

# Détection/Reconnaissance d'objets sur rétines artificielles programmables mixtes synchrones/asynchrones

*Rapport de stage de DEA  
Intelligence Artificielle et Reconnaissance des Formes et  
Applications*

Responsable de stage : Antoine MANZANERA (ENSTA)

Julien RICHEFEU  
jiceher@free.fr  
septembre 2002

## **Résumé**

Nous présentons dans ce document un travail de recherche visant à élaborer des algorithmes de reconnaissance de formes sur rétines artificielles numériques. Le but de ce travail, qui s'appuie sur les travaux menés par le laboratoire d'électronique et informatique de l'ENSTA, est de spécifier des techniques efficaces de détection d'objets tout en respectant l'aspect massivement parallèle des rétines. Les résultats obtenus auront comme cadre applicatif l'exemple de la lecture automatique de plaques d'immatriculation.

## Table des matières

<b>Article</b>	<b>3</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Présentation du sujet . . . . .	3
1.2 Intérêt des rétines artificielles en reconnaissance des formes . . . .	3
1.3 Démarche adoptée . . . . .	4
<b>2 Détection d'objets sur rétine artificielle</b>	<b>6</b>
2.1 Hypothèses de travail . . . . .	6
2.2 Localisation . . . . .	6
2.3 Etude Granulométrique . . . . .	9
<b>3 Reconnaissance de caractères</b>	<b>13</b>
3.1 Choix de l'outil de discrimination . . . . .	13
3.2 Mise en place de la reconnaissance . . . . .	13
<b>4 Résultats obtenus et interprétation</b>	<b>16</b>
4.1 Un exemple concret . . . . .	16
4.2 Cadre général . . . . .	19
<b>5 Conclusion</b>	<b>20</b>
<b>Références</b>	<b>21</b>
<b>Annexes</b>	<b>22</b>
<b>A Définitions</b>	<b>22</b>
A.1 Rétine artificielle . . . . .	22
A.2 Filtrage linéaire . . . . .	26
A.3 Morphologie mathématique . . . . .	27
A.4 Opérateurs géodésiques . . . . .	31
A.5 Histogramme . . . . .	32
A.6 Réseaux de neurones . . . . .	33
<b>B Applications développées pour le projet</b>	<b>34</b>

## Table des figures

1	Schéma général d'un système de reconnaissance des formes [Belaïd 1992].	3
2	Schéma général du processus de l'application. . . . .	5
3	Processus de localisation de la plaque. . . . .	7
4	Granulométrie et anti-granulométrie. . . . .	9
5	Analyse granulométrique de la plaque. . . . .	10
6	Seuillage par hystérésis. . . . .	11
7	Architecture du Perceptron Multicouche. . . . .	14
8	Comparaison entre mlp avec (bleu) et sans (rouge) couche cachées.	15
9	Première phase du traitement sur <i>f_360AEC95.bmp</i> . . . . .	16
10	Analyse granulométrique de <i>f_360AEC95.bmp</i> . . . . .	17
11	Histogramme de la fonction de répartition (bleu) et sa dérivée (rouge) pour <i>f_360AEC95.bmp</i> . . . . .	18
12	Résultat du traitement sur <i>f_360AEC95.bmp</i> . . . . .	18
13	Résultat du traitement sur diverses images. . . . .	19
14	Une dilatation binaire sur la rétine décomposée en opérations élémentaires [Manzanera 2000]. . . . .	22
15	principe d'acquisition en niveau de gris par multiseuillage [Manzanera 2000].	23
16	Principe de traitement en niveaux de gris par combinaison des traitements binaires effectués sur chaque seuil[Manzanera 2000]. .	24
17	Logiciel TraitLib . . . . .	34
18	Logiciel Granu . . . . .	35

# 1 Introduction

## 1.1 Présentation du sujet

Ce stage est un projet inter-disciplinaire; il fait intervenir trois champs de recherche : *la reconnaissance des formes, l'apprentissage numérique et le développement de logiciels*. Il a pour objectif de mettre en place des primitives de détection et de reconnaissance d'objets sur une **rétine artificielle**.

Les **rétines artificielles** programmables sur lesquelles travaille l'équipe d'accueil depuis une dizaine d'années sont, d'un point de vue informatique, des grilles 2-D de PE (processeurs élémentaires) à raison d'un PE par pixel. Grâce à l'introduction actuelle de primitives de calcul asynchrones dans le PE, certaines limitations architecturales fortes de ces grilles tombent, leur offrant la possibilité de supporter efficacement des tâches de vision qualifiées de moyen niveau et de contribuer vraiment à la détection/reconnaissance/identification d'objets. L'objectif du stage est d'exploiter cette nouvelle donne dans un esprit opportuniste. Il s'agit, en exploitant au mieux le nouveau modèle de calcul rétinien, de parvenir à un apprentissage automatique de forme ou de texture. La détection/lecture de plaques d'immatriculation est le cadre applicatif envisagé.

## 1.2 Intérêt des rétines artificielles en reconnaissance des formes

Dans un système de vision à base de **rétine artificielle**, les images sont traitées à l'endroit et à l'instant de leur acquisition. La rétine programmable est à la fois un capteur d'images et une machine **SIMD**, avec un processeur numérique logé dans chaque pixel. [Manzanera 2000]

Dans un processus classique de **reconnaissance des formes** une première étape consiste à coder les images provenant du monde physique puis à effectuer un prétraitement pour éliminer le bruit dû aux conditions d'acquisition puis à effectuer l'analyse de l'image ainsi obtenue avant de passer à la phase de reconnaissance à proprement parler.

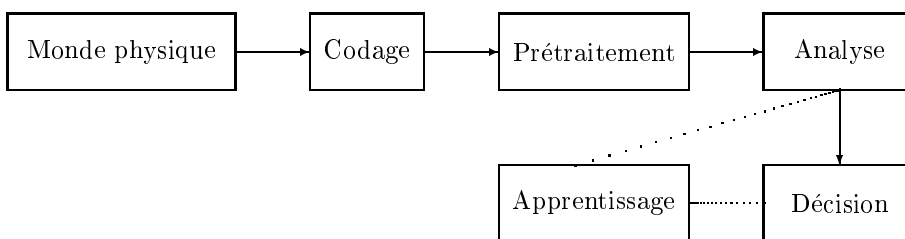


FIG. 1 – Schéma général d'un système de reconnaissance des formes [Belaïd 1992].

