efficient Data Compression Techniques in C*, M&T Books, 1991.

[NETB95] Sur Internet (WWW): http://www.cs.cmu.edu:8001/afs/andrew.cmu.edu/inst/ini/www/INI-home.html.

[NEUR92] Neuro-Nimes 92, Neural Networks and their Applications, Nimes, France, November 2-6, 1992.

[NEUR93] NeuroMetric Vision Systems, Inc., 500 Fairway Drive - Suite 205, Deerfield Beach, Florida 33441, USA, 1993.

[NGUY90] D. Nguyen, B. Widrow, «Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights», *International Joint Conference on Neural Networks*, Vol.3, pp 21-26, July 1990.

[NIGR93] Albert Nigrin, *Neural Networks for Pattern Recognition*, The MIT Press, Cambridge, MA, USA, 1993.

[PAIK92] W. H. Paik, E. Krause, J. Heller, "DigiCipher video compression technology for all digital, channel compatible, HDTV broadcast system", in *Proceedings of the Fourth International Workshop of Signal Processing of HDTV*, Kawasaki, Japan, November 18-20, 1992.

[PAIR77] C. Pair, M. C. Gaudel, *Les structures de données et leur représentation en mémoire*, Institut de Recherche d'Informatique et d'Automatique, Rocquencourt, Juin 1977.

[PAPI94] M. J. Papillon, J. Bérubé, M. Comeau, G. Lavoie, S. Kirouac, J. P. Fortin, «Futuristic Microchip Health Card Experiment in Québec», Canadian Medical Informatics, Vol. 1, No. 1, pp 16-17, May/June 1994.

[PARA94] P. Paradinas, J. J. Vandewalle, «A Personal & Portable Data Server: The CQL Card», *Lecture Notes in Computer Science 819*, Springer-Verlag, Ed. W.Litwin & T.Risch, 1994.

[PATI90] Yagyensh C. Pati, "Analysis and Synthesis of feedforward Neural Networks using affine wavelet transformations", TR 90-44 Technical Report, University of Maryland at College Park. Systems Research Center, 1990.

[PAUG89] Hélène Paugam-Moisy, "Réseaux connexionnistes", Rapport 89-16 du Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supé-

rieure de Lyon, France, 1989.

[PENT94] A. Pentland, B. Moghaddam, T. Starner, O. Oliyide, M. Turk, «View-Based and Modular Eigenspaces for Face Recognition», M.I.T. Media Laboratory Perceptual Computing Section Technical Report No.245, *IEEE Conference on Computer Vision & Pattern Recognition*, 1994.

Internet (WWW): http://tns-www.lcs.mit.edu/tns-www-home.html

[PERR90] J. L. Perry, J. M. Carney, «Human Face Recognition Using a Multilayer Perceptron», *I.J.C.N.N. 1990*, Washington D.C., Vol.2, pp 413-416, 1990.

[PERS77] E. Persoon, K. S. Fu, «Shape discrimination using Fourier descriptors», *IEEE transactions on Systems, Man and Cybernetics*, Vol.7, No.3, pp 170-179, 1977.

[PEYR94] P. Peyret, «RISC-based, Next-generation Smart Card Microcontroller Chips», in *Proceedings of the CardTech/SecurTech'94 International Conference*, Arlington,Virginia, USA, April 10-13, 1994.

[PIKE87] L. Piketty, J. Bakin, F. K. Dashiell, J. M. Dzierzanowski, "The Authorizer's Assistant: a large commercial expert system application", in *Proceedings of the Third Annual Artificial Intelligence and Advanced Computer Technology Conference*, Long Beach, CA, USA, April 22-24, 1987.

[PLAM88] R. Plamondon and M. Parizeau, "Signature verification from position, velocity and acceleration signals: a comparative study", in *Proceedings of the Ninth International Conference on Pattern Recognition* (Rome, Italy, November 14-17, 1988), Computer Society Press, Washington, DC, 1988, 260-265.

[PLUM93] P. PLUME, *Compression de données*, Eyrolles, 1993.

[POWE94] Robert S. Powers, «Smartcard applications of GSM authentication and privacy protocols», in *Proceedings of the CardTech/SecurTech'94 International Conference*, Arlington,Virginia, USA, April 10-13, 1994.

[PRIC86] W. L. Price, "A review of methods of personal identity verification", DITC 73/86 Technical Report, Division of Information Technology an Computing, National Physical Laboratory, Teddington, GB, 1986.

[RABI78] L. R. Rabiner, A. E. Rosenberg, S. E. Levinson, «Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition», *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. Assp-26, No.6, December 1978.

[RABI86] L. R. Rabiner, B. H. Juang, «An Introduction to Hidden Markov Models», *IEEE ASSP Magazine*, pp4-16, January 1986.

[RECO93] Recognition Systems, Inc., 62 S. San Tomas Aquino Road, Campbell, CA 95008, USA, 1993.

[REVE93] Alain Reverchon, Marc Ducamp, *Outils mathématiques en C++*, Armand Colin, 1993.

[RITT88a] H. Ritter, K. Schulten, "Convergence Properties of Kohonen's Topology Conserving Maps: Fluctuations, Stability and Dimension Selection", *Biology Cybernetics* 60, pp 59-71, 1988.

[RITT88b] H. Ritter, K. Schulten, "Kohonen Self-Organizing Maps: Exploring their Computational Capabilities", *IEEE International Conference on Neural Networks*, Vol.1, pp 109-116, 1988.

[ROMA93] Steve G. Romaniuk, K. R. Namuduri, N. Ranganathan, "A feature-based heuristic approach for lossless image compression, TRC1/93 Technical Report, Dept. of Information Systems and Computer Science, National University of Singapore, Kent Ridge, Singapore, January 1993.

