

# Mémoire de Recherche

## La reconnaissance de Forme

### Comment améliorer les techniques de reconnaissance de forme 3D ?

ESGI 5<sup>ème</sup> année

Spécialisation Animation Numérique Interactive

**Auteurs :**

- ✓ Damien Poisson
- ✓ Sami Mahjoub

**Maitre de Mémoire : Mr. Alain LIORET**

## Sommaire

Résumé .....	3
Introduction générale.....	4
La reconnaissance de Forme .....	4
Définition .....	5
Historique .....	5
Méthodes .....	6
Domaines d'application .....	7
Etat de l'art .....	8
Reconnaissance à l'aide d'arbres de classification .....	8
Description invariante d'objets 3D représentés par des images géométriques .....	11
Reconnaissance d'objets par vision focalisée .....	15
Indexation d'objets 3D : approche 2D-3D pour les requêtes à partir d'images.....	19
La méthode par HMM (Hidden Markov Model) .....	21
Approche structurelle .....	25
La méthode HTM.....	26
Autres méthodes .....	33
Base choisie à notre méthode .....	34
Introduction .....	34
Présentation de la technique du dodécaèdre .....	34
Le dodécaèdre dans notre cas.....	35
Conclusion .....	35
Amélioration de la méthode Indexation d'objets 3D.....	36
Introduction .....	36
Nos objectifs.....	36
Reconnaissance d'objet complexe.....	36
Optimisation du temps de reconnaissance .....	36
Ignorer les bruits .....	37
Utilisation d'un réseau de neurones .....	37
Description .....	37
Réseau de neurone utilisé .....	39
Les apports de notre réseau de neurone .....	40
Performances.....	41
Conclusion .....	45
Annexes.....	47
Bibliographie.....	49

## Résumé

Ce mémoire a pour sujet la reconnaissance de forme, il a pour objectif d'étudier les différentes méthodes de reconnaissance de forme et de fournir une amélioration aux techniques actuels. Dans celui-ci vous trouverez donc un détail des différentes techniques de reconnaissance de forme que nous avons étudié pour parvenir à notre solution. Lors de cette étude la technique que nous avons mise en avant est celle dite du « dodécaèdre » permettant de prendre plusieurs vue d'un objet afin de les reconnaître par la suite par un algorithme de reconnaissance de contours. Dans notre cas nous partons de ce principe du « dodécaèdre » afin de fournir une méthode de reconnaissance utilisant les couleurs pour permettre d'obtenir de meilleur résultat. Pour cela nous utilisons un réseau de neurone multi-couche qui apprend les différentes vues de l'objet fourni par la technique du « dodécaèdre » afin de pouvoir reconnaître n'importe quel objet par la suite peut importe les vues présenté.

## Introduction générale

La reconnaissance des formes a pour objet de simuler les activités de perception sensorielle du cerveau. En premier lieu, la perception visuelle et auditive, y compris dans des bandes spectrales non perçues par l'homme (infra rouge, radar, sonar, ...). La reconnaissance a besoin d'un modèle de l'objet. Pour l'être humain, ce modèle correspond à une représentation mentale de l'objet qui peut être apprise en retenant les caractéristiques les plus discriminantes de l'objet. Les caractéristiques peuvent être toutes sortes d'attributs de l'objet : forme, couleur, texture, taille, volume, etc.

Objet de recherches depuis l'apparition de l'informatique, cette discipline prend désormais un essor considérable dans l'industrie, car les progrès de l'électronique permettent désormais de doter les systèmes opérationnels de fortes puissances de calcul à des prix raisonnables. Elle regroupe un certain nombre de techniques allant de la simple identification d'objets isolés à l'analyse de scènes complexes ou la compréhension de la parole. Dans ces derniers cas, il faut tout à la fois segmenter en objets, identifier ces objets et établir un réseau de relations entre objets. La notion d'apprentissage est au cœur de la plupart des techniques développées. Ces techniques peuvent être utilisées soit seules dans le cas de systèmes dont la mission principale est d'identifier des objets, soit en relation étroite avec des techniques d'intelligence artificielle dans le cas de systèmes devant à la fois percevoir et raisonner sur les choses perçues.

En d'autres termes, le problème que cherche à résoudre la reconnaissance des formes est d'associer une étiquette à une donnée qui peut se présenter sous la forme d'une image brute ou d'un signal. Des données différentes peuvent recevoir la même étiquette ; ces données sont les réalisations ou les exemplaires de la classe identifiée par l'étiquette. Par exemple, l'écriture manuscrite du caractère A varie d'un scripteur à l'autre mais le lecteur identifiera le caractère A pour chacune de ces réalisations.

Des méthodes générales ont été développées en reconnaissance des formes pour extraire automatiquement des informations des données sensibles afin de caractériser les classes de formes (apprentissage) et d'assigner automatiquement des données à ces classes (reconnaissance). La mise en œuvre de ces méthodes générales pour des problèmes particuliers amène à introduire la notion de processus de reconnaissance qui pose la question de l'intégration des méthodes de la reconnaissance de formes dans un système qui a pour but de reconnaître des formes.

Parallèlement aux travaux sur les méthodes de reconnaissance, se développaient le traitement d'image et la vision par ordinateur. Ces domaines ont focalisé le problème de la reconnaissance sur des données spécifiques, mais par ailleurs ils ont permis de situer la reconnaissance dans un processus plus vaste d'interprétation d'image impliquant des niveaux de perception et des connaissances propres au domaine.

# La reconnaissance de Forme

## Définition

On désigne par **reconnaissance de formes** (ou parfois **reconnaissance de motifs**) un ensemble de techniques et méthodes visant à identifier des *motifs* à partir de données brutes afin de prendre une décision dépendant de la catégorie attribuée à ce motif. On considère que c'est une branche de l'intelligence artificielle qui fait largement appel aux techniques d'apprentissage automatique et aux statistiques.

Les *formes* ou *motifs* à reconnaître peuvent être de nature très variée. Il peut s'agir de contenu visuel (code barre, visage, empreinte digitale...) ou sonore (reconnaissance de parole), d'images médicales (rayon X, EEG, IRM...) ou multi spectrales (images satellitaires) et bien d'autres [wikip1].

## Historique

Or que ce soit pour déchiffrer un texte dactylographié ou manuscrit, pour compter des chromosomes, reconnaître une tumeur, un char ou un avion de guerre, la compréhension de l'image, sa classification passe toujours par la reconnaissance d'une forme. « *Plusieurs approches théoriques ont été développées* », explique *Olivier Faugeras*.

« Les premières consistaient à faire des calculs à partir de l'image et construire des représentations symboliques de plus en plus complexes, d'abord en deux dimensions tel que sur l'image, puis tridimensionnelles, pour tenter de restituer une description proche de notre propre vision. » Un peu partout dans le monde, les chercheurs ont mis au point des méthodes mathématiques permettant de détecter les contours des objets à partir des changements rapides de contraste dans l'image, des ombres et des lumières, des régions homogènes en couleur, en intensité, en texture.

« Dès 1964, des chercheurs français, Georges Matheron (1930-2000) et Jean Serra, ont développé une autre approche théorique (baptisée morphologie mathématique) et un outil spécifique (l'analyseur de texture breveté en 1965, ndlr) d'abord pour analyser des microphotographies de terrain et évaluer des teneurs en minerai, puis pour d'autres applications comme la cytologie (caractérisation et comptage de cellules) » rappelle *Olivier Faugeras*. En 1968, ils créent le Centre de morphologie mathématique de l'Ecole des Mines de Fontainebleau. Leurs outils d'analyse et d'interprétation d'images sont longtemps restés franco-français, jusqu'à ce qu'un américain, Robert Haralick (Université du Kansas à cette époque, de Seattle actuellement), en fasse une large publicité dans les années 1980, en les adaptant à de nombreuses applications : industrielles comme l'inspection radiographique des ailes d'avions de Boeing, aériennes ou médicales.

D'autres chercheurs, comme les américains Marvin Minsky et Seymour Papert du MIT (Massachusetts Institute of Technology) ont considéré le problème dans l'autre sens, en cherchant à formaliser et à faire reproduire par l'ordinateur notre propre processus de reconnaissance d'images, donc notre propre vision. Cette démarche était dans l'air du temps, au cœur des promesses de « l'intelligence artificielle » qui devait permettre de mettre l'intelligence en équations et doter les ordinateurs de toutes les capacités humaines de raisonnement, mémoire, perception. Or la vision s'est révélée être un domaine particulièrement complexe à modéliser tant elle est basée sur une quantité phénoménale de connaissances a priori fondées sur notre intelligence et notre expérience [sit1].

## Méthodes

La reconnaissance de motifs peut être effectuée au moyen de divers algorithmes d'apprentissage automatique tels:

- un réseau de neurones
- une analyse statistique
- l'utilisation de modèles de Markov cachés
- une recherche d'isomorphisme de graphes ou sous-graphes

Les formes recherchées peuvent être des formes géométriques, descriptibles par une formule mathématique, telles que :

- cercle ou ellipse
- courbes de Bézier, splines
- droite

Elles peuvent aussi être de nature plus complexe:

- lettre
- chiffre
- empreinte digitale

Les algorithmes de reconnaissance peuvent travailler sur des images en noir et blanc, avec en blanc les contours des objets se trouvant dans l'image. Ces images sont le fruit d'algorithmes de détection de contours. Ils peuvent aussi travailler sur des zones de l'image prédéfinies issues de la segmentation de l'image.

Méthodes de reconnaissance de formes :

- Mise en correspondance de graphes
- Méthode Bayésienne
- Estimation Paramétrique
- Classifieur linéaire
- Réseau de neurones
- Local feature focus
- SVM : Support Vector Machine
- Polytôpes de contrainte
- Méthode des hypercubes

Un algorithme bien connu pour la détection de formes, la transformée de Hough, est une méthode d'estimation paramétrique [wikip1].

## Domaines d'application

La reconnaissance de formes est utilisée dans plusieurs domaines d'activités. Parmi ces domaines on peut citer :

- Recherche d'images par le contenu : Cette technologie est actuellement intéressante pour la recherche de données sur l'imagerie médicale, ou cartographiques.
- Classification de documents : Son activité est essentielle dans de nombreux domaines économiques : elle permet d'organiser des corpus documentaires, de les trier, et d'aider à les exploiter dans des secteurs tels que l'administration, l'aéronautique, la recherche sur internet, les sciences.
- Reconnaissance de l'écriture manuscrite : Cette technologie fait appel à la reconnaissance de forme, mais également au traitement automatique du langage naturel. Cela veut dire que le système, tout comme le cerveau humain, reconnaît des mots et des phrases existant dans un langage connu plutôt qu'une succession de caractères. Ceci améliore grandement la robustesse.

## Etat de l'art

Plusieurs techniques ont été utilisées pour la reconnaissance des formes, dans ce chapitre on va essayer de faire le tour de ces principales techniques.

### Reconnaissance à l'aide d'arbres de classification

Dans une publication de *Bruno Jedynak* et *François Fleuret* datée de 1996, l'objectif était de développer un algorithme de reconnaissance d'objets 3D à partir d'images [publi1]. Pour cela, ils ont proposé une approche empirique basée sur un apprentissage à partir de bases de données d'images, constituées de nombreuses vues de chaque objet dans différentes positions, et cela sans utilisation de modèles pour la géométrie des objets ou la photométrie de la scène, et en supposant donnée une petite liste d'objets (de 10 à 100 objets).

L'idée de base est la suivante : tout d'abord associer à chaque pixel de l'image un code décrivant la topographie du voisinage (uniforme, bord, coin, texture, etc.) ; puis, poser de nombreuses questions sur la disposition spatiale de ces codes. Par exemple, nous pourrions demander « y a-t-il un pixel de code T1 au nord d'un autre de code T2 et au sud d'un troisième de code T3 ? ». La réponse est « oui » ou « non », selon que la configuration cherchée apparaît ou non dans l'image. Comme les configurations ne font intervenir que des contraintes relatives sur les angles, les questions associées jouissent de bonnes propriétés d'invariance.

Pour illustrer le fonctionnement et les résultats de cet algorithme ils ont utilisé les images de la « Columbia Object Image Library (COIL-20) ». Cette base de données contient 1440 images 128x128 codées sur 8 bits de 20 objets sur fond noir (72 images de chaque objet).

Les objets sont disposés tour à tour sur une table tournante. Une caméra fixe acquiert une image tous les 5 degrés d'angle. Les objets sont normalisés en dimensions de façon à remplir au mieux l'image, sans pour autant la déformer. Les objets sont présentés sur la figure 1. Le protocole d'utilisation de cette base de données, consiste à extraire une image sur deux (par exemple les images de numéro pair) pour l'apprentissage, et à effectuer le test, et donc à évaluer les résultats de l'algorithme sur l'autre moitié des images. Ces auteurs présentent d'excellents taux de reconnaissance avec ce protocole.



Fig. 1 – Columbia Object Image Library (COIL-20)

La première étape de cet algorithme consiste à recoder les images en « étiquette », c'est-à-dire à remplacer en chaque pixel la donnée de niveau de gris par un code entier qui décrit la topographie de l'imagette 5x5 centrée sur le pixel (figure 2). Des exemples d'images ainsi recodées sont présentés (figure 4).

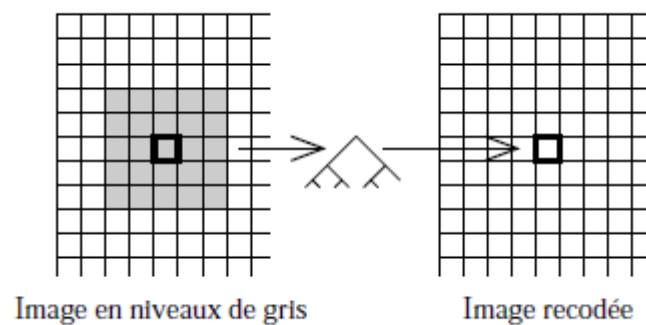


Fig. 2 – Le recodage consiste à construire une image dans laquelle chaque pixel porte un code décrivant son voisinage 5x5 dans l'image originale. Ce code est obtenu à l'aide d'un arbre de classification.

Comme l'illustre la figure 3 à chaque imagette 5x5 est associé un code entier compris entre 0 et 31. Chaque colonne de cette image contient des exemples d'imagettes ayant le même code (il y a donc 32 colonnes). Les imagettes d'un même groupe ont des propriétés géométriques communes.

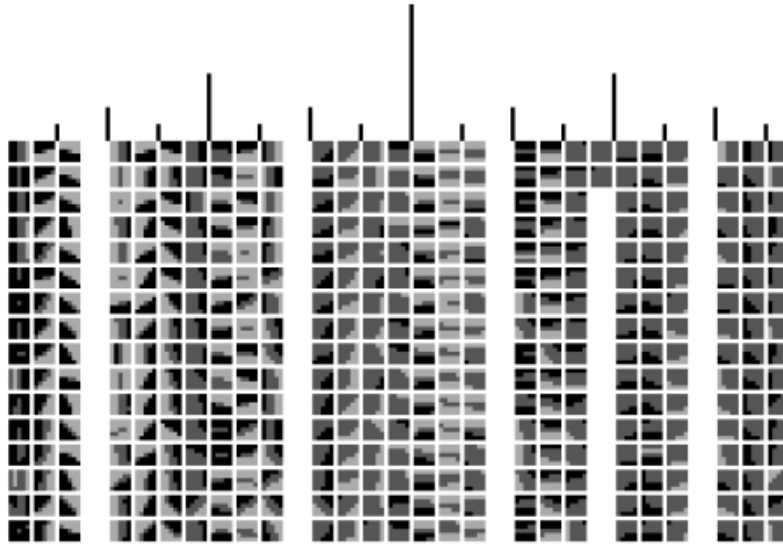


Fig. 3 – l'illustration de l'algorithme sur une imagerie

L'arbre utilisé pour le calcul des codes des imagerie est construit de manière à répartir le plus uniformément possible les imagerie dans les feuilles terminales. Il est construit de la manière suivante : On commence par extraire aléatoirement plusieurs dizaines de milliers d'imagerie 5x5 des images de la base d'apprentissage. Pour construire un nœud, on détermine la meilleure question à y placer. On dispose de trois questions : « le pixel est-il plus clair que le centre », « plus foncé » ou « de même gris », et comme chaque question teste un pixel de l'imagerie, il y a 25 de chaque type, soit finalement 75 questions. Chaque question partage l'ensemble des imagerie qui arrivent à un nœud en deux parties (le sous-ensemble des imagerie qui répondent oui, et celui des imagerie qui répondent non). Finalement, On choisit de placer au nœud la question qui fait le partage le plus équilibré.