Dans les rétines artificielles, les trois premières étapes de la reconnaissance des formes peuvent se dérouler directement sur l'appareillage de capture d'image. Le traitement à base de **rétilne artificielle** gagne en temps et en puissance de calcul pour ce qui concerne le traitement de bas niveau et de moyen niveau ; du moins jusqu'à un certain point défini par les limites physiques de la rétine en terme de taille mémoire (limitée aujourd'hui à quelques bits par pixels).

Cependant, nous espérons nous affranchir de ces limites avec la venue de la dernière génération de rétines supportant une plus grande mémoire embarquée et surtout une architecture mixte synchrone/asynchrone.

### 1.3 Démarche adoptée

Aucun travail de **reconnaissance de formes** n'a été jusqu'alors implanté sur une **rétilne artificielle** toutefois de nombreux algorithmes peuvent être facilement portés sur une architecture massivement parallèle. C'est le cas de la **morphologie mathématique** et des **opérateurs géodésiques** par [Manzanera 2000] et [Ducourthial 1999] et l'équipe de la *maille associative* de l'Université Orsay Paris-Sud.

Nous nous sommes donc surtout inspirés dans un premier temps des travaux effectués sur des architectures classiques afin de les transposer au contexte rétinienn. A la lecture de ces divers travaux, il ressort plusieurs techniques récurrentes [Brugge 1998] [Barroso 1997]. Tout d'abord, la reconnaissance des plaques d'immatriculation s'effectue en deux étapes distinctes, une de localisation et de segmentation de la plaque dans l'image et l'autre de reconnaissance des caractères ainsi segmentés lors de la première étape.

La grande difficulté de ce projet est de rester proche du contexte rétinienn, c'est à dire d'implanter des algorithmes qui soient transposables sur une **rétilne artificielle** sans pouvoir réellement tester ceux-ci. En effet, nous disposons dans les laboratoires d'une rétine utilisée par Antoine MANZANERA pour ses travaux de recherche et notamment sa thèse [Manzanera 2000] mais le modèle de rétine sur lequel nous travaillons n'est à ce jour toujours pas opérationnel. Nous disposons cependant d'un simulateur et de toute l'expérience qu'ont accumulée Antoine MANZANERA et Thierry BERNARD dans leurs activités de recherche.

C'est pourquoi, tout au long de notre travail, nous avons préféré les opérations non linéaires à celles qui sont linéaires et les développements en parallèle aux développements séquentiels, le point critique étant la ressource mémoire.

Ici seule la première étape de localisation de la plaque peut être utilisée dans le cadre de notre travail de recherche et après avoir effectué une première tentative infructueuse il s'est avéré que c'est l'**analyse granulométrique** qui semble le plus appropriée à notre problème.

Pour ce qui est de la phase de reconnaissance, nous avons décidé de mettre en place un **réseau de neurones** en sortie de la rétine en s'inspirant des travaux menés par l'équipe de [Sackinger 1992] sur la reconnaissance de caractères. Cette équipe avait en effet utilisé le processeur de **réseau de neurones ANNA** en émulant une **architecture pyramidale** afin d'accélérer le processus de reconnaissance. Ces recherches ont été une source d'inspiration importante pour notre projet.

La figure 2 illustre le processus employé par le logiciel pour traiter les images contenant des plaques d'immatriculation. Nous allons par la suite développer tour à tour ces différentes étapes et expliquer leurs utilités dans le processus.

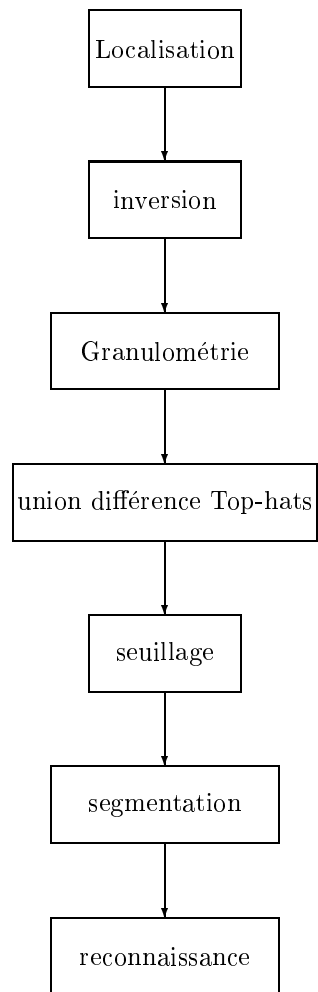


FIG. 2 – Schéma général du processus de l'application.

## 2 Détection d'objets sur rétine artificielle

### 2.1 Hypothèses de travail

Afin de limiter le domaine de recherche du projet, nous avons posé certaines hypothèses qui nous ont semblées raisonnables étant donné le cadre applicatif des résultats. En effet, le dispositif final serait par exemple un dispositif comprenant une **rétine artificielle** disposée sur un rail de sécurité le long d'une route ou d'une autoroute. Ainsi les images traitées par la rétine sont des plaques dont l'angle de prise de vue et la distance sont à peu près toujours identiques.

Nous pouvons donc poser comme hypothèse que les caractères traités sont normalisés en terme de taille et d'angles de visée. Cela implique que les images que nous avons en entrée du **réseau de neurones** sont toutes de même dimension mais que les images traitées par le processus de localisation et de segmentation ont des tailles de caractères de tout type. De même nous nous sommes fixés comme objectif la reconnaissance de chiffres et non de lettres et de chiffres vu la base d'images restreinte dont nous disposons [Brekke 2002]. Cependant, étendre l'application aux lettres ne pose en théorie pas de difficultés et ne modifie que l'architecture de **réseau de neurones** et bien entendu la vitesse de calcul de celui-ci.

### 2.2 Localisation

Ce premier travail à effectuer directement sur l'image originale peut en fait se découper en une suite de petits procédés simples qui permettent de localiser l'emplacement verticale de la plaque dans l'image.