[ROSE57] F.Rosenblatt, «The Perceptron: a Perceiving and Recognizing Automaton», Project PARA, Cornell Aeronautical Lab. Report 85-460-1, 1957.

[ROSE61] F. Rosenblatt, *Principles of Neurodynamics*, Washington D.C.: *Spartan Press*, 1961.

[ROSE87] Charles R. Rosenberg, "Revealing the structure of Nettalk's internal representations", CSL-9 Technical Report, Cognitive Science Laboratory, Princeton University, N.J., USA, 1987.

[ROTH90] J. A. Rothi, D. C. Yen, "Why American Express gambled on an expert data base", *Information Strategy*, Vol.6, No.3, pp 16-22, Spring 1990.

[ROUS90] Philippe Roussel, «Un Interprête Prolog dans une Carte à Puce», RD2P, LIFL, Rapport interne, Juillet 1990.

[RSA78] R. L. Rivest, A. Shamir, A. Adleman, «A method for obtaining digital signature and public key crypto-system», *Communications of the ACM*, February 1978.

[RUME86] D. E. Rumelhart, G. E. Hinton, R. J. Williams, «Learning internal representation by error propagation», D.Rumelhart and J.McClelland editors, *Parallel Distributed Processing: exploring the Microstructure of Cognition*, Vol.1, M.I.T. Press, Cambridge, MA, 1986.

[SAND90] C. P. Sandbank, *Digital Television*, Chichester, West Sussex, England; New York: Wiley, 1990.

[SANG89] Terence D. Sanger, "Optimal unsupervised learning in feedforward Neural Networks", Technical Report 1086, AI-Lab., Massachusetts Institute of Technology, MA, USA, 1989.

[SEME92] Alain Semesteys, *Le Multimedia*, Dunod, 1992.

[SHEP94] Colin Sheppard, «A Neural Network Approach to Fingerprint Verification», in *Proceedings of the CardTech/SecurTech'94 International Conference*, Arlington,Virginia, USA, April 10-13, 1994, pp 183-190.

[SHER92] Robin L. Sherman, «Biometric Futures», *Computer & Security*, Elsevier Science Publishers, Vol.2, No.2, 1992, pp 128-133.

[SONE89] N. Sonehara, M. Kawato, S. Miyake, K. Nakane, "Image Data Compression Using a Neural Network Model", in *Proceedings of the International Joint Conference on Neural Networks*, Washington D.C., June 18-22, 1989, Vol.II, pp 35-41.

[SPEN91] R. Spencer-Smith, *Logic and Prolog*, Al Group, Middlesex Polytechnic, A Harvester Wheatsheaf Publication, 1991.

[STAB86] Edward Stabler, "Object-Oriented Programming in Prolog", *AI Expert*, pp 46-57, October 1986.

[STAR93] Startek Eng. Inc., 3F, No.54, Park Ave II, Science-Based Industrial Park, Hsinchu 300, Taïwan, R.O.C., 1993.

[STER86] Leon Sterling and Ehud Shapiro, *The Art of Prolog: Advanced Programming Techniques*, The MIT Press, 1986.

[STRA92] Gilbert Strang, "Wavelet Transforms versus Fourier Transforms", CICS-P-332 Technical Report, Center for Intelligent Control Systems, MIT, Cambridge, MA, 1992.

Internet (WWW): http://tns-www.lcs.mit.edu/tns-www-home.html

[STRO91] B.Stroustrup, *The C++ Programming Language*, Addison-Wesley, 1991.

[TERR92] Jacques TERRASSON, *Les Outils du Multimedia*, Armand Colin, 1992.

[TOLA89] Viral V. Tolat, Allen M. Peterson, "A Self-Organizing Neural Network for Classifying Sequences", in *Proceedings of the IJCNN'89*, Washington D.C., June 18-22, 1989, Vol.II, pp 561-568.

[VOGL88] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, D. L. Alkon, «Accelerating the convergence of the backpropagation method», *Biological Cybernetics*, Vol.59, pp 257-263, 1988.

[WALL91] Gregory K. Wallace, «The JPEG Still Picture Compression Standard», *Communications of the ACM*, Vol.34, No.4, April 1991.

[WATS91] Mark Watson, *Common LISP Modules: Artificial Intelligence in the era of Neural Networks and Chaos Theory*, Springer-Verlag, New York, 1991.

[WEI91] Z. Wei, et al.,"Approximation Property of Multi-Layer Neural Net and its Application in Non-linear Simulation", *IJCCNN-Seattle*, 1991.

[WHIT91] Darrell Whitley, N. Karunanithi, "Generalization in feedforward neural networks", CS-91-108 Technical Report, Colorado State University, 1991.

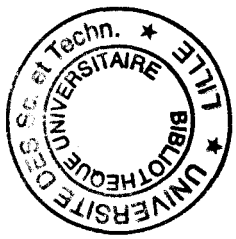
[YANG91] Jihoon Yang, Vasant Honavar, "Experiments with the cascade-correlation algorithm", TR 91-16a Technical Report, Iowa State University, Ames, Iowa, 1991.

[YU90] Yeong-Ho Yu, "Descending epsilon in back-propagation: a technique for better generalization", AI-90-130 Technical Report, University of Texas, Austin, USA, 1990.

[YUIL92] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates", *International Journal of Computer Vision*, Vol. 8, 1992, pp 99-111.

[ZAHN72] C. T. Zahn, R. Z. Roskies, «Fourier descriptors for plane closed curves», *IEEE transactions on Computers*, Vol.21, No.3, pp 269-281, 1972.

[ZIPL92] Zi plc, Access Control Systems, Highfields Science Park, Nottingham, UK, 1992.


UNIVERSITE DES Sc. et Techn.
BIBLIOTHEQUE UNIVERSITAIRE
LILLE