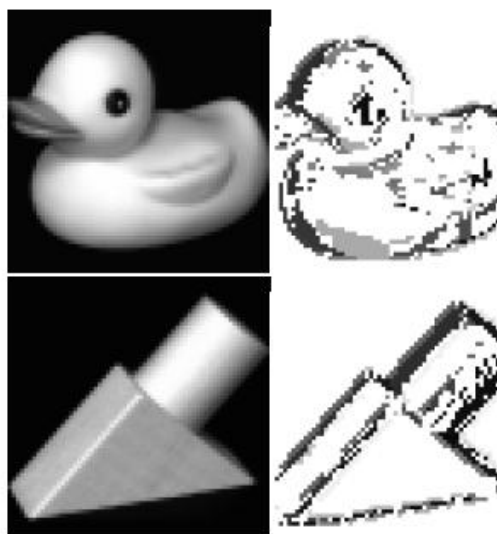


Fig. 4 – La première étape de l'algorithme est un recodage des images, qui permet d'obtenir en chaque point un code décrivant le voisinage.

- ✚ **Notre point de vue (les points positifs + et les points négatifs -) :**
  - + Reconnaissance assez précise.
  - + Reconnaissance d'objets assez complexe.
  - Les benchmark sont vieux.
  - Apprentissage lent.

## Description invariante d'objets 3D représentés par des images géométriques

Cette technique a été décrite dans un communiqué fait par *Moez Chakchouk* et *Faten Chaieb*. En effet, un maillage triangulaire 3D est représenté par une structure régulière 2D appelée image géométrique [publi2]. Cette représentation est obtenue en appliquant une paramétrisation de surfaces triangulaires. Comme cette paramétrisation suppose que les maillages sont homéomorphes à un disque, une procédure de coupe est requise pour les surfaces fermées. Nous proposons d'utiliser l'image géométrique d'un objet 3D afin de ramener le problème de description 3D à un problème 2D. Cette nouvelle description vérifie les propriétés d'invariance aux similitudes et de complétude. Elle est obtenue en calculant les invariants de moments de l'image géométrique associée aux normales de l'objet 3D. Pour cela, on va passer par trois étapes.

La première étape est la paramétrisation de maillages triangulaires. En effet, la paramétrisation d'une surface  $\Gamma$  de  $\mathbf{R}^3$  homéomorphe à un disque est une fonction bijective mettant en correspondance la surface avec un domaine  $\mathbf{D}$  de  $\mathbf{R}^2$ , appelé domaine paramétrique. Elle est définie par :

$$\begin{aligned} \Psi : \quad \mathcal{D} &\longrightarrow \mathbf{R}^3 \\ (u, v) &\longmapsto (x(u, v), y(u, v), z(u, v))^t \end{aligned}$$

La fonction inverse  $\phi = \Psi^{-1}$ , est appelée paramétrisation inverse de la surface  $\Gamma$ .

Si la surface est approchée par un maillage triangulaire 2-manifold noté  $\mathbf{T}_s$  où  $\mathbf{S}$  est un ensemble de points extraits de la surface, la paramétrisation inverse  $\phi$  sera évaluée aux sommets  $\mathbf{S}$  du maillage en considérant une fonction discrète  $\varphi$  1 définie par :

$$\begin{aligned} \varphi : \quad \mathcal{S} &\longrightarrow \mathbf{R}^2 \\ s_i &\longmapsto (u_i, v_i) \end{aligned}$$

La fonction  $\phi$  peut être alors définie par morceaux comme l'interpolation linéaire de  $\varphi$  sur chacun des triangles  $\mathbf{T} = [S_i, S_j, S_k]$  de  $\mathbf{T}_s$ .

Plusieurs méthodes ont été introduites dans la littérature pour déterminer une paramétrisation des maillages triangulaires.

Le calcul de cette paramétrisation par des méthodes linéaires consiste à la considérer comme solution d'un système d'équations linéaires.

En effet, il faut choisir tout d'abord pour chaque sommet  $\mathbf{s}$  du bord du maillage un ensemble de points  $\phi(\mathbf{s})$  transformant le bord en polygone de  $\mathbf{R}^2$ . Ensuite, il faut déterminer pour tout sommet interne, un ensemble de réels strictement positifs pondérant chaque arête concourante à  $\mathbf{s}$  et dont la somme est égale à 1. L'image de chaque point interne  $\mathbf{s}$  par  $\phi$  est la solution unique d'un système d'équations linéaires donné par :

$$\Phi(s) = \sum_{s_k \in V(s)} \lambda_{s,s_k} \Phi(s_k), \forall s \in S_I,$$

Où  $S_I$  est l'ensemble de sommets internes et  $\lambda_{s,s_k}$  les poids pondérant chaque arête concourante au sommet  $\mathbf{s}$ .

La résolution de ce système requiert un choix judicieux des pondérations qui déterminent la nature et les caractéristiques de la paramétrisation. Si les poids sont constants pour chaque sommet, l'image de chaque sommet par la paramétrisation est le barycentre de ses voisins. Il s'agit d'une paramétrisation uniforme.

La deuxième étape est la construction de l'image géométrique d'un objet 3D.

L'idée de base de cette représentation consiste à utiliser la paramétrisation de surfaces couramment utilisée dans les problèmes de plaquage de texture pour ramener un maillage triangulaire irrégulier 3D à une structure 2D régulière. Comme précisé dans la section précédente, la paramétrisation s'applique uniquement à des surfaces homéomorphes à un disque. Par ailleurs, il est bien connu que toute surface fermée peut être transformée en une surface homéomorphe à un disque en la coupant suivant un ensemble approprié d'arêtes. Une méthode introduite dans [1] utilise un algorithme itératif pour déterminer cet ensemble d'arêtes en minimisant l'étirement des triangles dans le domaine paramétrique. L'étirement d'un triangle  $\mathbf{T}$  est défini par :

$$\mathcal{E}(T) = \sqrt{\frac{\alpha^2 + \beta^2}{2}}$$

Où  $\alpha$  et  $\beta$  sont les valeurs singulières du Jacobien de la paramétrisation  $\Psi$ .

Dans ce travail, on a utilisé la paramétrisation de Floater qui permet de conserver la forme des triangles du maillage dans le domaine paramétrique [3]. L'image géométrique  $I$  est obtenue en discrétisant le domaine paramétrique  $\mathbf{D}$  par une grille uniforme de taille  $m \times n$ . Chaque pixel couleur  $(i, j)$  est défini par :

$$I(i, j) = (R(i, j), V(i, j), B(i, j))^t = \Psi(i, j),$$

Où  $R$ ,  $V$  et  $B$  sont respectivement les composantes rouge, vert et bleue de l'image  $I$ . De la même manière, d'autres images géométriques associées aux attributs du maillage 3D peuvent être définies. Dans cette contribution, nous avons utilisé les images géométriques associées aux normales du maillage 3D.

La troisième étape est la description invariante d'un objet 3D en utilisant l'image géométrique, qui consiste tout d'abord à caractériser la transformation entre deux images géométriques associées à deux objets 3D obtenus à une similitude près et à extraire ensuite les primitifs invariants par rapport à cette transformation. La méthode de paramétrisation utilisée et le choix de l'attribut du maillage 3D à partir duquel l'image géométrique est construite caractérisent le type de cette transformation.

Dans ce contexte, on a utilisé la paramétrisation de *Floater* qui permet de conserver les angles des triangles dans le domaine paramétrique. Cette propriété assure que les paramétrisations de deux maillages, obtenus à une similitude près, sont à une rotation près. En effet, cette rotation résulte du changement du point de départ lors du parcours du bord du maillage.

Par conséquent, les invariants d'images couleurs par rapport aux transformations euclidiennes peuvent être utilisés pour décrire les images géométriques associées aux normales. Ici, on propose d'utiliser les invariants de moments largement utilisés en indexation d'images.

Par ailleurs, les opérations de simplification de maillage n'engendrent pas une transformation euclidienne entre les images géométriques. Donc, les moments calculés sur les images géométriques ne semblent pas être appropriés pour décrire un objet 3D d'une manière invariante par rapport à ces transformations.

Sur les figures 5 et 6, on peut voir l'illustration des différentes étapes requises pour la construction des images géométriques respectivement à partir d'un maillage fermé (*mushroom*) et un maillage avec bord (*distcap*). Les images géométriques présentées dans cette section sont de taille 256 x 256.

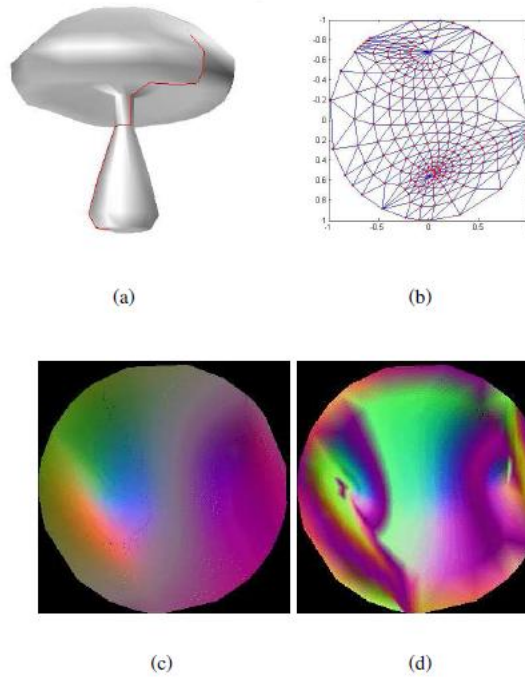


Fig. 5 – Les différentes étapes de construction des images géométriques de l'objet **mushroom**.

- (a) la ligne en rouge représente aux arêtes de la coupe.
- (b) la paramétrisation de Floater du maillage après l'avoir coupé.
- (c) l'image géométrique.
- (d) l'image géométrique associée aux normales du maillage.

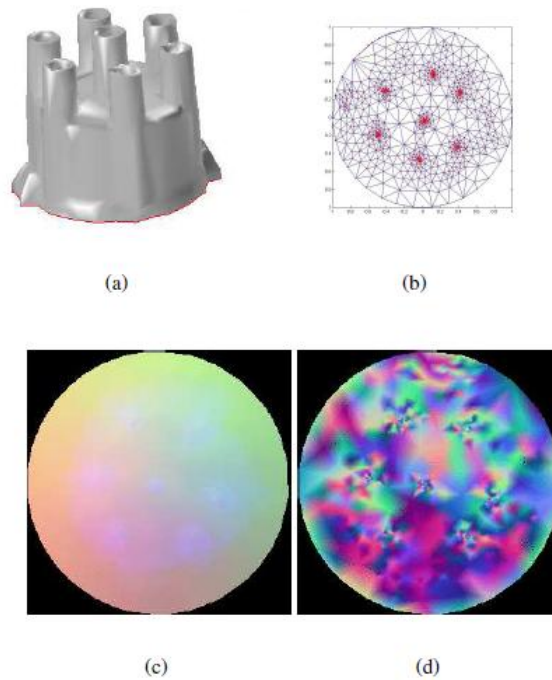


Fig. 6 – Les différentes étapes de construction des images géométriques de l'objet **distcap**.

- (a) la ligne en rouge représente aux arêtes du bord.
- (b) la paramétrisation de Floater du maillage.
- (c) l'image géométrique.
- (d) l'image géométrique associée aux normales du maillage.

- ✚ **Notre point de vue (les points positifs + et les points négatifs -) :**
- + Les objets 3D sont représentés par des images géométriques 2D en couleurs.
  - + Cette description des objets 3D est assez fiable en termes de reconnaissance de l'objet à partir de sa représentation sous forme d'image en couleurs.
  - Les descripteurs de moments invariants entre les objets 3D ne sont pas adaptables aux opérations de simplification de maillage.

## Reconnaissance d'objets par vision focalisée

Cette approche a été proposée dans une publication faite par *Noël Trujillo, Frédéric Chausse et Roland Chapuis*. Elle est capable de s'auto-adapter à l'instance de l'objet [publi3]. En effet, elle utilise un processus de focalisation de l'attention dans la totalité de l'espace des caractéristiques et non pas dans le seul espace géométrique.

Il en résulte une diminution de la combinatoire de recherche des correspondances de caractéristiques par rapport aux approches structurelles classiques. Deux exemples issus de l'analyse de scènes routières ont été présentés à titre d'illustration.

Le processus proposé est récursif : à partir de la valeur initiale des paramètres du modèle de l'objet, une hypothèse est émise pour rechercher la caractéristique la plus nécessaire à sa reconnaissance. Si la détection de cette caractéristique est possible, elle est prise en compte pour remettre le modèle à jour. Sinon, une autre hypothèse est sélectionnée. La remise à jour se traduit par une focalisation de plus en plus serrée qui contribue à accélérer le processus. De plus, l'absence d'extraction d'une caractéristique n'est pas bloquante ; elle conduit simplement à une autre sélection d'hypothèse.

En effet, la forme géométrique de l'objet est modélisée par une fonction. Ce modèle est complété par une représentation multi-échelle. Il inclut des informations d'apparence aussi bien que des informations structurelles cumulant les avantages de chacune. La partie structurelle du modèle est utilisée pour la prise en compte des relations entre des caractéristiques qui ne seront donc jamais considérées comme a priori indépendantes (structuration globale de l'objet).

La partie apparence rend compte de l'aspect local de ces caractéristiques. La définition du modèle est un point crucial de cette méthode.