Comme l'illustre la figure 3, le processus de localisation de la plaque dans l'image s'effectue à l'aide de filtres de SOBEL et de gradients horizontaux. En pratique sur la rétine, il est très difficile d'implanter une opération de convolution comme le filtre de SOBEL, par contre on peut aisément implanter un gradient comme cela est rapporté dans [Manzanera 2000]. Cependant il s'agit là de **gradient morphologique** et non de gradient issue d'une opération de filtrage par **convolution**. Sur l'architecture de **rétine artificielle** actuelle, il est difficile d'implanter une opération de convolution et donc un gradient. on utilise donc le **gradient morphologique** qui n'a pas les mêmes propriétés notamment en terme d'élimination du bruit. voir Annexe A.1 et A.3

Nous utilisons sur la rétine un algorithme pour ces **opérations de convolution** de la forme : voir Annexe A.1 et A.2

```

convolution(masque_de_convolution, image) : pvar
pvar resultat ← 0;
mgraphe mg;
pour i ← 0 à b-1
    mg ← masque_de_convolutioni;
    resultat ← resultat + 2i Plus_association_elementaire(mg, image);
fin pour
retourner resultat;

```

Nous utilisons ici les masque suivants :

$$SobelY = \begin{vmatrix} 1 \\ 2 \\ 1 \end{vmatrix} \quad GradientEW = \begin{vmatrix} -1 & 0 & 1 \end{vmatrix}$$

En fait l'idée d'utiliser des gradients provient de [Adorni 1998] dans le projet de reconnaissance de plaques d'immatriculation par un automate cellulaire. Ce filtrage permet de détecter les importantes variations de contrastes horizontales. Or les plaques d'immatriculation sont composées de caractères clairs sur fond foncé ou inversement caractères foncés sur fond clair. Appliquer un gradient sur l'image originale après l'avoir lissée perpendiculairement au sens du gradient fait ressortir ce contraste. Pour le lissage, nous utilisons un filtre de SOBEL comme cela est fait assez classiquement comme par exemple dans [Brugge 1998].

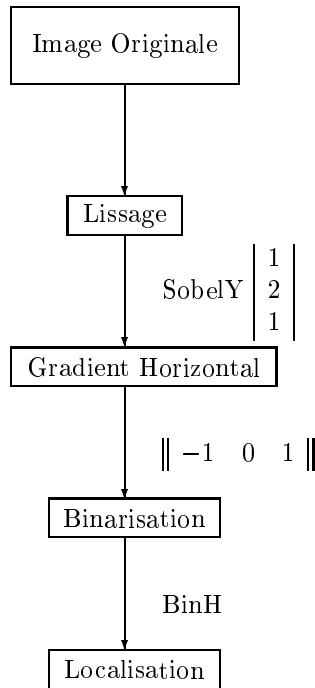


FIG. 3 – Processus de localisation de la plaque.

Les ressources mémoires de la rétine étant très limitées, nous ne pouvons nous permettre de conserver une image en niveaux de gris. C'est pour cette raison que nous effectuons un seuillage particulier pour obtenir une image binaire qui ne nous prend alors qu'un seul bit mémoire.

Afin d'optimiser le seuillage de l'image obtenue après la combinaison du filtrage de SOBEL et du gradient horizontal, nous utilisons la formule [1.1] introduite dans [Adorni 1998] et utilisée dans le traitement automatique de plaques d'immatriculation.



$$B(x, y) = \begin{cases} 0 & , \partial_x I > \overline{\partial_x I} + a([\partial_x I]_m - \overline{\partial_x I}) \\ 255 & \text{sinon} \end{cases} \quad [1.1]$$

- $\partial_x I$  : le gradient horizontal ;
- $\overline{\partial_x I}$  : la moyenne du gradient sur l'image ;
- $[\partial_x I]_m$  : le maximum du gradient sur l'image ;
- $a$  : coefficient de luminance (variation de contraste), arbitraire.

En pratique, nous obtenons une bonne binarisation pour des valeurs de  $a$  comprises entre  $-\frac{3}{10}$  et  $-\frac{1}{2}$ . Il est à noter que pour  $a=1$ , on obtient une image uniformément noire (tous les pixels à "0") et pour  $a=0$ , on obtient une binarisation où les pixels noirs sont ceux pour lesquels le niveau de gris est supérieur à celui du niveau moyen de l'image. Il est à noter que tous ces calculs sont globaux. Le maximum global de l'image intervient dans la formule, or celui-ci est un terme très sensible au bruit, une technique qui permettrait d'améliorer cette formule consisterait donc à s'affranchir de ce terme.

Pour réaliser un tel traitement sur une **rétine artificielle**, il nous faut calculer les valeurs moyennes et maximales des niveaux de gris des pixels de l'image puis de comparer ensuite la valeur en chaque pixel avec ces valeurs calculées. Si l'on considère que travailler sur une image en niveau de gris revient à travailler sur un nombre d'images binaires correspondant au nombre de niveau de gris de l'image, alors ce traitement est certes long mais réalisable en un temps raisonnable. Pour ce faire, on utiliserait une opération d'extraction globale de l'information comme le OU global seuil par seuil pour obtenir une approximation de la valeur maximale du gradient de l'image et de la moyenne de ce gradient.

Afin de détecter la plaque, nous comptons le nombre de pixels allumés (niveaux de gris=0) dans  $binH$  par ligne. Nous ne gardons que les lignes pour lesquelles cette somme est supérieure à un certain seuil  $\beta 1$ , puis on cherche la plus grande suite de lignes de pixels allumés. Nous obtenons ainsi l'emplacement vertical de la plaque dans l'image. Le coefficient  $\beta 1$  joue ici un rôle assez important puisqu'il permettra de resserrer plus ou moins la largeur de la fenêtre localisée sur la plaque.

Cette technique est inspirée des **cross-sections** expliqués en [Barroso 1997] et adaptée au contexte "*rétinien*". Dans l'article original, l'équipe de chercheurs projetait horizontalement et verticalement la somme des pixels de chaque ligne et de chaque colonne et en déduisait l'emplacement de la plaque d'immatriculation puis de chaque caractère de la plaque. Ici, vu les capacités mémoires de la rétine, nous ne travaillons que sur les lignes et nous ne cherchons que la plaque, la segmentation se fera dans une étape ultérieure. La méthode des **cross-sections** est similaire du point de vue de sa réalisation sur une **rétine artificielle** à la technique du OU global par ligne. Pour ce donner une idée d'une manière de l'implémenter sur une rétine, nous donnons en Annexe A.5 un exemple d'un algorithme effectuant un histogramme sur une machine semblable. Un **cross-section** est similaire et effectue un histogramme par ligne ou par colonne. [Manzanera 2000]

### 2.3 Etude Granulométrique

Le cadre de recherche s'est considérablement réduit puisque nous connaissons l'emplacement de la plaque dans l'image et que nous nous focalisons maintenant sur l'extraction des caractères afin ensuite de passer à la phase de reconnaissance. Ainsi nous allons effectuer une **analyse granulométrique** comme cela est expliqué dans le cours de [Manzanera 2002] sur la plaque d'immatriculation extraite de l'image originale.

La granulométrie est l'étude de la taille des objets. En physique, elle repose principalement sur le tamisage (grains), et l'injection de mercure (milieux poreux). Or, les grains refusés par le tamis de taille  $\lambda$ , qui sont une partie de la population initiale, sont refusés aussi par tout nouveau tamisage de taille  $\mu \leq \lambda$ . Ainsi l'**analyse granulométrique** consiste à appliquer des ouvertures et des fermetures par des éléments structurants, des boules de tailles croissantes et de construire les "*top-hat*" ou "*top-hat conjugués*" correspondants. Pour une boule donnée, nous faisons disparaître certains détails de l'image. [Serra 2000]



FIG. 4 – Granulométrie et anti-granulométrie.

On note :

- $B_\lambda$  : la boule de taille  $\lambda$  ;
- $\gamma_{B_\lambda}(X)$  : l'ouverture d'élément structurant la boule de taille  $\lambda$  ;
- $\varphi_{B_\lambda}(X)$  : la fermeture d'élément structurant la boule de taille  $\lambda$  ;
- $\tau_{B_\lambda}(X)$  : le top-hat d'élément structurant la boule de taille  $\lambda$ .

voir Annexe A.3.

Nous appliquons donc cette suite d'ouverture, de fermeture et de top-hat sur l'image originale et lorsque la taille de la boule correspond à peu près à l'empattement des caractères de la plaque d'immatriculation, nous obtenons une image en niveaux de gris ne comportant que les détails qui nous intéressent. Afin de déterminer automatiquement quelles sont les tailles des boules qu'il faut retenir, nous pouvons tracer une fonction de distribution sur les images issues de la granulométrie  $(\gamma_\lambda)_{\lambda \in \mathcal{N}}$ . cette fonction peut être la somme des niveaux de gris des pixels de l'image ou la moyenne de ces niveaux de gris, peu importe à condition qu'elle rende compte de l'évolution de la quantité d'information contenue dans l'image. C'est la dérivée de cette fonction qui nous renseigne par la présence de maxima locaux.

Avant même de commencer les opérations d'ouvertures et de fermetures il faut savoir si la plaque de l'image que nous traitons est claire sur fond foncé ou inversement car c'est cette information qui nous permet de choisir entre la granulométrie (respectivement traitement des ouvertures) et l'anti-granulométrie (resp. fermetures). Pour ce faire il nous suffit de compter le nombre de pixels clairs (dont le niveau de gris est supérieur à un certain seuil, de valeur arbitraire) et de prendre la décision adéquate.

Si nous effectuons une différence entre deux des top-hats obtenus, nous n'obtenons plus que les détails qui disparaissent de l'image originale à chaque étape de l'**analyse granulométrique** (resp. **anti-granulométrique**). Afin d'améliorer la qualité des résultats et parce qu'il est très vraisemblable que la taille de la boule optimale  $\lambda$  ne coïncide pas exactement avec une des boules discrètes utilisées, le résultat de notre étude granulométrique est la réunion (le **O**U logique) entre les deux images résultantes des différences des top-hats pour les boules de tailles  $[\lambda - 1, \lambda]$  et  $[\lambda, \lambda + 1]$ .

Dans la figure 5 nous ne mentionons que les opérations d'ouvertures mais les opérations de fermetures sont menées aussi en même temps, à chaque étape.

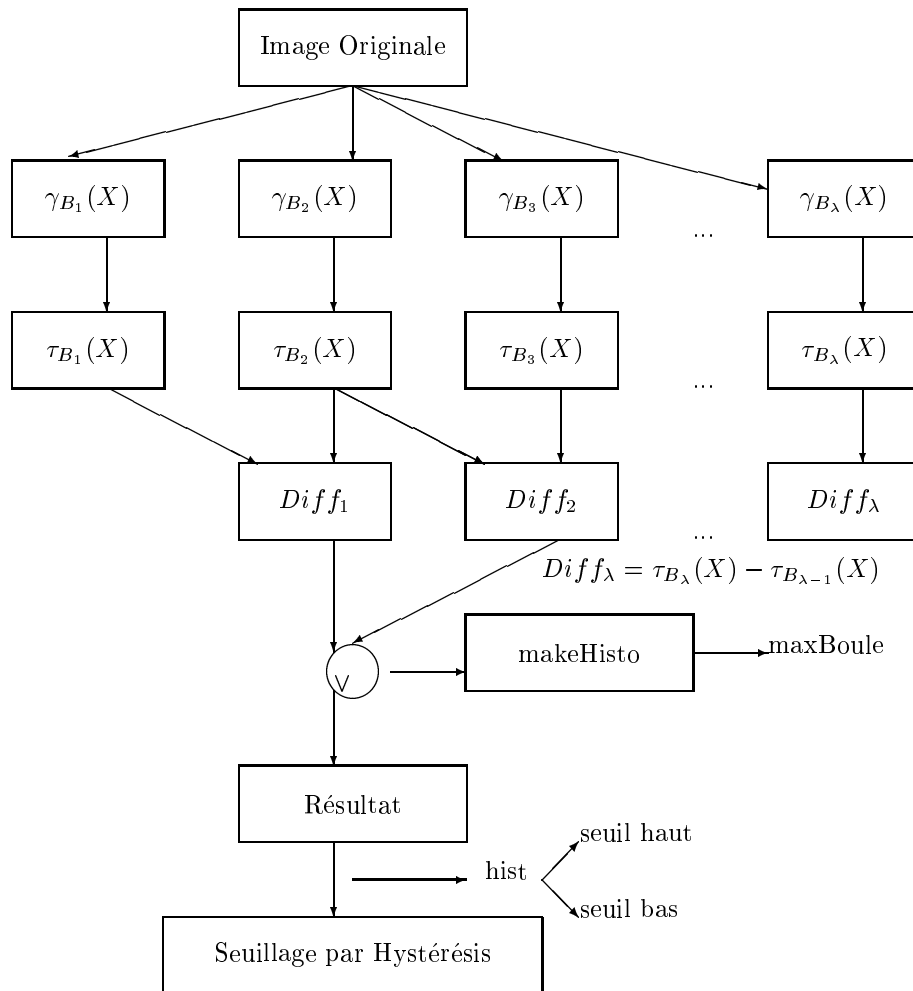


FIG. 5 – Analyse granulométrique de la plaque.

Les images issues de l'**analyse granulométrique** sont encore à ce stade en niveau de gris et nous ne souhaiterions obtenir qu'une image binaire moins coûteuse en terme de place mémoire dans la rétine. Il nous faut donc procéder à une binarisation des images en niveau de gris provenant du processus d' **analyse granulométrique** qui nous permettra aussi d'éliminer certains détails qui ne nous intéressent pas (bruit).