Le modèle de l'objet est constitué de  $N$  caractéristiques  $F_i$ . Pour chacune d'elle est définie :

– un vecteur de paramètres  $P_i$  et sa matrice de covariance  $C_{P_i}$

– un poids  $w_i$  significatif de la pertinence de la caractéristique dans le processus de reconnaissance

- une fonction de détection  $F_i(P_i, C_{P_i})$  qui fournit une estimation  $\hat{P}_i$  de  $P_i$
- une fonction de coût  $X_i(X, C_x)$

L'organigramme de la figure suivante (figure 7) décrit parfaitement l'algorithme de reconnaissance employé ;

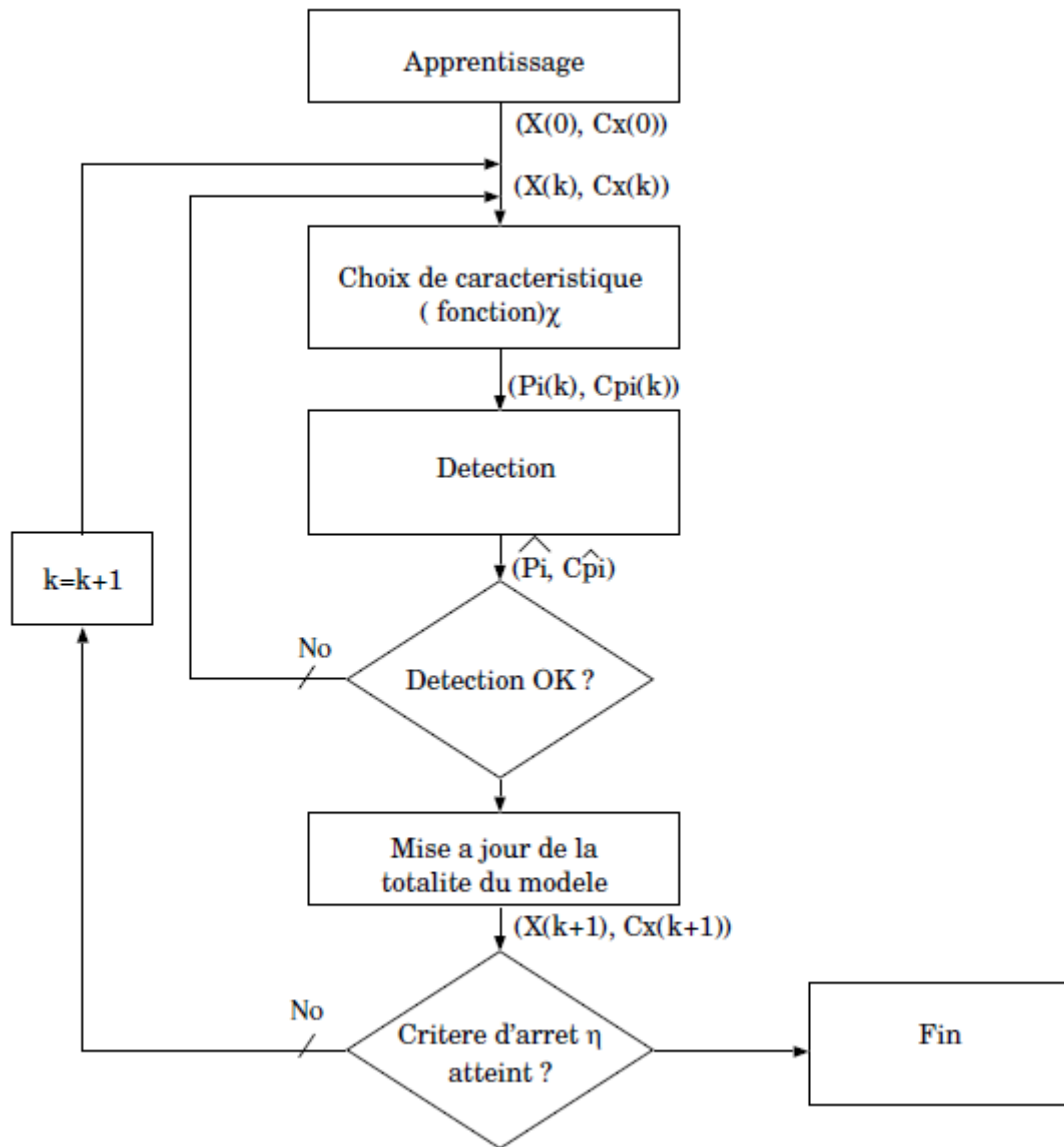


Fig. 7 – Organigramme simplifié de l'algorithme de reconnaissance.

Pour illustrer cet algorithme le premier exemple a pour but de localiser un véhicule sur sa voie de circulation en détectant les bords de cette voie dans les images. C'est une fonctionnalité importante des systèmes de transports intelligents (figure 8).



Fig. 8 – Exemple de reconnaissance de bords de voie.

Le deuxième exemple a comme objectif la reconnaissance dans une image de l'arrière d'un véhicule qui circule devant et dans le même sens que le véhicule porteur de la caméra. L'application concerne la détection et la localisation d'obstacle pour un système d'aide à la conduite.

Le modèle utilisé est constitué de 18 caractéristiques  $F_i$  correspondant au contour de l'arrière du véhicule (figure 9).



Fig. 9 – Position dans l'image des 18 caractéristiques  $F_i$  formant le modèle du véhicule.

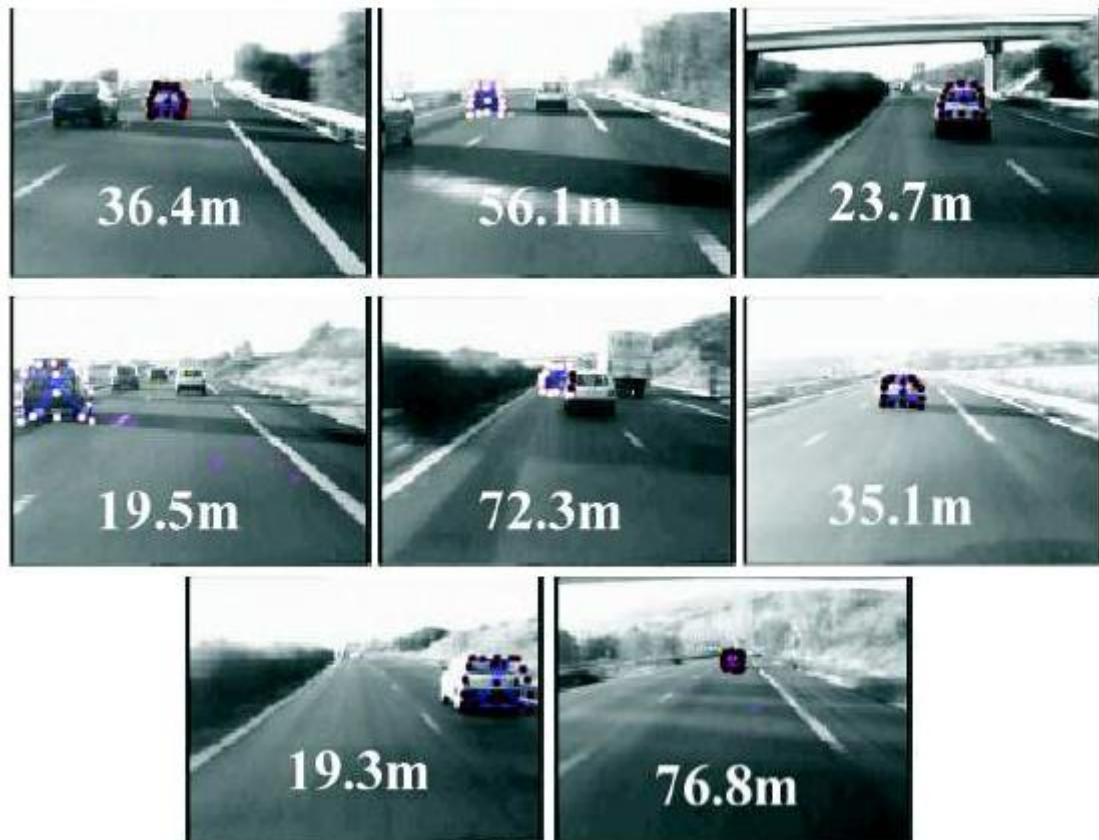


Fig. 10 – Exemple de résultats de reconnaissance de véhicules.

- ✚ **Notre point de vue (les points positifs + et les points négatifs -) :**
- + L'algorithme utilisé est assez compréhensible et donc plus facilement adaptable.
- + Une approche hybride (entre l'approche structurel et l'approche par apparence).
- Lourdeur du calcul.

## Indexation d'objets 3D : approche 2D-3D pour les requêtes à partir d'images

Cette méthode a été décrite dans une publication faite par *Fabrice El Gohary* et *Charles Letailleur*. Elle est un peu similaire à la méthode précédente. En effet, à partir d'une ou plusieurs vues 2D d'un objet, elle permet d'identifier l'objet correspondant dans une bibliothèque d'objets 3D [publi4]. Le programme qu'ils ont développé prend en argument une ou plusieurs images ou même directement un objet 3D. Il compare ces données à la base de données des objets 3D, tout en se basant sur les travaux de *Chen, Tian, Shen* et *Ouhyoung* qui ont développé une méthode très efficace pour la recherche d'objets 3D.

Il s'agit d'une méthode 2D-3D, c'est-à-dire, que même lorsque l'on compare deux modèles 3D, la comparaison se fait via des projections en 2D de ces objets. Cela permet, avec la même méthode, d'obtenir un moyen de soumettre, soit des images 2D soit un modèle 3D. Mais surtout, ce passage par la 2D permet une comparaison rapide des objets. Pour que cette comparaison soit efficace, il ne faut pas, dans ce passage à la 2D, perdre la composante solide de l'objet, c'est-à-dire les relations entre les vues 2D. Ceci se comprend aisément en remarquant qu'une vue de face, et de trois-quart, d'un même objet ne sont pas indépendantes.

Pour comparer les vues 2D entre elles, on utilise des valeurs caractéristiques des images qui sont les moments de Zernike. Les moments de Zernike sont des moments orthogonaux complexes dont la magnitude possède une invariance par rapport à la rotation. Après, pour savoir quelles vues on doit choisir pour l'indexation, il faut, en premier lieu, « discrétiser » l'objet 3D. La méthode *Chen, Tian, Shen* et *Ouhyoung* considère la notion de « lightfield descriptor » qui peut se traduire par « positionnement des cameras ». Cette technique consiste à englober l'objet 3D dans un dodécaèdre et prendre des vues de l'objet depuis les sommets de ce dodécaèdre. Cela nous permettra d'obtenir des vues 2D uniformément réparties autour de l'objet. Mais le plus important réside dans la correspondance entre ces projections et le sommet du dodécaèdre dont elles sont issues. En effet la comparaison de deux objets 3D consiste à faire tourner ces « lightfield » pour retrouver la bonne orientation. On peut penser qu'un lightfield (un dodécaèdre) par objet suffirait mais pour balayer le mieux possible tout l'espace des vues réalisables de chaque objet, on utilise plusieurs lightfields (une dizaine typiquement) pour décrire chaque objet. Ils sont obtenus par décalage (par rotation) du lightfield de base.

La figure suivante résume le principe de lightfield (figure 11) :

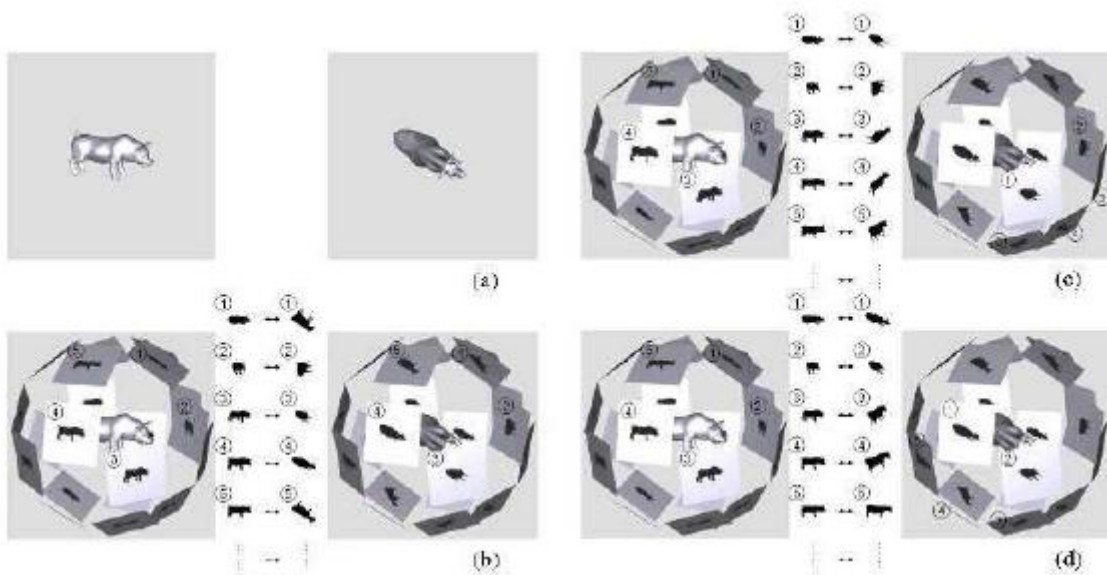


Fig. 11 – Comparaison de descripteurs lightfield entre deux modèles 3D.

Les vues 2D issues de la « discrétisation » précédente sont binarisées pour ne garder que la silhouette. Il faut ensuite trouver un moyen de caractériser numériquement ces images, pour n'avoir que des comparaisons de valeurs numériques (très rapides à effectuer) lors des comparaisons. Pour décrire une forme de manière mathématique, il existe 2 types d'approche : l'approche basée sur les contours et une approche basée sur les régions.

L'approche basée sur les contours s'appuie sur les descripteurs de Fourier. Ils sont calculés par application de la transformée de Fourier à une signature de forme de l'image, la distance centroïde par exemple. Ces descripteurs étaient utilisés dans la méthode donnée pour affiner les résultats (et non en première approximation). Nous ne les avons pas utilisés, et nous ne nous attarderons pas sur eux. Nous nous concentrerons sur les descripteurs les plus discriminants.

Et l'approche basée sur les régions s'appuie sur les moments de Zernike.

**+ Notre point de vue (les points positifs + et les points négatifs -) :**

- + Recherche dans une base d'objets 3D.
- + Vrai bibliothèque 3D utilisé.
- Les descripteurs lightfield peuvent designer plusieurs objets 3D à la fois.

## La méthode par HMM (Hidden Markov Model)

Cette technique est décrite dans la publication d'un étudiant nommé *Nghiem Anh Tuan* et de son professeur *Alain Boucher* en 2005. Dans cette publication, la technique est utilisée pour la reconnaissance de l'écriture manuscrite [publi5]. En effet, quand on utilise HMM pour reconnaître l'écriture manuscrite, chaque mot est modélisé par un HMM. Avec une image entrée, on va la tester sur tous les HMMs, et on va assigner cette image à la classe dont le HMM nous donne la plus grande probabilité d'appartenance. Pour augmenter la vitesse, on peut appliquer la technique de clustering des mots selon le taux *longueur/hauteur*. Donc, l'étape de reconnaissance est divisée en deux sous étapes (figure 12).

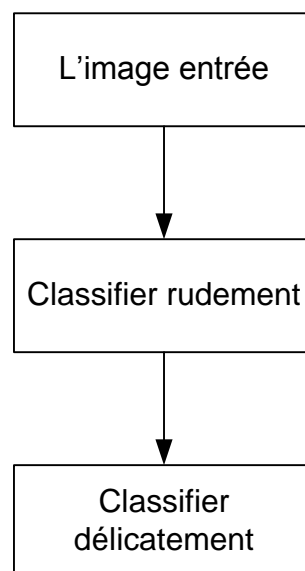


Fig. 12 Les étapes de reconnaissance de la méthode HMM

La première étape c'est de classier grossièrement les mots en groupes de mots dont le taux *longueur/hauteur* est presque le même. Pour réduire les erreurs introduites par ces sous-étapes, on va diviser le taux *longueur/hauteur* en périodes de recouvrement. Par exemple, deux périodes consécutives sont (0, 7; 1, 0) et (0, 9; 1, 2). Si le taux *longueur/hauteur* de l'image d'un mot est 0,95, on va classier ce mot en deux groupes, et le résultat final est la classe qui nous donne la plus grande probabilité d'appartenance.

Pour reconnaître l'écriture manuscrite avec HMM, la caractéristique conventionnelle consiste en des profils verticaux parce que chaque caractère est modélisé par un HMM. Pourtant, cette caractéristique n'est pas stable quand l'orientation des caractères est changée, par exemple lorsque les caractères sont obliqués.

Dans la figure suivante (figure 13), si le caractère est obliqué, le profil vertical a trois périodes noires, sinon, il en a seulement deux.

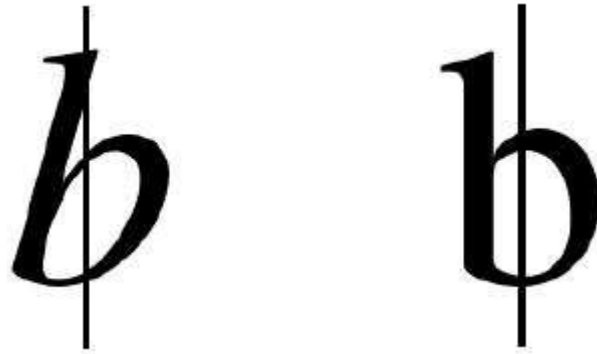


Fig. 13 – Application du profil vertical sur une lettre

Par contre, le profil horizontal est assez stable dans ce cas. Dans la figure suivante (figure 14), le nombre des périodes noires de chaque profil ne dépend pas de la direction du caractère.

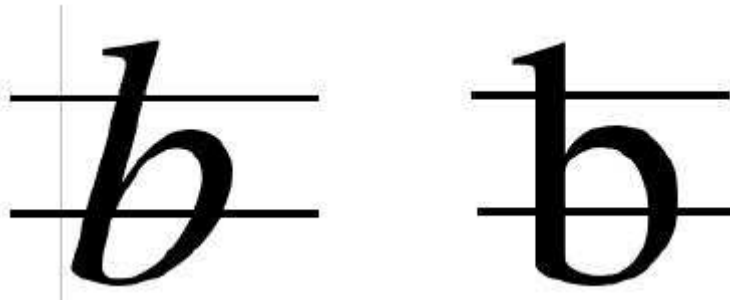


Fig. 14 – Application du profil horizontal sur une lettre

Si on choisit les profils horizontaux comme caractéristiques, il faut normaliser la longueur des profils. Car le taux *longueur/hauteur* varie selon le mot, et on ne peut pas choisir une taille fixe pour tous les mots. Il faut donc avoir un critère pour déterminer la taille de profil pour chaque mot.

Le but d'utiliser les profils horizontaux, c'est de capturer les traits verticaux d'un mot.



Fig. 15 – Application du profil horizontal sur un mot

Par exemple, dans le mot "không", il y a 10 traits verticaux. Donc, on choisit la longueur du profil pour qu'elle soit proportionnelle au nombre de traits verticaux de chaque mot. Cette longueur doit être plus grande que ce nombre. Sinon, il n'y a pas de place pour les périodes blanches sur le profil.

Dans ce système, la longueur d'un profil est égal au nombre de traits multiplié par  $n$  (coefficient choisi). Maintenant, chaque profil est modélisé par un vecteur et il faut déterminer la valeur dans chaque élément de ce vecteur.

Une méthode simple est la suivante :

- Compter le nombre de pixels noirs dans la période correspondant à l'élément courant :  $x$ .
- Calculer le taux  $x/l$  dans lequel  $l$  est le nombre total de pixels dans cette période.
- Assigner ce taux à l'élément courant.

C'est la méthode de normalisation utilisée pour reconnaître des caractères imprimés. Dans ce cas, l'épaisseur des caractères est proportionnelle à leur grandeur. Cependant, ce n'est pas vrai dans notre cas. Pour l'écriture manuscrite, même si le caractère est grand, l'épaisseur de ses traits est presque la même parce qu'on utilise le même stylo. C'est pourquoi on utilise la méthode suivante :

- Compter le nombre de pixels noirs dans la période correspondant à l'élément courant :  $x$ .
- La valeur de cet élément va être 1 dans les cas suivants :  $x > l/2$  ou  $x > \text{seuil}$ .
- Dans les autres cas elle va être 0.

Puis, on résout la question de savoir combien de profils on va prendre pour chaque mot. On s'aperçoit que, dans un mot, horizontalement, il y a trois régions principales.

Dans chaque région, on prend un nombre fixe de profils. Donc, le nombre de profils de chaque mot dépend du nombre de régions couvertes par ce mot.

Par exemple, le mot « nghin` » contient trois régions (figure 16), donc, on va prendre environ six profils. Pour le mot « hai », il contient deux régions, le nombre de profils sera quatre.

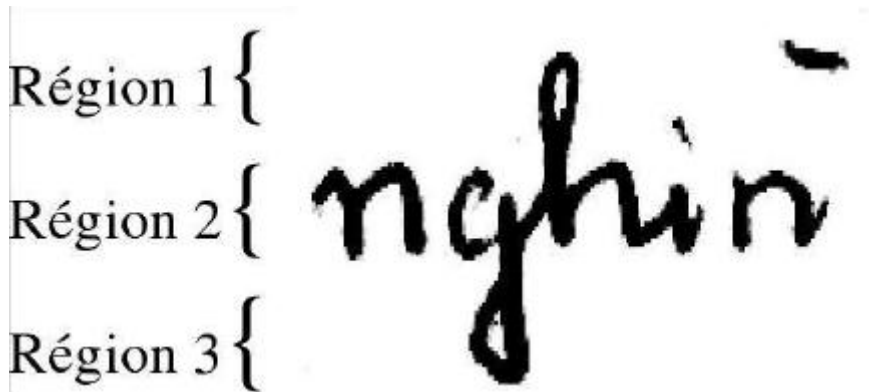


Fig. 16 – Les trois régions qui composent le mot « nghin` »

L'hypothèse utilisée dans cette méthode est : "Le profil qui nous donne le plus d'informations est celui qui traverse le plus de traits".  
 Finalement, on peut résumer l'algorithme pour trouver la position qui prend en compte l'échantillonnage pour chaque mot de la manière suivante :

- Calculer la longueur de chaque pas d'échantillonnage.

$$l = \frac{d \times h}{n}$$

Où **l** : la longueur de chaque pas  
**d** : constant, une factor pour contrôler le nombre de pro\_ls  
**h** : la hauteur du mot  
**n** : nombre de régions du mot

- Chercher les régions pour l'échantillonnage : les régions blanches et les régions qui contiennent des pixels noirs
  - Répéter pour chaque région.
  - Si la région courante est blanche, on prend un profil.
  - Si la région courante contient des pixels noirs, on la divise en plusieurs pas.
  - Dans chaque pas, on prend un profil. C'est le profil valide (déjà défini au-dessus) qui traverse le plus de traits.
  - S'il n'y a pas de traits valides, on prend un profil quelconque.
- ✚ **Notre point de vue (les points positifs + et les points négatifs -) :**
- + Algorithme adaptable à toutes les formes d'écriture.
  - Le nombre de paramètres à prendre en compte est relativement grand.
  - Taux de reconnaissance pas assez important.

## Approche structurelle

C'est dans son rapport de stage effectué chez LIRIS (Laboratoire d'Informatique en Images et Systèmes d'information) en 2005 que *Guillaume Ryder* décrit cette technique de reconnaissance.

Elle s'effectue dans le cadre d'un moteur de recherche : on dispose d'une base de données d'images sur laquelle l'utilisateur peut exécuter une requête en fournissant un modèle [publi6]. Le moteur de recherche doit comparer les images au modèle fourni et retourner un classement par similarité décroissante.

Pour comparer deux images, on distingue habituellement trois niveaux d'interprétation :

- **Bas niveau** : On reste à l'échelle des pixels et du signal pour comparer les images selon la couleur ou les textures, par exemple en leur appliquant un filtre antibruit pour les rendre comparables pixel par pixel.
- **Niveau intermédiaire** : On découpe les images en régions auxquelles l'on attache des propriétés. L'enjeu et la difficulté consistent à trouver des régions qui correspondent à des unités sémantiques de l'image (figure 4), pour que comparer deux images revienne à comparer les propriétés des régions en tenant compte de leur agencement spatial. Par exemple, si l'on recherche un marteau, on voudra trouver une région allongée pour le manche, collée à une autre plus compacte pour la tête. La région « reflet sur le manche », non-porteuse d'information sémantique, ne nous intéresse pas.
- **Haut niveau** : On suppose qu'un système automatique ou opérateur humain a interprété les images en isolant et nommant ses éléments constitutifs. La comparaison se limite à un travail sur des mots-clés.

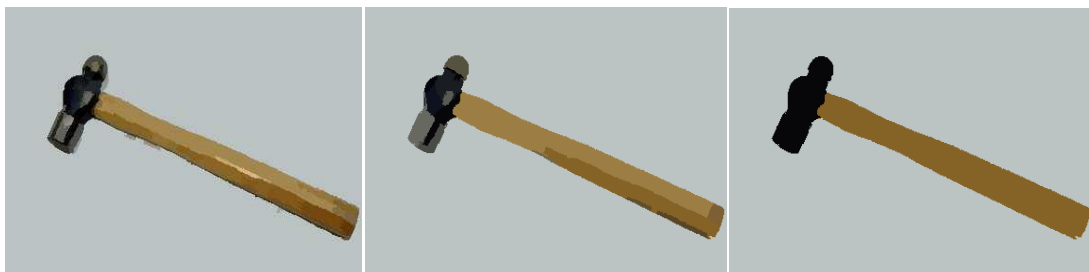


Fig.17 – À gauche, deux segmentations de la même image. Noter la persistance des reflets et des effets d'éclairage. À droite, la segmentation idéale obtenue manuellement.

Dans cette étude, on a privilégié le niveau intermédiaire, dit également structurel car il concerne la structure de l'image, son « squelette spatial ».

Et cela dans un triple objectif :

- Rapidité : La recherche s'effectue sur une quantité à la fois restreinte et discriminante de données.
- Robustesse : S'élever au niveau structurel assouplit la comparaison : la notion de sous-objet est robuste aux changements de taille, de couleur ou de position entre image et modèle. Cela de manière prévisible et ajustable: on peut trouver les bons paramètres par le calcul, sans tâtonnement, et facilement les adapter d'une série d'images à l'autre. A l'inverse, les approches basées sur des pixels sont difficiles à paramétrer numériquement et sont parfois très sensibles aux conditions d'éclairage.
- Automatisation : On ne demandera pas à l'utilisateur de designer ni de nommer les sous-objets de son modèle, le moteur devra pouvoir le faire tout seul à partir d'une image du modèle.

✚ **Notre point de vue (les points positifs + et les points négatifs -) :**

- + Reconnaissance rapide.
- + Base de donnée correct (600 objets / 30s).
- Objet structurellement simple.
- Pas de création automatique de la base de données.

## La méthode HTM

Cette technique a été décrite par *Jeff Hawkins* et *Dileep George* dans un résumé fait au sein de Numenta, société fondée le 24 mars 2005, par Jeff Hawkins et son associé de longue date Donna Dubinsky, et l'Étudiant de troisième cycle de Stanford Dileep George. Cette société est basée dans la Ville de Redwood, Californie. Ce papier décrit la théorie à travers les produits de Numenta, mais ne décrit pas les produits eux-mêmes [publi7].

La documentation séparée décrit ces produits et l'application de la technologie HTM aux problèmes du monde réel.

Les HTMs sont organisés comme une hiérarchie sous forme d'arbre de nœuds, où chaque nœud met en œuvre une étude commune et une fonction de mémorisation. Ces HTMs stockent les informations dans toute la hiérarchie de manière à modéliser le monde extérieur. Tous les objets dans le monde tel que les voitures, les personnes, les bâtiments, les discours, ou les flux d'informations à travers le réseau d'un ordinateur ont une structure. Cette structure est hiérarchique dans l'espace et le temps.

La mémoire des HTMs est hiérarchique aussi dans l'espace et le temps, et peut donc efficacement capturer et modéliser le monde extérieur.

Les HTMs sont semblables aux réseaux Béyesiens; cependant, ils diffèrent de la plupart de ces derniers dans la façon avec laquelle le temps, la hiérarchie, l'action et l'attention sont utilisés. Ils peuvent être mis en œuvre avec un logiciel sur le matériel informatique traditionnel, mais il est préférable de penser à un système de mémorisation.

Les HTMs exécutent les quatre fonctions de base suivantes indépendamment du problème particulier auquel ils sont appliqués. Les deux premiers sont obligatoires et les deux derniers sont facultatifs.

- **Découvrir des causes dans le monde**

La figure 1 montre comment un système HTM communique avec le monde. À gauche de cette figure, on a une boîte représentant un monde que le HTM doit apprendre. Le monde englobe les objets et les relations qui les relie. La plupart des objets dans le monde sont des structures physiques : personnes, véhicules, bâtiments ... Mais la plupart des objets dans le monde peuvent ne pas être des structures physiques : idées, sons, ou flux d'informations dans un réseau.

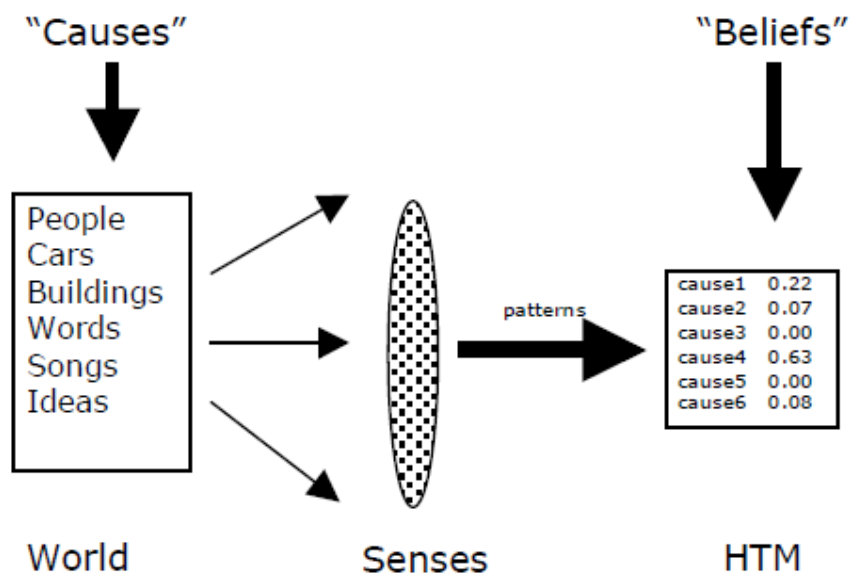


Fig. 18 – schéma représentatif de la façon avec laquelle un système HTM interagit avec le monde

Le HTM à droite de la figure interagit avec le monde à travers un ou plusieurs sens représentés au milieu de la figure. Les sens présentent des échantillons de quelques attributs du monde comme la lumière ou le contact, quoique les sens utilisés par un HTM ne doivent pas être exactement les mêmes que ceux des humains.

Typiquement les sens ne détectent pas directement les objets dans le monde. On n'a pas un sens spécial pour les voitures ou bien un autre pour les mots.

En effet, l'un des buts recherché par un HTM est de déduire, à partir de l'apport sensoriel brut, des objets tels que des voitures ou des mots.

Concrètement, pour un HTM, les sens présentent un tableau de données, où chaque élément du tableau est une certaine mesure d'un faible attribut du monde. Dans le corps humain, le nerf optique qui porte les informations de la rétine au cortex est formé d'environ un million de fibres où chaque fibre porte des informations sur la lumière dans une petite partie de l'espace visible.

Le nerf auditif est formé d'environ trente mille fibres, où chaque fibre porte des informations sur le son dans une petite gamme de fréquence. Les sens attachés à un HTM auront généralement un arrangement semblable. C'est-à-dire que les données sensorielles seront une collection topologiquement rangée d'apports, où chaque apport mesure une quantité locale et simple.