Pour ce faire, nous allons utiliser un **seuillage par hystérésis** car cette technique permet de récupérer certains détails de l'image sans se soucier du bruit. En effet, ce seuillage consiste à prendre un seuil haut et un seuil bas et à faire une opération de reconstruction géodésique du résultat du seuillage haut dans le seuillage bas. Le résultat de seuillage haut étant inclus dans celui du seuillage bas sur ces images, nous espérons ne garder que l'information qui nous intéresse en éliminant tout ce dont nous n'avons pas besoin.

Voir annexe A.4 et figure 6.

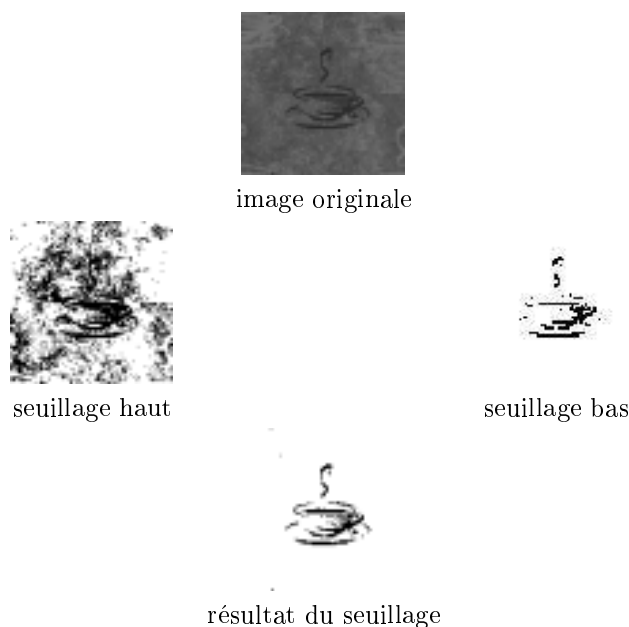


FIG. 6 – Seuillage par hystérésis.

Il faut cependant veiller à bien fixer les seuils haut et bas afin d'obtenir le résultat souhaité. La difficulté réside alors sur le choix pour la détermination de façon automatique des valeurs de ces seuils haut et bas. On somme les niveaux de gris sur l'image résultante de l'**analyse granulométrique** et l'on prend les valeurs correspondants à 90% et 98% de cette somme. Ces valeurs correspondant à peu près aux seuils transformant l'image en une image toute noire avec quelques pixels appartenant à la plaque et à celui transformant l'image en une image où la plaque est bien présente. On récupère ainsi une image binaire de la plaque d'immatriculation seule.

D'un point de vue technique, toutes les opérations de **morphologie mathématique** utilisées ici ont été portées sur la rétine par Antoine MANZANERA dans [Manzanera 2000] et sont traitées de façon optimale par cette architecture. cette

phase du processus sera donc menée très efficacement sur une **rétine artificielle**. Il en est de même pour les **opérateurs géodésiques** et notamment pour la **reconstruction géodésique** qui peut s'effectuer de manière parallèle sur une rétine. C'est ici que l'on introduit l'idée d'utiliser une **rétine artificielle** disposant d'une architecture asynchrone. En effet, une telle architecture est très efficace pour effectuer ce type d'opérations. L'idée est d'assouplir l'architecture SIMD synchrone par le biais des opérations de  $\theta$  - *associations*. Une **reconstruction géodésique** peut être réalisé à l'aide de *ou-associations* et de *max-associations*. Si notre application qui simule le travail d'une **rétine artificielle** met un certain temps sur une station de travail classique à effectuer le travail demandé, ce même processus sera très rapide sur une vraie rétine de par son architecture massivement parallèle.

voir Annexe A.3

## 3 Reconnaissance de caractères

### 3.1 Choix de l'outil de discrimination

Dans le domaine de la reconnaissance de caractères à l'aide de processeur numériques massivement parallèles, des travaux ont été menés par les équipes [Sackinger 1992] et [Cantoni 2000] avec des systèmes à base de **réseau de neurones**. C'est dans cet état d'esprit que nous avons réalisé un système à base de **perceptron multicouche** (**PMC** en français, **MLP** pour **Multi Layer Perceptron** en anglais) pour notre projet.

En effet, nous ne nous intéressons ici qu'à la reconnaissance des chiffres. Or, nous savons qu'il est assez simple de construire un **arbre de décision** pour discriminer les différents chiffres. Un **PMC** sera donc efficace pour réaliser une telle tâche puisque son rôle est de construire automatiquement un tel arbre en simulant sa structure avec la valeur des poids associés aux différents neurones. [Belaïd 1992]

Cet essai a été fructueux puisque le réseau mis en place a remarquablement joué son rôle. Nous avons obtenu tout de suite une classification efficace de notre base d'image. Ici nous sortons de la rétine et le **réseau de neurones** est implanté dans un module externe. Nous n'avons donc plus de problème de taille mémoire ni de capacité ou de vitesse de calcul. Nous verrons cependant en conclusion si une extension intéressante de ce travail ne serait justement pas de tenter de rester un peu plus sur la rétine en effectuant des prétraitements plus importants, notamment de moyen niveau, avant d'envoyer la sortie de la rétine en entrée du **réseau de neurones**. L'objectif final serait alors d'aller vers l'implémentation du **réseau de neurones** directement sur la rétine.

### 3.2 Mise en place de la reconnaissance

Un **réseau de neurones** est un ensemble de neurones interconnectés. Ses caractéristiques sont son architecture (qui définit la classe des fonctions calculables par un réseau), ses poids (qui définit une fonction de transition particulière) et ses hyperparamètres (dans la fonction de coût). Il fonctionne selon deux modes d'utilisation, l'un d'apprentissage à partir des données et l'autre d'utilisation sur des données nouvelles. Le premier mode servant à entraîner le réseau afin qu'il apprenne la structure de l'arbre de décision associé et le second mode sert à tester cet apprentissage et donc à discriminer les nouveaux exemples qu'on lui présente. [Gallinari 2002]

Le **perceptron multicouche** est une application du **perceptron** réalisé par ROSENBLATT dans les années 60 sur les **réseaux de neurones** actuels. C'est un **réseau de neurones** dont chaque cellule joue le rôle de **perceptron**. Globalement, son comportement se rapporte à l'**algorithme du Widrow-Hoff**. Voir Annexe A.5