- **Déduire les causes de nouvel apport**

Après qu'un HTM a appris quelles sont les causes dans son monde et comment les représenter, il peut exécuter « l'inférence ». Cette dernière est similaire à la reconnaissance des formes. Étant donné un nouveau flux d'entrée sensoriel, un HTM "déduira" quelles causes connues vont probablement être présentes dans le monde à ce moment là. Par exemple, si on a un HTM basé sur un système de vision, on peut lui montrer des images et il déduira les objets correspondants. Le résultat va être une distribution de concepts à travers toutes les causes apprises. Si l'image est ambiguë, la distribution de concepts atteindra un niveau maximal. Dans le cas d'une image extrêmement ambiguë, la distribution de concepts serait plate parce que le HTM n'était pas certain de ce qu'il regardait.

Les concepts courants déduits par un HTM peuvent être directement lus par le système et être exploités ailleurs, à l'extérieur du HTM (ce qui est impossible à accomplir par les cerveaux humains !). Alternativement, la croyance actuelle peut être utilisée à l'intérieur par le HTM pour faire des prédictions ou produire un comportement.

Dans la plupart des systèmes HTM, l'apport sensoriel sera toujours quelque chose de nouveau. Dans un système de vision attaché à une caméra, l'apport sensoriel peut être sous forme d'un million de pixels. Si la caméra regardait des scènes réelles du monde, il est fortement peu probable que le même modèle serait présenté au HTM deux fois.

Ainsi, les HTMs doivent manipuler de nouveaux apports tant pendant l'inférence que pendant l'apprentissage. En fait, les HTMs n'ont pas de mode d'inférence séparé. Ils déduisent toujours des causes en apprenant en même temps (en déduisant mal avant d'effectuer une grande partie de l'apprentissage).

Comme il a été mentionné plus tôt, il est possible de désactiver l'apprentissage après la formation et continuer à faire de l'inférence.

La déduction des causes de nouvel apport est très importante. Il y a beaucoup de problèmes de reconnaissance de formes que les gens trouvent faciles mais les algorithmes informatiques existants sont incapables de les résoudre. Les HTMs peuvent résoudre ces problèmes d'une manière rapide et précise, exactement comme les humains.

De plus, il y a beaucoup de problèmes d'inférence que le cerveau humain a du mal à résoudre tandis que les systèmes basés sur des HTMs sont capables de le faire.

- **Faire des prédictions**

Les HTMs consistent en une hiérarchie de nœuds de mémoire où chaque nœud apprend des causes et forme des croyances. La partie de l'algorithme d'apprentissage exécuté par chaque nœud doit stocker les ordres probables des modèles. En combinant la mémoire d'ordres probables avec l'apport actuel, chaque nœud a la capacité de faire les prédictions de ce qui va arriver probablement par la suite. Un HTM entier, étant un ensemble de nœuds, fait aussi des prédictions. De même qu'un HTM peut déduire les causes d'un nouvel apport, il peut aussi faire des prédictions concernant de nouveaux événements.

La prévision de l'avenir de nouveaux événements est l'essence de la créativité et de la planification. En mettant de côté les détails de fonctionnement de ces travaux, on peut maintenant exposer la raison pour laquelle une prédiction peut être utilisée. Il y a plusieurs utilisations pour la prédiction dans un HTM, y compris l'amorçage, l'imagination et la planification, et la génération de comportement.

- L'amorçage :

Quand un HTM prévoit ce qu'il va probablement arriver par la suite, la prédiction peut agir comme ce qu'on peut appeler « une probabilité antérieure », ce qui veut dire que le système va être influencé pour déduire les causes prévues. Par exemple, si un HTM traitait le texte ou la langue parlée, il prévoirait automatiquement quels sont les sons, les mots et les idées qui vont probablement suivre. Dans un HTM, on a la possibilité de régler manuellement des probabilités antérieures, en plus d'avoir un ensemble de probabilités antérieures via la prédiction.

C'est-à-dire qu'on peut manuellement dire au HTM de prévoir ou de chercher une cause particulière ou un ensemble de causes, mettant ainsi en œuvre une recherche dirigée.

➤ l'imagination et la planification :

Les HTMs prévoient et anticipent automatiquement ce qui va probablement arriver par la suite. Au lieu d'utiliser ces prédictions pour l'amorçage, les prédictions d'un HTM peuvent être retransmises dans le HTM comme une compensation pour les données sensorielles. Ce processus est similaire à ce que font les humains quand ils pensent. La pensée, l'imagination, la planification de l'avenir et la répétition silencieuse dans nos têtes représentent la même chose et sont réalisés par une série de prédictions. Toutes ces opérations peuvent être aussi bien effectuées par un HTM.

L'imagination de l'avenir peut être précieuse dans beaucoup d'applications. Par exemple, supposez qu'une voiture est équipée d'un HTM pour évaluer le trafic autour d'elle. Si une nouvelle situation survient, le HTM peut suivre une série de prédictions pour prévoir quels sont probablement les événements qui vont se produire, et peut donc imaginer les situations dangereuses avant qu'elles ne surviennent.

La prédiction est aussi au cœur du mécanisme avec lequel les HTMs peuvent gérer les comportements, ce qui est aussi la quatrième et dernière capacité d'un HTM.

- **Comportement direct**

Lorsqu'un HTM a appris les causes dans son monde, et le mode d'évolution de ces dernières dans le temps, on peut dire qu'il a créé un modèle de son monde. On suppose maintenant qu'un HTM est attaché à un système qui interagit physiquement avec le monde. On peut imaginer un HTM attaché à un robot, mais il ne doit pas y être limité. Ce qui est important c'est que le système peut déplacer ses capteurs à travers son monde et/ou manipuler des objets dans ce dernier. Dans un tel système, le HTM peut apprendre à produire un comportement avec des objectifs à la fois complexes et bien orientés. On donnera une explication brève ici.

La figure suivante (figure 19) montre un système avec un HTM ayant la capacité de produire des comportements simples. Les composants du moteur de ce système ont des comportements incorporés, "réfléchis", ou câblés. Ceux-ci sont les comportements simples qui existent indépendamment du HTM.

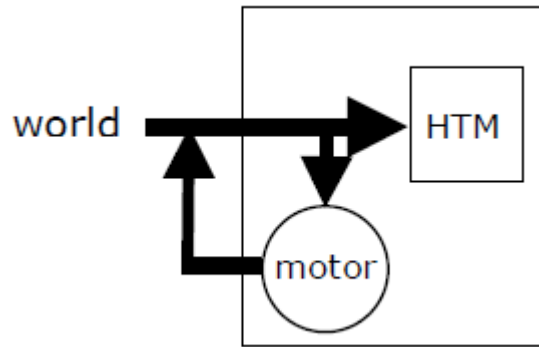


Fig. 19 – Un HTM modélise le monde en construisant les représentations des causes, y compris les comportements programmés du moteur

Au même temps qu'un HTM découvre les causes dans son monde, il apprend à représenter ses comportements incorporés ainsi que ceux des objets dans le monde extérieur. En se basant sur la perspective du HTM, le système auquel il est connecté est simplement un autre objet dans le monde. Le HTM forme les représentations des comportements auxquels le système est attaché, et ce qui est important c'est que ce dernier apprend à prévoir son activité.

Ensuite, par un mécanisme de mémoire associatif, les représentations de base d'un HTM des comportements incorporés sont appareillées avec les mécanismes créant les comportements incorporés eux-mêmes (figure 20). Après cette union associative, quand le HTM invoque la représentation interne d'un comportement, il ne peut causer que le déclenchement de ce dernier. Si le HTM prévoit qu'un comportement arrivera, il peut provoquer son arrivée anticipée. Maintenant le HTM est dans une position qui lui permet de diriger le comportement.

En alignant les ordres de ces comportements simples, il peut créer des nouveaux comportements orientés vers un objectif bien précis. Pour cela, le HTM exécute des étapes identiques à celles qu'il franchit quand il produit une série de prédictions et imagine l'avenir. Cependant, au lieu de simplement imaginer l'avenir, le HTM aligne les comportements incorporés pour les réaliser en réalité.

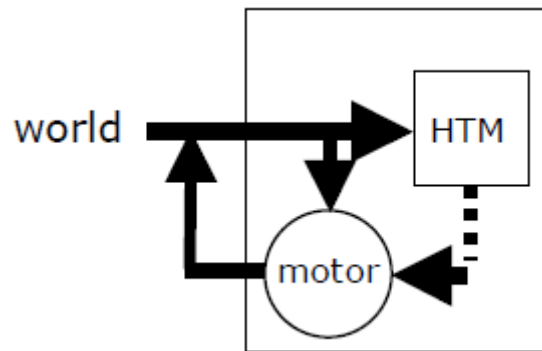


Fig. 20 – Les représentations de comportement du moteur sont auto-associativement appareillées avec les générateurs du moteur ce qui permet au HTM de gérer le comportement.

On peut observer les bases de ce comportement en apprenant le mécanisme de notre propre corps. Les comportements comme les mouvements de l'œil, le mâchement, la respiration, la rétractation du bras au touché d'un objet coupant, le fait de marcher et même celle de courir sont générés en grande partie dans les parties les plus vieilles du cerveau, et pas dans le néocortex. La plupart du temps, ces comportements sont produits avec peu de participation, ou sans la participation néocorticale. Par exemple, généralement nous ne sommes pas conscients de la façon dont notre mâchoire et notre langue se déplacent pendant la mastication, ou de la façon dont nos jambes se déplacent pendant la marche, de même que nous sommes normalement inconscients de notre respiration.

Cependant, on peut consciemment contrôler notre respiration et les mouvements de nos yeux, ou même marcher d'une manière inhabituelle. Quand on fait cela, c'est que notre néocortex est sous contrôle. Ce dernier ignorait ces comportements au moment de la naissance de l'humain. Il a dû en faire l'apprentissage en se basant simplement sur la manière dont ils étaient décrits.

Les HTMs peuvent diriger le comportement de beaucoup de systèmes de type différents; ils ne sont pas limités à la robotique traditionnelle. Si on imagine un immeuble de bureaux avec chauffage et climatisation, il y a des commandes séparées de température pour chaque étage. C'est là qu'on attache un HTM au bâtiment. Des apports sensoriels au HTM consistent à installer des capteurs de température partout dans le bâtiment ainsi que des commandes pour régler la température. Le HTM pourrait aussi obtenir des apports représentant le moment de la journée, le nombre de personnes entrant et sortant de l'immeuble, la température actuelle à l'extérieur, etc.

Au moment où le HTM apprend, il construit un modèle du bâtiment, qui inclut la façon dont les commandes de température réagissent aux différents événements qui surviennent à l'extérieur et à l'intérieur du bâtiment.

Maintenant, le HTM utilise son modèle pour prévoir des événements, incluant les moments où les commandes de température seront activées et désactivées, ou bien à quel moment il faudra augmenter ou baisser la température.

En combinant les représentations internes du HTM de ces actions avec les commandes de température, le HTM peut commencer à diriger "le comportement" du bâtiment. Le HTM peut être meilleur dans l'anticipation des demandes maximales et être donc meilleur dans le maintien de températures souhaitables, ou dans la réduction de consommation d'énergie.

**+ Notre point de vue (les points positifs + et les points négatifs -) :**

- + Les propriétés algorithmiques d'un HTM sont très flexibles et donc une fois maîtrisés, ces algorithmes peuvent être appliqués parfaitement dans des domaines autres que les capacités humaines.
- + Chaque nœud dans le réseau d'un HTM peut utiliser sa propre mémoire d'ordres pour prévoir ce qui devrait arriver ensuite. Ce qui rend les prédictions et la génération de comportement plus logiques et adéquates par rapport aux situations confrontées.
- L'utilisation est relativement compliquée pour résoudre des problèmes relativement simples.

## Autres méthodes

Il existe tellement de méthodes de reconnaissance de formes qu'il n'est pas possible d'en faire une liste exhaustive. Lors de nos recherches, nous en avons étudié un nombre important, celles qui ne sont pas citées se trouvant en référence dans notre bibliographie.

## Base choisie à notre méthode

### Introduction

On a choisit comme base la méthode « Indexation d'objets 3D : approche 2D-3D pour les requêtes à partir d'images ». Elle se base en grande partie sur **la technique du dodécaèdre**. En effet, cette méthode est assez adaptable à notre réseau de neurones qu'on veut ajouter pour renforcer notre technique de reconnaissance.

Une grande partie de cette méthode développée se base sur les travaux effectués dans ce domaine par *Chen, Tian, Shen* et *Ouhyoung*.

### Présentation de la technique du dodécaèdre

L'idée de base de cette technique s'inspire de cette phrase : « si deux modèles 3D sont similaires c'est qu'ils ont les mêmes représentations des différents points de vue qu'on peut avoir d'eux ».

Donc la similarité entre deux modèles 3D se mesure par un test de similarité entre toutes les images issues de chacun de leurs lightfields.

La figure suivante (figue 21) montre les images des silhouettes d'une chaise.

Ces images sont prises avec dix angles de vues différents.



Fig. 21 – Un exemple typique présentant les dix silhouettes d'un modèle 3D.

Notre idée est d'utiliser ces différentes images pour chaque modèle 3D et les affecter aux entrées d'un réseau de neurones multicouches classique. Les sorties de notre réseau vont correspondre à l'identifiant de l'objet 3D.

## Le dodécaèdre dans notre cas

On utilise un dodécaèdre convexe à 8 sommets.

On placera nos prises de vues sur ses sommets ce qui nous donnera huit représentations différentes de chaque objet.

La distance de prise de vue de l'objet est calculé en fonction de la taille de ce dernier, afin d'avoir une représentation correcte quelle qu'elle soit.

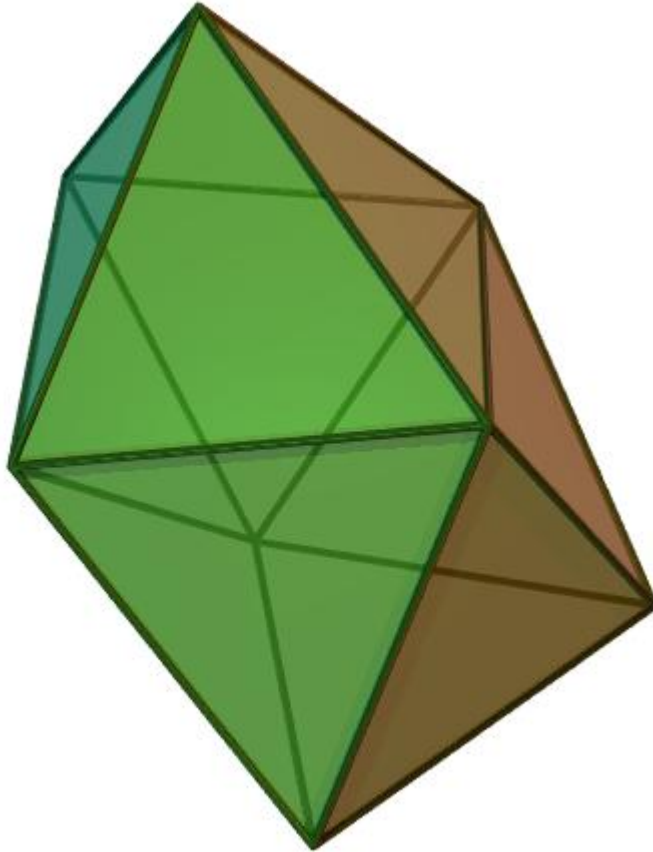


Fig. 22 – Représentation graphique du dodécaèdre utilisé.

## Conclusion

Théoriquement, avec l'utilisation de cette technique et l'ajout d'un réseau de neurones bien adapté, on va aboutir à une reconnaissance plus intelligente et plus complète.

# Amélioration de la méthode Indexation d'objets 3D

## Introduction

Dans cette partie nous allons définir les objectifs de notre méthode qui nous permettrons d'aboutir à une technique de reconnaissance de formes plus complexe et qu'il sera possible d'exploiter dans des nouveaux domaines plus exigeants en terme de précision et de rapidité tel que le jeu vidéo et la médecine.

## Nos objectifs

Nous avons, lors de la phase de réflexion sur notre méthode de reconnaissance de forme, réfléchi à différents objectifs, qui pourraient permettre d'améliorer la reconnaissance de forme.

Les voici :

### Reconnaissance d'objet complexe

La plupart des techniques de reconnaissance d'objet se contentent de reconnaître les objets par grandes familles. Par exemple, on pourra reconnaître une lettre parmi d'autres lettres, ou un objet spécifique comme une table, une chaise, un avion ou une voiture. La plupart de ces techniques ne proposent pas de distinguer une table basse d'une table de jardin, ou un avion de plaisance d'un avion de chasse. Mais c'est un des objectifs à atteindre dans l'avenir. De notre côté, nous n'irons sûrement pas jusqu'au bout de cet objectif, mais afin de pouvoir s'en rapprocher le plus possible, nous intégrons la reconnaissance de la couleur à notre méthode et augmentons ainsi la précision de notre reconnaissance afin de l'appliquer à des objets plus complexes.

### Optimisation du temps de reconnaissance

Plus qu'un objectif, l'optimisation du temps de reconnaissance est une nécessité. Pouvoir reconnaître sur des systèmes plus ou moins performants avec une vitesse correcte, peut être très utile dans de nombreux domaines, telle que la robotique, où le temps de reconnaissance d'un objet doit être de courte durée, ou encore le jeu vidéo, où il n'est pas pensable que la réactivité soit réduite à cause du temps de reconnaissance. Egalement dans le domaine de la médecine, où les cas urgents doivent nécessiter une reconnaissance rapide. Pour notre méthode nous avons donc pour objectif de réduire au maximum le temps de reconnaissance en augmentant celui de l'apprentissage, le temps d'apprentissage n'étant pas une notion importante dans le cadre d'une simple reconnaissance d'objet.

## Ignorer les bruits

Dans les contextes réels, les objets ne sont pas isolés au milieu de nulle part : il y a toujours d'autres objets, un environnement autour d'eux. C'est ce que l'on appellera dans notre cas le bruit. Ignorer le bruit permet ainsi de reconnaître un objet en partie caché, ou tout simplement d'ignorer ce qui entoure l'objet et qui pourrait perturber sa reconnaissance.

## Utilisation d'un réseau de neurones

### Description

Un réseau de neurones artificiel est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement des neurones biologiques [wikip2].

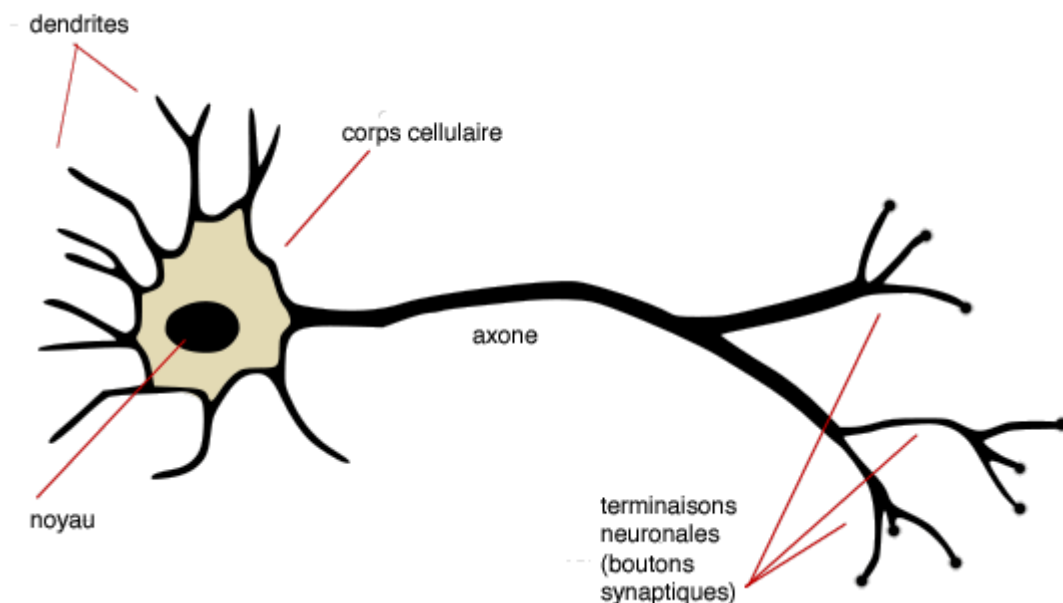


Fig. 23 – Représentation schématique d'un neurone.

Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type statistique, si bien qu'ils sont placés d'une part dans la famille des applications statistiques, qu'ils enrichissent avec un ensemble de paradigmes permettant de générer de vastes espaces fonctionnels, souples et partiellement structurés, et d'autre part dans la famille des méthodes de l'intelligence artificielle qu'ils enrichissent en permettant de prendre des décisions s'appuyant davantage sur la perception que sur le raisonnement logique formel .

En modélisation des circuits biologiques, ils permettent de tester les hypothèses fonctionnelles issues de la neurophysiologie ou de tester les conséquences de ces hypothèses afin de les comparer aux réseaux réels.

Il existe différents types de réseaux de neurones, on peut les classer de la manière suivante:

- Réseaux à apprentissages supervisés
  - Sans rétropropagation
    - ✓ Perceptron monocouche
    - ✓ Perceptron multicouche
    - ✓ Adaline (ADaptive LInear NEuron)
    - ✓ Machine de Cauchy
  - Avec rétropropagation
- Réseaux à apprentissage non supervisé

Le réseau de neurone qu'on va utiliser pour améliorer notre technique de reconnaissance de formes est un réseau multicouche avec une seule couche cachée.

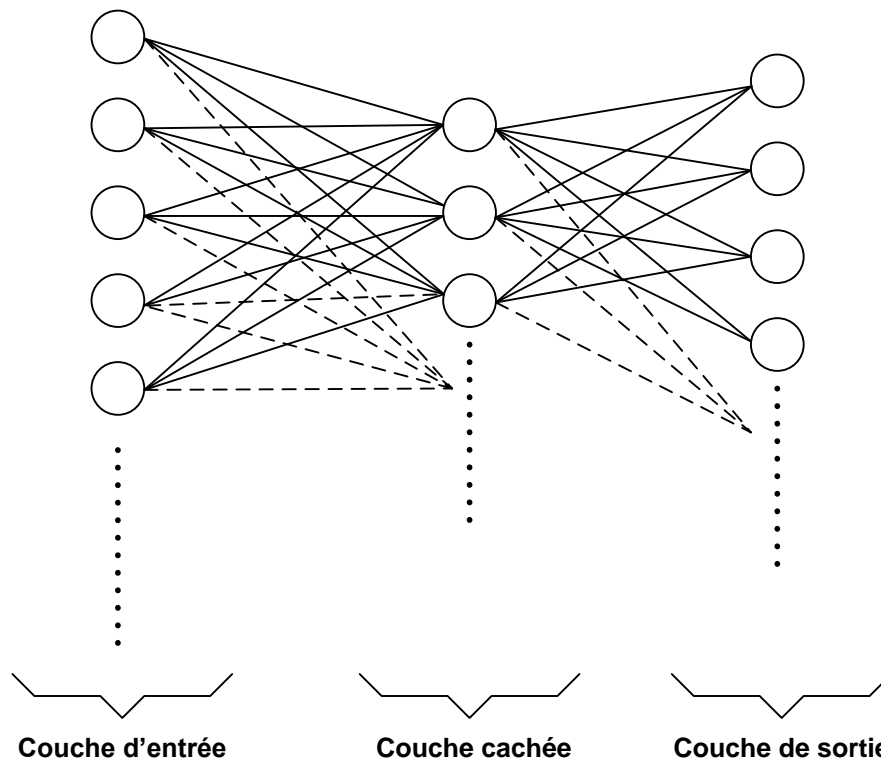


Fig. 24 – Représentation schématique d'un réseau de neurones standard avec une seule couche cachée.

## Réseau de neurone utilisé

Notre réseau de neurones est un réseau multicouche utilisant un algorithme de rétropropagation pour l'apprentissage, et il possède une couche cachée, son nombre d'entrée  $N$  dépend directement de la taille de l'image.  $N$  est donc défini de la manière suivante :  $N = (T/P)*3$ ,  $T$  représentant la taille de l'image,  $P$  un nombre carré ( $1^2, 2^2, 3^2 \dots$ ) représentant la précision, et 3 correspondant aux trois couleurs de base d'une image (RGB). Si le nombre de neurones de sortie est égal au nombre d'objets que l'on veut enregistrer dans notre base de données. Ainsi chaque neurone de sortie correspondra à un objet différent. Concernant le nombre de neurones composant la couche cachée  $C$  de notre réseau, il est compris entre  $N$  et  $S$  ainsi  $C \in [N, S]$ .

Les neurones d'entrée recevront eux l'image en couleur découpée de la manière suivante :

On prend l'image d'origine et on prend un nombre de pixels en carré dépendant de la précision, et on fait la moyenne de chacune des couleurs des pixels pour chaque zone ainsi définie. Les couleurs de chaque zone sont ensuite mises en entrée dans notre réseau de neurones.

Par exemple, pour la reconnaissance d'un objet, on entre une image dans le réseau de neurones en la soumettant au même découpage que les images mises en entrée lors de l'apprentissage, ensuite on regarde tous les neurones de sortie du réseau, et celui qui a la valeur la plus élevée correspond à l'objet reconnu. Si la valeur est inférieure à un certain pallier, aucun objet n'a été reconnu dans l'image.

Dans le schéma suivant (figure 25) on a pris comme exemple une image de taille 27 et on a fixé la précision à 3.

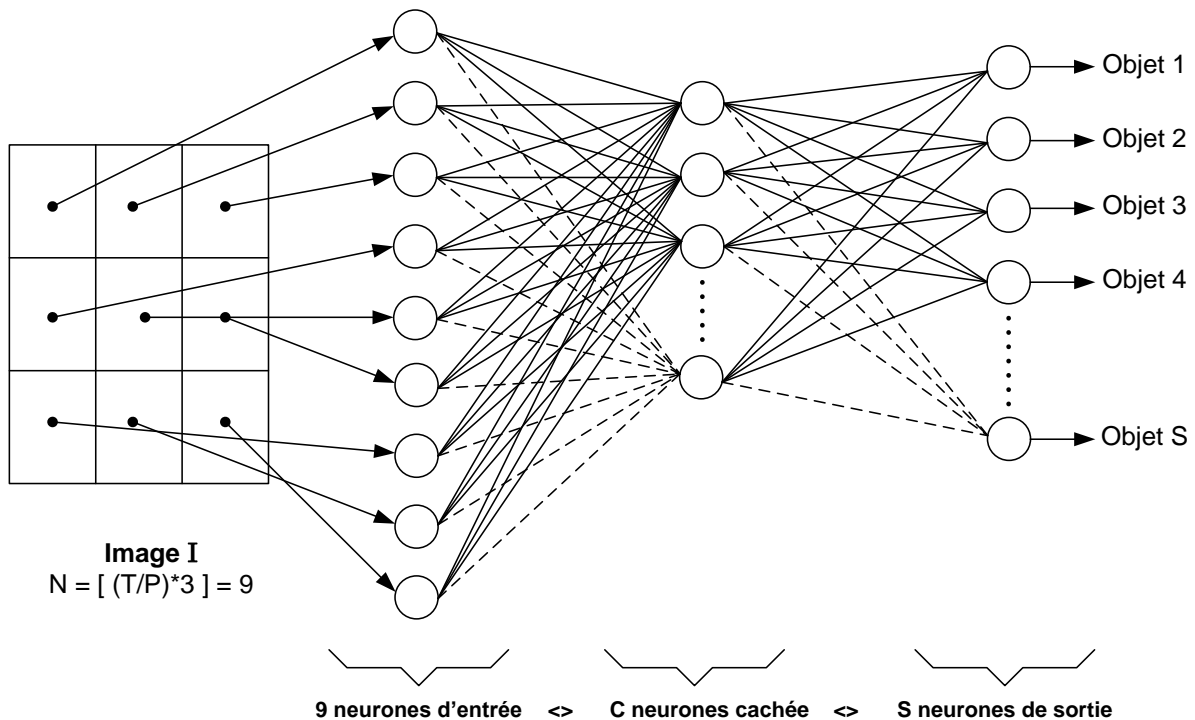


Fig. 25 – Schéma représentatif de notre réseau de neurones (avec  $N = 9$ ).

### Les apports de notre réseau de neurone

L'utilisation de notre réseau de neurones avec le système du dodécaèdre offre de nombreux avantages par rapport à d'autres techniques existantes. Dans un premier temps, le principal problème pour la reconnaissance à partir d'un réseau de neurones est qu'il faut lui fournir suffisamment d'informations pour pouvoir reconnaître l'objet en toutes circonstances, chose réalisable facilement et automatiquement par un réseau de neurones. En effet, à partir d'une banque de données d'objets 3D, il est possible de réunir automatiquement une vue d'ensemble de chaque objet.

Dans un second temps, notre réseau de neurones a la particularité de prendre en compte pour chaque vue de l'objet ses couleurs. Ainsi, ce dernier ayant une particularité basée sur sa couleur, pourra être reconnu. Le réseau pourra facilement faire la différence entre deux objets ayant les mêmes caractéristiques physiques mais ayant une couleur différente, tels un citron et un citron vert, chose impossible avec les reconnaissances en niveau de gris ou les reconnaissances de contours.

De plus, le réseau de neurones fournit des informations approximatives : il ne dira pas tel objet correspond à tel objet mais informera que tel objet correspond à X% à celui reconnu sur l'image. Ainsi un objet au milieu d'autres objets est tout de même potentiellement reconnaissable, de même pour un objet ayant une singularité ou étant partiellement caché.

Le réseau de neurones offre donc une souplesse que n'offrent pas d'autres systèmes de reconnaissance.

Pour finir, une autre possibilité pour obtenir ces points aurait consisté à faire une reconnaissance 3D pure, sans utiliser le dodécaèdre, en apprenant à un réseau de neurones toutes les coordonnées de l'objet dans un espace 3D. Mais une telle méthode n'aurait pas permis de reconnaître à partir d'une image 2D un objet 3D, tel que le fait une méthode utilisant le dodécaèdre.

## Performances

Nos performances ont été calculées sur un processeur **Intel core 2 duo 2.5 GHz**. Mais notre outil de test n'exploite qu'un seul des deux cœurs.

Nos premiers résultats ont été réalisés sur cinq objets différents, dont un cube vert et un cube jaune afin de tester la reconnaissance de la couleur. Pour cette première série de test, nous avons choisi pour nos images une précision de 2, ce qui nous a donné des images en 140\*195 soit 81 900 entrées. Pour cela, nous avons choisi de façon empirique de prendre dix neurones dans la couche cachée, et nous avons bien sûr cinq neurones pour notre sortie. De plus, nous avons un nombre de boucles d'apprentissage de 150 et un taux d'apprentissage de 0.75.

Pour cette première série de test, nous avons fait notre reconnaissance sur un emplacement de la camera ne se trouvant pas sur le dodécaèdre.