Nous avons choisi de faire deux essais, l'un avec une architecture de réseau avec une couche cachée et l'autre sans couche cachée. Les résultats sont visibles sur la figure 8 et illustre l'efficacité de l'apport d'une couche cachée qui n'est pourtant pas nécessaire théoriquement. En effet, les couches cachées servent dans le cas de problèmes d'**apprentissages supervisés non linéairements séparables** ce qui n'est pas le cas ici. [Gallinari 2002]

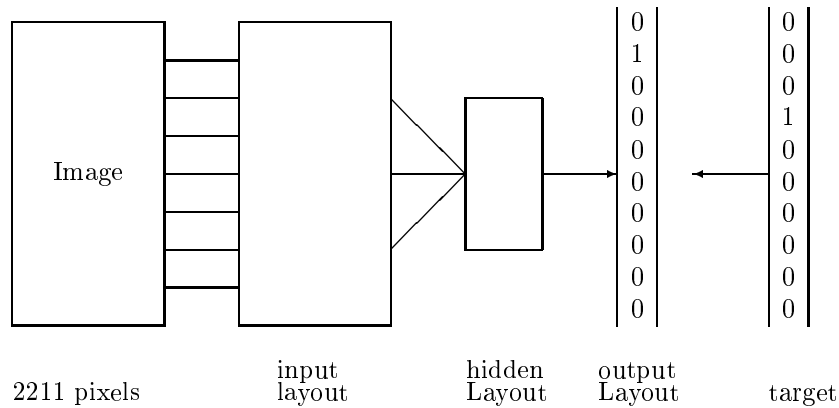


FIG. 7 – Architecture du Perceptron Multicouche.

Comme nous le montre la figure 8, la couche cachée permet d’avoir un erreur de classification qui chute très rapidement, c’est à dire une vitesse d’apprentissage plus élevée qu’un réseau sans couche cachée. De plus, cette erreur est bien moindre avec une couche cachée car le réseau cherche à séparer les classes d’apprentissage de façon non linéaire. Le résultat est alors bien plus complexe mais aussi bien plus efficace.

La présence de couches cachées dans un **réseau de neurones**, le nombre de couches et leurs tailles dépendent beaucoup d’un problème à l’autre, c’est pourquoi il est difficile de justifier précisément l’emploi d’une couche cachée. Cependant c’est la difficulté à modéliser et à résoudre un problème qui nous renseigne. De plus, et c’est le cas ici, on peut toujours ajouter une couche cachée afin de rendre le réseau efficace dans sa tâche de discrimination.

Afin de rendre plus simple le travail d’apprentissage, nous avons normalisé les images en entrée dans la mesure où nous avons fixé la taille des images à un rectangle 33x67 pixels (2211 pixels) qui correspondent aux 2211 neurones sur la couche d’entrée (*input layer*) du **PMC**. Nous avons choisi de normaliser les entrées conformément aux hypothèses de travail apportées au paragraphe 2.1. Ensuite comme nous l’avons dit, nous avons une couche cachée (*hidden layer*) de 100 neurones et enfin une couche de sortie (*output layer*) de la taille du nombre d’éléments à discriminer, soit 10 ici, pour les 10 chiffres. Nous comparons aussi lors de l’apprentissage le vecteur de sortie à un vecteur de sortie espéré (*target*) et nous ajustons les poids du réseau.

Voir figure 7.

Nous entraînons alors le réseau ainsi formé une centaine de fois afin de multiplier le nombre d’exemples à présenter. A la fin de cette phase d’apprentissage l’erreur de classification est de l’ordre du centième et nous pouvons donc désormais considérer que le réseau est apte à discriminer correctement les exemples qu’on lui présente. Ainsi, durant la phase de tests nous n’obtenons qu’un seul rejet pour toute la base d’apprentissage.

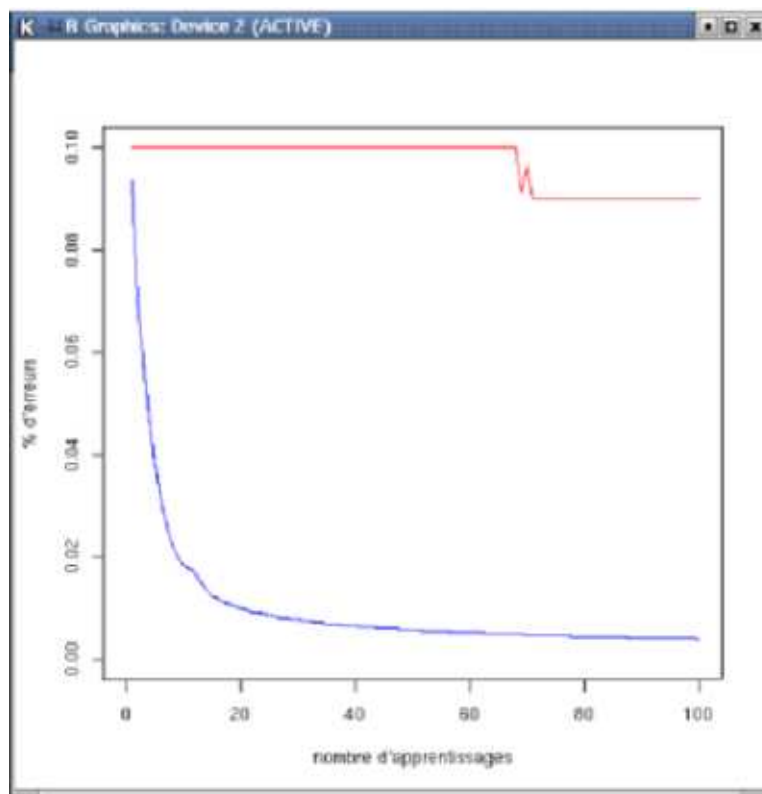


FIG. 8 – Comparaison entre mlp avec (bleu) et sans (rouge) couche cachées.



## 4 Résultats obtenus et interprétation

### 4.1 Un exemple concret

Pour illustrer le projet nous allons étudier le cas d'une image de référence. La première phase de localisation de la plaque dans l'image nous amène à effectuer un lissage de SOBEL en Y, un gradient horizontal et une binarisation. La figure 9 illustre les résultats que nous obtenons.

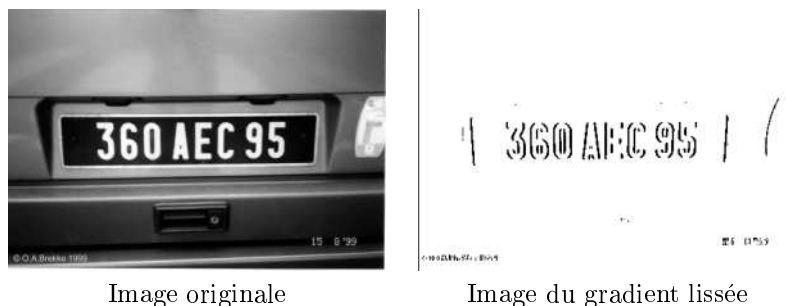


FIG. 9 – Première phase du traitement sur *f\_360AEC95.bmp*.

Après avoir déterminé s'il est nécessaire d'inverser ou non l'image que nous traitons (ici ce n'est pas le cas), nous effectuons l'**analyse granulométrique** à proprement parler, c'est à dire la suite d'ouvertures et de fermetures comme nous le montre la figure 10. Puis nous effectuons les top-hats correspondants ainsi que leurs différences.

Ensuite, il nous faut déterminer quelle est la taille de la boule optimale pour l'image traitée. Nous avons visualisé les calculs effectués automatiquement par le logiciel à l'aide de l'outil que nous avons justement nommé "*histo*" et dont le résultat est donné sur la figure 11. L'histogramme représente la fonction de répartition de la somme des niveaux de gris des pixels de chaque image (de la fermeture la plus large à l'ouverture la plus large) et la dérivée correspondante. Ainsi le logiciel peut déterminer à l'aide du maxima local correspondant à la **granulométrie** ou l'**anti-granulométrie** en fonction on non de l'inversion de l'image de la taille optimale de l'élément structurant qui est elle-même en fonction de l'empattement des caractères de la plaque minéralogique de la voiture photographiée. Il est à noter qu'ici nous évoluons dans le cas général où nous tentons de nous affranchir des hypothèses de travail. Ainsi, nous ne supposons pas connaître l'empattement des caractères et donc effectuons une **analyse granulométrique** complète.

Le logiciel peut alors réunir deux des images issues de la différence des top-hats puis effectuer le **seuillage par hystérésis** dont les valeurs de seuil haut et bas ont été déterminées selon la méthode décrite ci-avant. L'image résultante de tous ce processus peut enfin servir à la segmentation et la reconnaissance de caractères. La localisation nous a permis de connaître l'emplacement de la plaque, il nous suffit maintenant de repérer dans cette zone les pixels allumés (niveaux de gris "255") ou éteints (niveau de gris "0") pour séparer et donc segmenter les caractères de la plaque. Ces images sont ensuite normalisées pour être enfin envoyées dans le **PMC**.



FIG. 10 – Analyse granulométrique de *f\_360AEC95.bmp*.

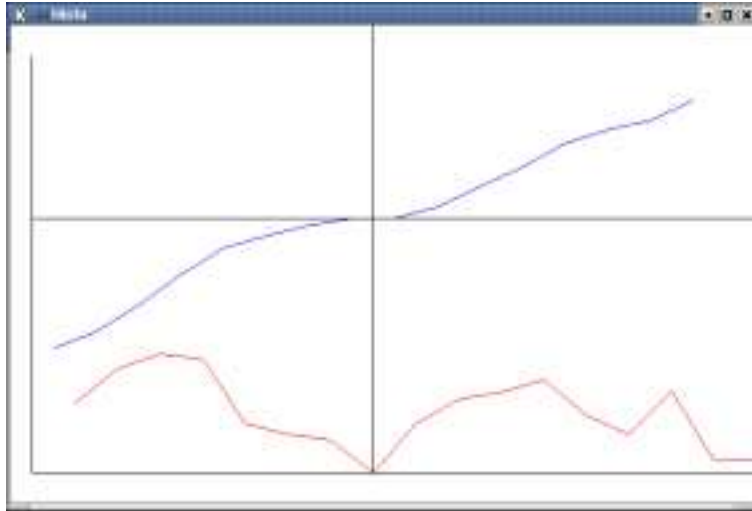


FIG. 11 – Histogramme de la fonction de répartition (bleu) et sa dérivée (rouge) pour *f\_360AEC95.bmp*.



Résultat de la granulométrie

Image seuillée

FIG. 12 – Résultat du traitement sur *f\_360AEC95.bmp*.

## 4.2 Cadre général

L'exemple que nous venons de détailler est, comme nous l'avons dit, un cas d'école ne nous servant que de référence. Or, il existe un nombre considérable d'images pour lesquelles tout ne se passe pas aussi bien. Tout d'abord, lors de la phase de localisation, des bruits divers peuvent perturber le logiciel qui place alors mal la fenêtre autour de la plaque. De ce fait, le logiciel peut prendre une mauvaise décision quant à l'inversion ou non de l'image. C'est la principale cause d'erreur du processus.

Le tableau ci-dessous illustre, pour la base dont nous disposons, des résultats que nous obtenons et leur répartition statistique. Nous appelons "*plaques localisées*" les images pour lesquelles la plaque d'immatriculation a été correctement localisée; "*traitées et localisées*" les images pour lesquelles tout le processus c'est correctement déroulé et enfin "*traitées parmi localisées*" les images pour lesquelles le processus de **granulométrie** et de segmentation s'est bien déroulé parmi les images pour lesquelles la phase de localisation a été une réussite.

nombre d'images	plaques localisées	traitées et localisées	traitées parmi localisées
65	50/65	50/65	50/50
-	76.93 %	76.93 %	100 %

Pour améliorer grandement ces résultats il nous faudrait revoir la méthode que nous employons pour la détermination automatique du choix de l'**analyse granulométrique** ou **anti-granulométrique** de l'image traitée. Cependant nous n'avons à ce jour toujours pas trouvé de technique nous permettant avec certitude quelle analyse employer.



FIG. 13 – Résultat du traitement sur diverses images.

## 5 Conclusion

Pour conclure, les objectifs que nous nous avons fixés pour ce projet ont été atteints dans la mesure où nous avons réalisé une application qui traite la détection et la reconnaissance de chiffres d'une plaque d'immatriculation avec un taux de réussite tout à fait acceptable. De plus, nous avons veillé à ce que les outils employés soit implémentable sur une **rétine artificielle** en les testant sur un simulateur déjà éprouvé. Cependant, deux points restent à explorer afin de parachever ce travail et apporter un sens nouveau au problème de reconnaissance de plaques d'immatriculation sur une **rétine artificielle**

Tout d'abord, il est probable que nous pouvons nous affranchir de la localisation de la plaque dans l'image et de ne faire qu'une **analyse granulométrique** (technique qui se prête particulièrement bien à une machine disposant d'une architecture massivement parallèle) en mettant au point une autre technique pour déterminer l'éventuel inversion des niveaux de gris de l'image. Une étude approfondie des fonctions de distributions résultantes de la **granulométrie** et de l'**anti-granulométrie** avait été menée mais sans apporter d'éléments probants.