Voici les vues mémorisées pour l'objet Jeep

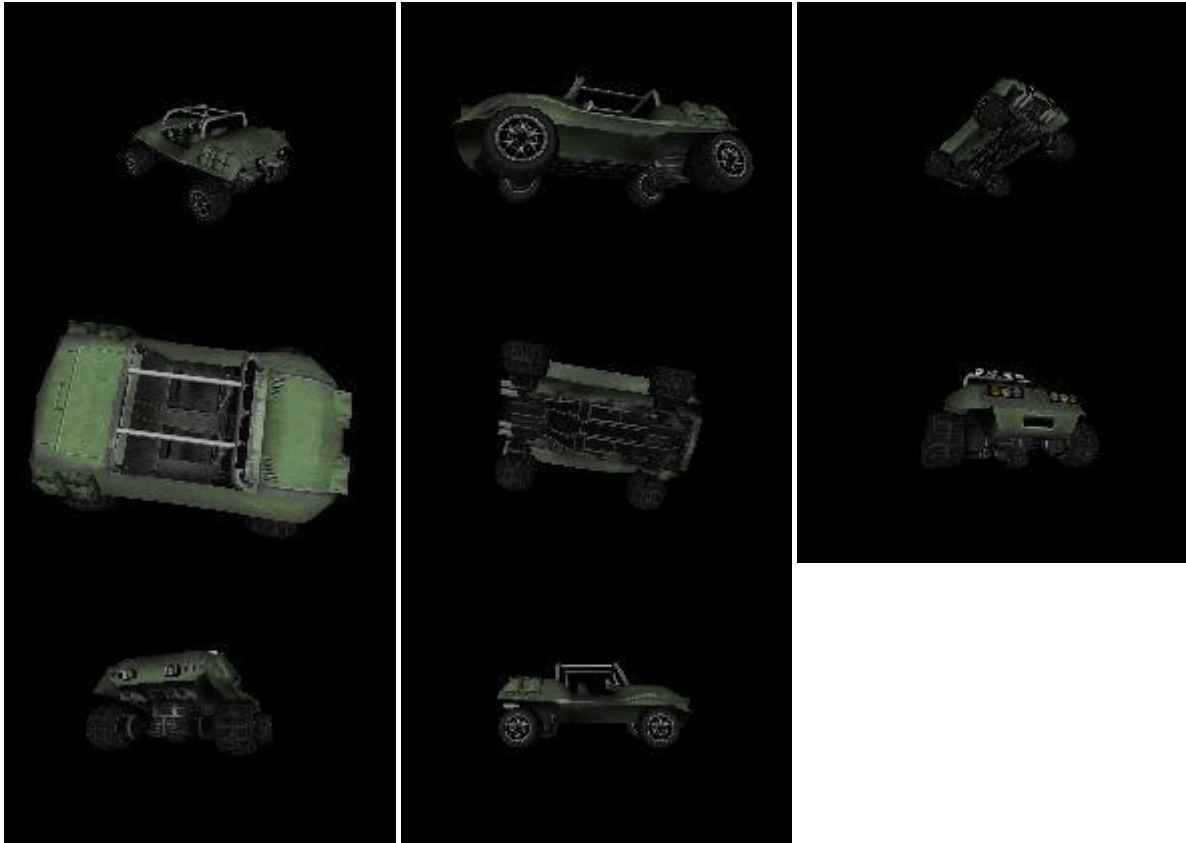


Fig. 26 – Les différentes vues apprises de l'objet Jeep

Et voici la vue qui a dû être reconnue par notre réseau :



Fig. 27 – Vue reconnue par le réseau de neurones  
Cette vue est donc globalement assez différente des vues mémorisées par notre réseau de neurones.

Voici les résultats des tests que nous avons obtenus en sortie de notre réseau de neurones. A partir de ces résultats, nous considérons que l'objet reconnu a le résultat le plus élevé.

Libellé	Jeep	Crayon	Cube jaune	Cube vert	Mitraillette
ID	0	1	2	3	4
Résultat 0	0.941123	0.00267197	0.0117666	0.0145518	0.00888625
Résultat 1	0.00026655	0.974981	0.00536585	0.0122349	0.019411
Résultat 2	0.0458933	0.0144926	0.972907	0.0148986	0.0016441
Résultat 3	0.0264871	0.0147598	0.0108101	0.974464	0.00223631
Résultat 4	0.0411907	0.00984677	0.0110712	0.0151512	0.989394

Dans ce tableau, pour la vue supplémentaire de la jeep, le résultat est de 0.94 pour « résultat 0 », 0 étant l'ID de la Jeep. Celle-ci est donc bien reconnue. On peut remarquer qu'il en est de même pour les objets Cube Vert et Cube jaune, ce qui prouve que notre technique reconnaît bien les différences de couleur.

Par la suite, il nous a fallu étendre nos résultats. Pour cela, nous avons réalisé nos tests pour un plus grand nombre d'objets. Etant donné le nombre d'entrées de notre réseau de neurones il n'était pas possible sur nos machines d'évaluer un trop grand nombre d'objets car notre prototype fit preuve d'une très grande consommation en mémoire vive. Nous avons donc poursuivi nos tests sur une base de vingt, puis de dix objets, leurs résultats en termes de précision furent globalement aussi bons que ceux sur cinq objets.

De ces tests, nous avons pu, en plus des résultats en termes de précision, récupérer des résultats en termes de performance. Pour plus de facilité de lecture, ces résultats ont été représentés sous forme de courbes. La première courbe représente la variation du temps d'apprentissage en fonction du nombre d'objets, et la seconde, le temps de détection par objet.

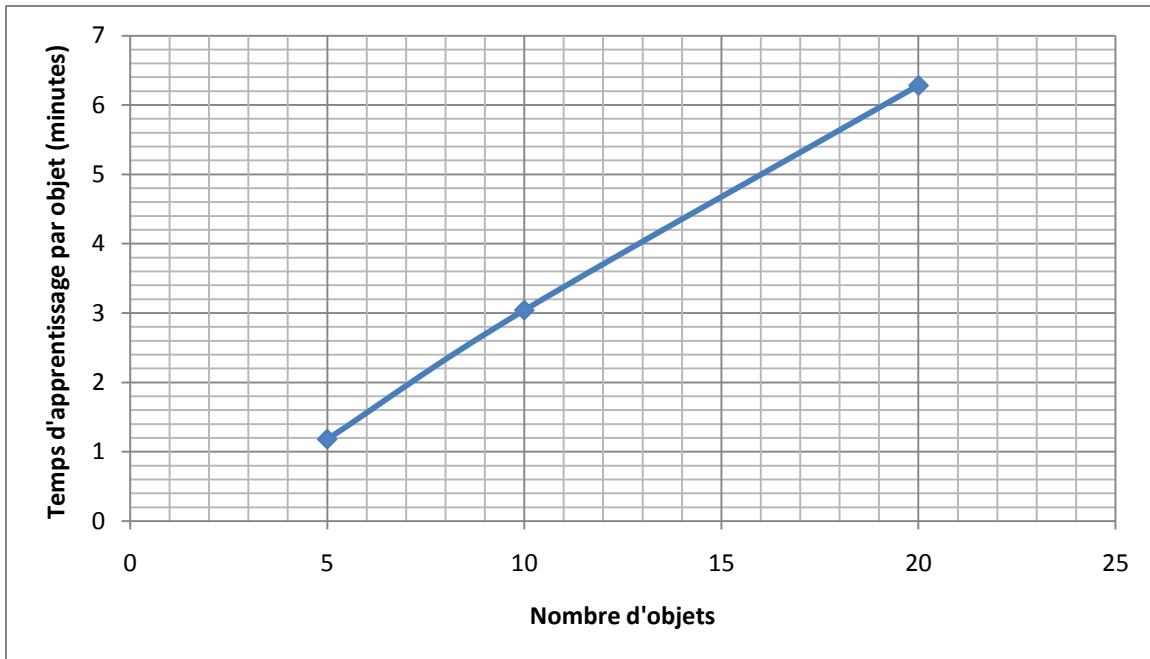


Fig. 28 – Courbe 1 : Temps d'apprentissage en fonction du nombre d'objets.

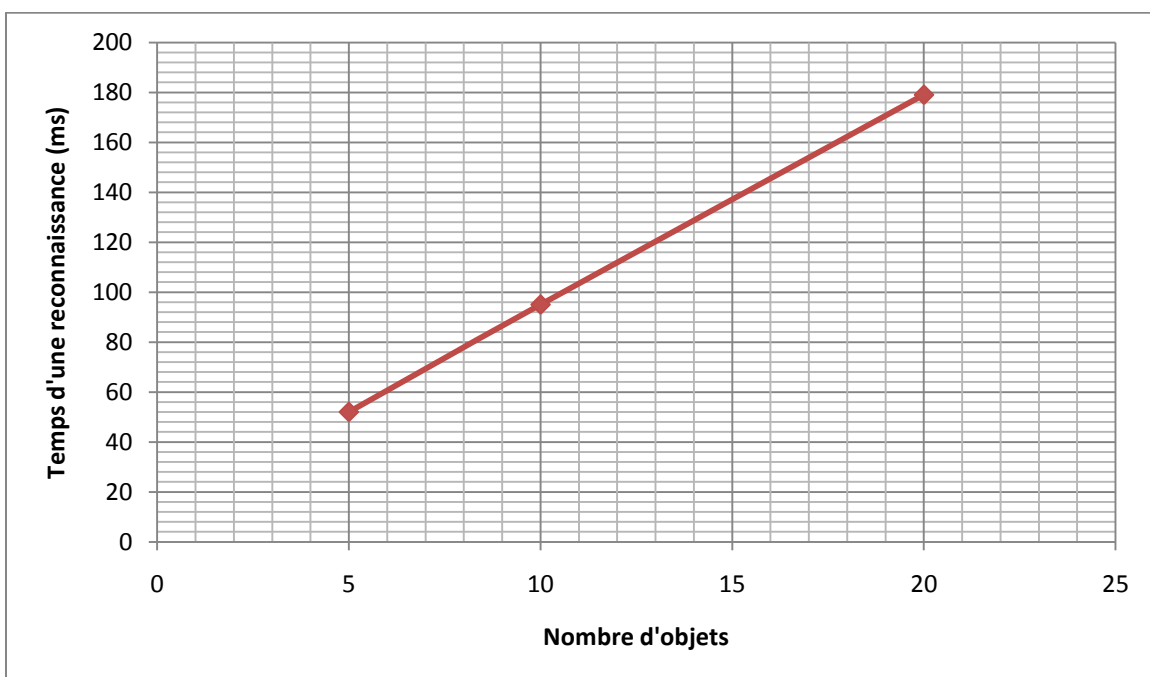


Fig. 29 – Courbe 2 : Temps de reconnaissance en fonction du nombre d'objets.

Comme on peut le voir sur ces deux courbes, le temps de reconnaissance et celui d'apprentissage semblent être des fonctions linéaires. Il est donc possible d'estimer le temps d'apprentissage et de reconnaissance pour un plus grand nombre d'objets.

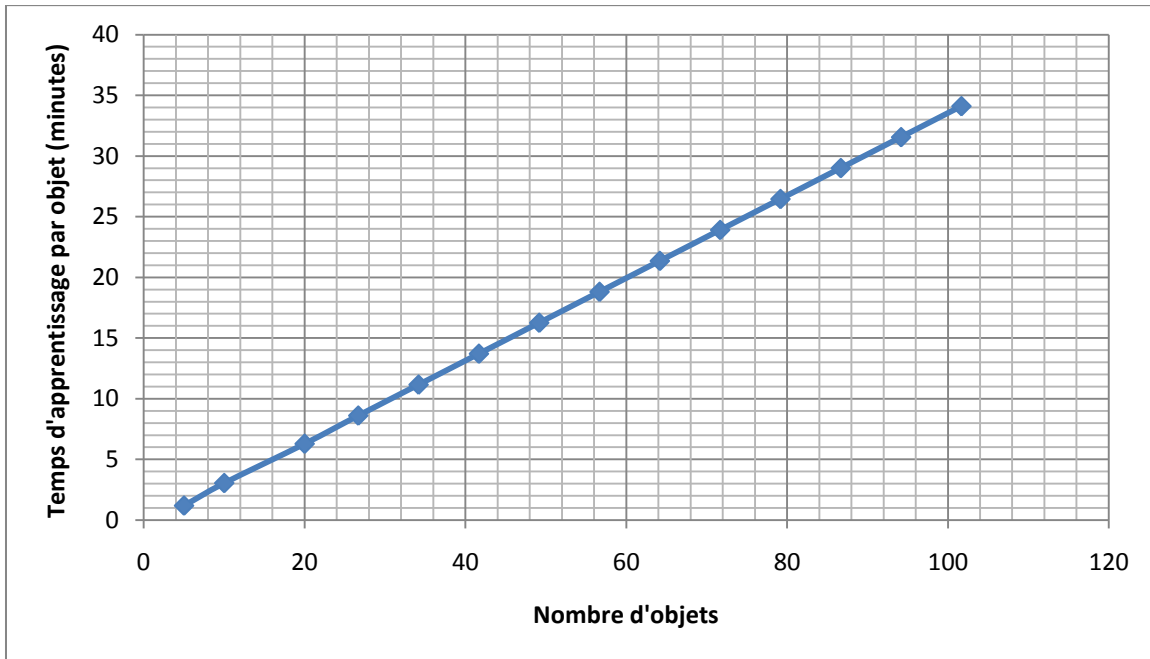


Fig. 30 – Courbe 3 : Temps d'apprentissage étendu.

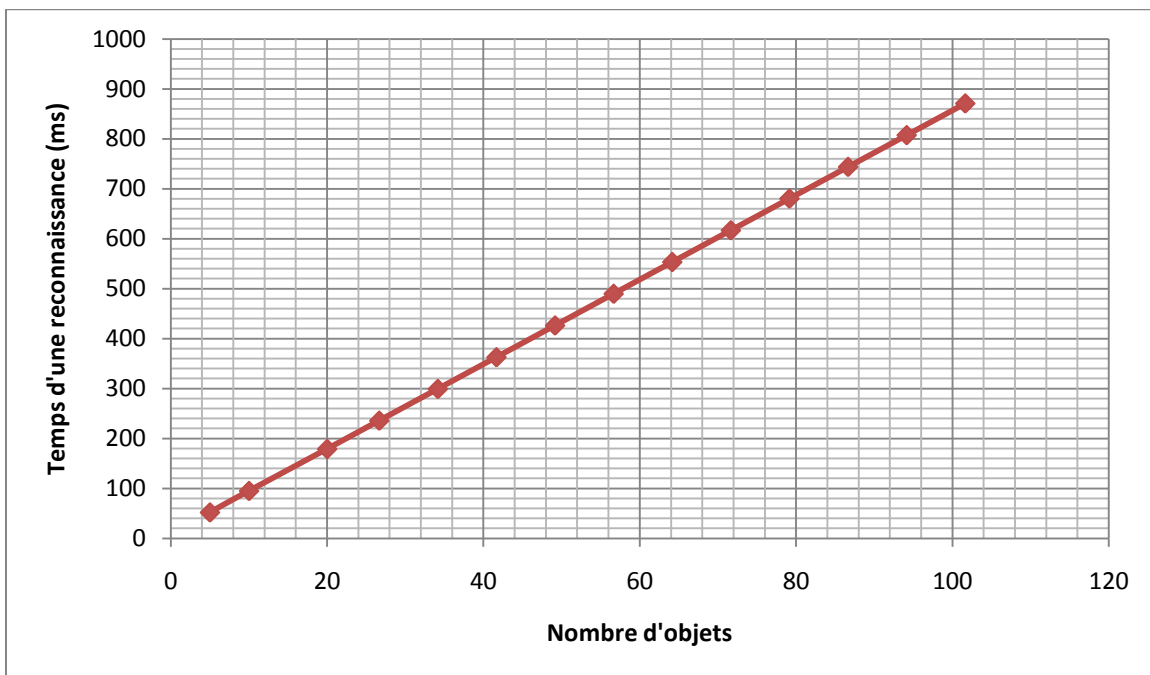


Fig. 31 – Courbe 4 : Temps de reconnaissance étendu

## Conclusion

Dans notre méthode, l'élément le plus significatif pour la reconnaissance est la couleur. C'est elle qui définit en premier si l'objet est reconnu ou non. Ainsi notre système ne se trompe dans sa reconnaissance que lorsque la couleur et la forme sont proches, alors que pour d'autres méthodes, il suffit que seule la forme soit proche pour qu'ils se trompent. Mais notre méthode possède un désavantage majeur : à chaque fois que l'on veut modifier la base d'objets, il est nécessaire de recommencer l'apprentissage de zéro.

Cela dit, on peut envisager plusieurs pistes pour améliorer notre système et le rendre plus performant et plus précis. Pour la précision, il est possible de prendre en compte la taille de l'objet ainsi que son opacité pour obtenir des informations plus détaillées sur celui-ci. En ce qui concerne les performances, celles-ci sont liées directement aux nombres de critères pris en compte par notre réseau de neurones. Pour réduire ces critères, il est possible d'utiliser une palette de couleurs et d'intégrer une méthode PCA (Principal Component Analysis).

## Annexes

### Résultat pour 20 objets

Libellé	Jeep	Pencil	Cube Jaune	Cube Vert	Mitraillette
ID	0	1	2	3	4
Résultat 0	<b>0.991902</b>	5.29E-06	1.51E-04	0.0017574	2.03E-04
Résultat 1	2.51E-07	<b>0.887757</b>	6.80E-06	1.75E-05	0.00130719
Résultat 2	0.00642977	0.000138219	<b>0.757263</b>	0.00180088	2.76E-05
Résultat 3	0.000150974	8.51E-05	0.0046547	<b>0.819401</b>	4.95E-06
Résultat 4	1.05E-02	4.54E-03	3.45E-05	2.68E-06	<b>0.699363</b>
Résultat 5	2.46E-05	1.52E-06	3.47E-05	2.62E-05	9.88E-09
Résultat 6	0.0144138	0.0229464	0.00135692	0.00173404	9.76E-02
Résultat 7	1.20E-06	3.71E-06	0.00451863	4.07E-04	0.000645137
Résultat 8	1.50E-02	2.57E-07	2.53E-04	6.76E-05	3.24E-07
Résultat 9	0.00256072	0.0532467	0.0110921	0.000537693	0.0127684
Résultat 10	9.12E-05	0.0230627	7.94E-03	1.14E-03	0.00328166
Résultat 11	0.00106229	0.000151667	0.144243	0.229999	6.55E-02
Résultat 12	0.00414258	1.79E-03	4.00E-05	0.00164442	7.22E-06
Résultat 13	0.00111003	7.45E-05	2.01E-04	0.00123618	2.45E-03
Résultat 14	2.53E-03	1.10E-02	4.59E-03	0.00241322	1.81E-07
Résultat 15	5.08E-05	0.0361856	3.29E-07	9.00E-06	2.25E-02
Résultat 16	2.08E-05	1.51E-06	6.61E-06	6.89E-06	0.00182075
Résultat 17	0.0131838	1.04E-07	6.68E-05	0.0466052	0.000184234
Résultat 18	1.19E-08	0.000448755	7.02E-06	2.10E-03	1.08E-09
Résultat 19	1.70E-04	0.000909636	1.99E-03	1.32E-05	8.65E-05

Libellé	Vase Grec	Épée	Disquette	Nain	Stylo
ID	5	6	7	8	9
Résultat 0	0.000660204	0.000152094	0.00025061	6.89E-04	6.96E-05
Résultat 1	1.43E-05	0.00546049	1.78E-06	1.07E-06	0.00475811
Résultat 2	0.00256906	0.00098856	0.000126492	9.87E-04	4.53E-04
Résultat 3	2.08E-07	0.0006446	0.000149243	0.00104058	5.04E-05
Résultat 4	1.45E-03	2.80E-04	6.23E-07	4.36E-05	0.000578734
Résultat 5	0.0928796	6.83E-06	4.62E-04	3.59E-04	8.63E-06
Résultat 6	1.14E-05	<b>0.962407</b>	0.000110917	3.33E-03	7.00E-03
Résultat 7	0.000337638	4.49E-04	<b>0.530028</b>	0.00241889	8.15E-06
Résultat 8	0.0450381	3.65E-06	0.000828471	<b>0.672319</b>	2.31E-07
Résultat 9	0.000621074	0.00372244	8.60E-05	7.18E-05	<b>0.884066</b>
Résultat 10	0.00501592	4.70E-03	0.000824481	0.00201083	0.00142126
Résultat 11	0.000402676	4.04E-05	1.62E-04	1.51E-05	0.000932467
Résultat 12	9.35E-04	0.000124286	0.00840359	0.000268156	6.86E-04
Résultat 13	5.65E-05	9.61E-07	0.0779727	1.08E-04	6.10E-04
Résultat 14	0.000287326	0.000183369	9.66E-05	0.0427364	0.00108925
Résultat 15	5.82E-06	0.0105363	1.55E-05	3.57E-05	0.00515029
Résultat 16	0.000172284	7.48E-05	2.79E-02	0.00227134	3.26E-07
Résultat 17	1.34E-05	8.63E-06	0.0991122	0.273737	1.34E-06
Résultat 18	1.44E-08	9.60E-07	1.15E-01	3.01E-04	2.27E-07
Résultat 19	<b>0.908168</b>	1.33E-05	8.52E-05	1.64E-05	0.000275322

Libellé	Coupe	Arbre	Roue	Table de Nuit	Fauteil
ID	10	11	12	13	14
Résultat 0	4.12E-08	2.18E-04	2.85E-04	0.000345926	1.80E-05
Résultat 1	0.000252409	1.01E-05	4.37E-06	3.10E-06	1.67E-03
Résultat 2	1.16E-02	0.0179835	7.33E-06	6.12E-05	1.28E-04
Résultat 3	4.11E-05	6.28E-02	0.000308809	0.00184823	0.000767961
Résultat 4	0.0186105	3.99E-05	3.75E-07	1.31E-05	5.07E-03
Résultat 5	0.000393036	8.37E-06	1.10E-04	9.29E-05	1.28E-04
Résultat 6	1.04E-02	0.00136958	1.29E-04	8.39E-04	5.47E-03
Résultat 7	0.000135545	2.30E-03	7.27E-03	0.00143446	7.87E-09
Résultat 8	9.17E-08	2.85E-06	0.000591413	0.000307514	3.72E-06
Résultat 9	0.00180238	0.0144038	0.000481147	5.50E-04	5.37E-04
Résultat 10	<b>0.992374</b>	0.000393958	5.83E-05	0.000218122	0.0214537
Résultat 11	1.78E-03	<b>0.89134</b>	3.01E-05	1.37E-03	1.44E-04
Résultat 12	6.08E-05	2.52E-05	<b>0.0437154</b>	5.86E-05	8.95E-06
Résultat 13	3.42E-06	9.77E-04	0.00184396	<b>0.963872</b>	0.000599799
Résultat 14	0.00370768	0.000417917	0.0330551	9.89E-05	<b>0.995221</b>
Résultat 15	3.99E-04	2.28E-06	7.87E-05	6.21E-05	9.80E-05
Résultat 16	4.63E-05	7.01E-06	8.59E-05	1.46E-04	1.05E-06
Résultat 17	3.13E-09	3.84E-03	1.60E-01	0.00393266	5.67E-08
Résultat 18	3.45E-08	5.27E-05	1.08E-01	2.83E-01	7.15E-03
Résultat 19	0.196236	1.27E-04	9.09E-06	2.45E-05	2.53E-05

Libellé	Clé	Lampe	Tasse	Tabouret	Commode
ID	15	16	17	18	19
Résultat 0	2.19E-05	7.79E-04	0.00104776	7.22E-05	0.000658153
Résultat 1	0.0548639	1.15E-05	1.95E-06	2.52E-05	1.43E-05
Résultat 2	9.92E-07	8.23E-06	1.73E-05	8.04E-05	2.57E-03
Résultat 3	0.000161031	0.00130262	0.00175484	0.0296338	2.08E-07
Résultat 4	1.11E-02	1.20E-06	5.37E-06	4.68E-07	0.00144971
Résultat 5	3.91E-06	0.000450808	5.75E-04	0.000211782	0.0928749
Résultat 6	0.019163	7.84E-04	2.00E-05	2.09E-03	1.14E-05
Résultat 7	6.41E-07	0.0436963	0.00292279	0.00230074	0.000336773
Résultat 8	5.54E-07	0.00064801	6.60E-03	1.10E-04	0.0449936
Résultat 9	0.021857	1.63E-05	2.06E-05	3.06E-05	0.000620871
Résultat 10	5.10E-03	8.13E-05	0.000243958	0.0001975	5.03E-03
Résultat 11	2.44E-05	0.000138974	4.41E-04	0.00264323	0.000402492
Résultat 12	2.21E-03	0.00459936	0.000500245	7.25E-04	0.000932829
Résultat 13	0.00151221	1.60E-02	0.00249807	0.0512019	5.65E-05
Résultat 14	2.01E-04	0.000142245	0.00285763	1.68E-04	2.87E-04
Résultat 15	<b>0.950688</b>	2.28E-05	9.71E-06	6.96E-06	5.82E-06
Résultat 16	2.42E-05	<b>0.095149</b>	4.23E-04	1.09E-02	0.000172167
Résultat 17	8.81E-05	0.00931269	<b>0.0876605</b>	0.000280291	1.34E-05
Résultat 18	6.31E-06	0.941292	0.0268904	<b>0.990466</b>	1.44E-08
Résultat 19	1.30E-06	1.44E-05	9.38E-06	6.78E-05	<b>0.908141</b>

## Bibliographie

### Référence

**[wikip1]** Wikipedia (31/05/2010)

[http://fr.wikipedia.org/wiki/Reconnaissance\\_de\\_formes](http://fr.wikipedia.org/wiki/Reconnaissance_de_formes)

**[wikip2]** Wikipedia (01/06/2010)

[http://fr.wikipedia.org/wiki/Réseau\\_de\\_neurones\\_artificiel](http://fr.wikipedia.org/wiki/Réseau_de_neurones_artificiel)

**[wikm1]** Wikimedia (01/06/2010)

[http://commons.wikimedia.org/wiki/File:Neurone\\_-\\_commenté.svg](http://commons.wikimedia.org/wiki/File:Neurone_-_commenté.svg)

**[sit1]** Isabelle Bellin (31/05/2010)

[http://interstices.info/jcms/c\\_5952/histoire-du-traitement-dimages?part=2](http://interstices.info/jcms/c_5952/histoire-du-traitement-dimages?part=2)

**[publi1]** Bruno Jedynak et François Fleuret, *Reconnaissance d'objets 3D à l'aide d'arbres de classification\**, 09/04/1996, 9 pages, 2-3-4-5

**[publi2]** M. Chakchouk et F. Chaieb, *Description invariante d'objets 3D représentés par des images géométriques*, CORESA Lille, France, 26/05/2004, 4 pages, 1-2-3

**[publi3]** Noël Trujillo, Frédéric Chausse et Roland Chapuis, *Reconnaissance d'objets par vision focalisée*, LASMEA - UMR 6602 UBP/CNRS, 02/05/2005, 6 pages, 2-3-4-5

**[publi4]** Fabrice El Gohary et Charles Letailleur, *Indexation d'objets 3D : approche 2D-3D pour des requêtes à partir d'images*, 27/05/2007, 7 pages, 1-2

**[publi5]** NGHIEM Anh Tuan, *Reconnaissance d'écriture manuscrite*, 15/07/2005, 33 pages, 22-23-24-25

**[publi6]** Guillaume Ryder, *Reconnaissance d'objets Approche structurelle*, Laboratoire d'Informatique en Images et Systèmes d'information, 05/09/2005, 21 pages, 1-2

**[publi7]** Jeff Hawkins et Dileep George, *Hierarchical Temporal Memory - Concepts, Theory, and Terminology*, Numenta Inc, 17/05/2006, 20 pages, 1-2-3-4

### Supplément d'information

**[1]** Mikaël Bourges-Sévenier, Patrik Horain et Françoise Prêteux, *Recalage d'un modèle 3D générique sur une séquence d'images 2D*, Mars 1997.

**[2]** Thierry Géraud, Isabelle Bloch et Henri Maître, *Reconnaissance de structures cérébrales à l'aide d'un atlas et par fusion d'informations structurelles floues*, Février 2000.

**[3]** Cédric Dumas, Patricia Plénacoste et Samuel Degrande, *Les indices dynamiques dans les interfaces 3D - au delà des ombres portées -*, Décembre 2000.

- [4] TRAN Thi Thanh Hai et Augustin LUX, *Extraction de Caractéristiques Locales: Crêtes et Pics*, Février 2003.
- [5] Mourad ZRIBI, Faouzi GHORBEL, Raymond MOCHÉ, Alain HILLION et Valérie BURDIN, *Une famille d'invariants stables pour l'analyse des objets 3D à niveaux de gris*, Février 2004.
- [6] Svetlana Lazebnik, Cordelia Schmid et Jean Ponce, *Semi-Local Affine Parts for Object Recognition*, Juillet 2004.
- [7] Abdelmalek TOUMI, Brigitte HOELTZENER et Ali KHENCHAF, *Multi-level in radar automatic target recognition*, Mars 2005.
- [8] Dipti Deodhare, NNR Ranga Suri et R. Amit, *Preprocessing and Image Enhancement Algorithms for a Form-based Intelligent Character Recognition System*, Mai 2005.
- [9] Fethi SMACH, Cédric LEMAITRE, Johel MITERAN et Jean-Paul GAUTHIER, *Reconnaissance d'objets 3D couleur à l'aide des descripteurs de Fourier Généralisés et des SVM*, Janvier 2006.
- [10] Abdelhalim BOUTARFA, *Reconnaissance des formes 3D par approche neuronale associant la transformée de Hough en Robotique mobile Application a la production*, Mai 2006.
- [11] Faisal Tehseen Shah et Kamran Yousaf, *Handwritten Digit Recognition Using Image Processing and Neural Networks*, Juillet 2007.
- [12] Devi Parikh, C. Lawrence Zitnick et Tsuhan Chen, *From Appearance to Context-Based Recognition: Dense Labeling in Small Images*, Avril 2008.
- [13] IA 2002-2003 - © C. Pellegrini, *Neurones artificiels: modèles et réseaux AIMA chap. 19 (19.1-19.5)*, Avril 2003
- [14] IA 2002-2003 - © C. Pellegrini, *Applications des réseaux neuronaux*, Mai 2003
- [15] Fabrice Rossi, *Réseaux de neurones : le perceptron multi-couches - cours à L'Université Paris-IX Dauphine*, Avril 2003
- [16] Claude TOUZET, *Les Réseaux de neurones artificiels Introduction au connexionnisme Cours, Exercices et Travaux pratiques*, Juillet 1992
- [17] Marc Parizeau, *Réseaux de neurones GIF-21140 et GIF-64326*, Aout 2004