Ensuite, comme cela avait été stipulé dans les hypothèses de travail, nous avons normalisé les images issues de la phase de segmentation afin de les envoyer dans le **réseau de neurones**. Concrètement cela revient à rééchantillonner ces images par un opérateur et donc à réduire la part d'automatisme du logiciel. Il serait bon de rendre automatique cette opération à l'aide d'un module externe qui n'a pu être écrit, faute de temps .

Enfin, une fois les poids du réseau déterminés pendant la phase d'apprentissage, il semble tout à fait plausible d'utiliser les caractéristiques de ceux-ci en terme de répartition et de valeurs dans le réseau pour construire des liaisons de voisinage cette fois-ci directement sur la rétine. Ainsi, c'est sur la rétine que se déroulerait la première étape de la reconnaissance. L'idéal serait de rester le plus longtemps sur la rétine afin d'user au mieux les capacités de calculs de celle-ci et son architecture massivement parallèle.

La reconnaissance de plaques d'immatriculation n'est pas un problème nouveau. Il a été déjà traité et résolu sur des machines classiques par d'innombrables équipes avec des approches différentes. La question que nous devons alors se poser en réalisant un tel projet est de se demander ce qu'il apporte? Son approche est-elle si différente de celles déjà abordées?

La vraie nouveauté du présent travail est qu'il utilise une machine utilisant une architecture massivement parallèle telle que la **rétine artificielle**. Il est difficile de comparer efficacement les résultats obtenus ici et ceux obtenus pour les autres approches, cependant il est certain que la structure spécifique d'une rétine permet d'avoir des temps de calcul très faibles. L'utilisation de l'**analyse granulométrique** s'impose alors d'elle-même comme nous l'avons vu précédemment.

Contrairement à ce qui était annoncé dans le sujet de ce stage, nous n'avons pas ou peu exploité les **architectures pyramidales** pour la phase de **reconnaissance des caractères** pour des raisons de temps et d'efficacité. En effet, nous nous sommes rendu compte que l'élaboration d'un **réseau de neurones** comme cela est fait ici est bien plus efficace et rapide que de chercher des structures de plus en plus complexes dans des voisinages de pixels. De plus, nous nous confrontons au problème de ne pouvoir tester nos algorithmes puisque la rétine sur laquelle nous travaillons de façon théorique n'existe pas encore aujourd'hui.

## Références

- [Adorni 1998] Giovanni ADORNI and Federico BERGENTI and Stefano CAGNONI, *Vehicule License Plate Recognition by means of Cellular Automata*, IEEE, 1998, 689–693.
- [Barroso 1997] J. BARROSO and A. RAFAEL and E.L. DAGLESS and J. BULAS-CRUZ, *Number plate reading using computer vision*, IEEE, 1997, 761–766.
- [Barthélémy 2000] Sylvain BARTHÉLÉMY, [http ://www.barth.netliberte.org](http://www.barth.netliberte.org), une librairie permettant de developper des perceptrons multicouches, 2000.
- [Belaïd 1992] Abdel BELAÏD and Yolande BELAÏD, *Reconnaissance des Formes*, InterEditions, 1992.
- [Bernard 1992] Thierry BERNARD, *Des Rétines artificielles intelligentes*, Thèse de doctorat, Université Paris-Sud, UFR Scientifique d’Orsay, Octobre 1992.
- [Brekke 2002] Olav Arne BREKKE, [www.olavsplates.com](http://www.olavsplates.com), [http ://www.olavsplates.com](http://www.olavsplates.com), une incroyable base de données d’images de plaques d’immatriculation sur l’Internet, 2002.
- [Brugge 1998] M.H. ter BRUGGE and J.H. STEVENS and J.A.G. NIJHUIS and L. SPAANENBURG, *License Plate Recognition Using DTCNNs*, IEEE, 1998, 212–217.
- [Cantoni 2000] Virginio CANTONI and alfredo PETROSINO, *2-D objet recognition by structural neural networks in a Pyramidal architecture*, IEEE, 2000, 81–86.
- [Ducourthial 1999] Bertrand DUCOURTHIAL, *Un modèle de programmation à parallélisme de données pour algorithmes et données irréguliers, à primitives de calcul asynchrones*, Thèse de doctorat, Université Paris-Sud, UFR Scientifique d’Orsay, Janvier 1999.
- [Dulac 1996] Didier DULAC, *Contribution au parallélisme massif en analyse d’image : Une architecture SIMD fondée sur la reconfigurabilité et l’asynchronisme*, Thèse de doctorat, Université Paris-Sud, UFR Scientifique d’Orsay, Janvier 1996.
- [Gallinari 2002] Patrick GALLINARI, *Apprentissage Numérique et Réseaux de neurones*, DEA IARFA, 2002.
- [Louchet 2002] Jean LOUCHET, *Introduction à la vision artificielle*, ENSTA, 2002.
- [Manzanera 2000] Antoine MANZANERA, *Vision Artificielle Rétinienne*, Thèse de doctorat, Ecole Nationale Supérieure des Télécommunications, 2000, Paris, France, Novembre.
- [Manzanera 2002] Antoine MANZANERA, *Cours de morphologie mathématique*, ENSTA 2e année, 2002.
- [Sackinger 1992] Eduard SÄCKINGER and Bernhard E. BOSER and Jane BROMLEY and Yann LECUN and Lawrence D. JACKEL, *Application of the ANNA Neural Network Chip to High-Speed Character*, IEEE Trans. on computer, 1992, 3, 3, 498–505.
- [Serra 2000] Jean SERRA, *Cours de Morphologie Mathématique*, Ecole des Mines de Paris, 2000.

# Annexes

## A Définitions

### A.1 Rétine artificielle

Le concept de "*rétine artificielle intelligente*" consiste en le rapprochement de fonctions d'acquisition et de traitements d'image au sein d'un même circuit intégré.

Le paradigme rétinien s'appuyant sur une structure périodique bidimensionnelle de transduction et de traitement, combinant en chaque pixel un élément photosensible, un dispositif analogique connexioniste paramétrable et un processeur booléen cellulaire programmable.

[Bernard 1992]

L'image physique est constituée d'une matrice d'échantillonnage spatial en deux dimensions d'une ou plusieurs caractéristiques de la scène réelle à un instant  $t$ . Chaque élément unitaire d'échantillonnage est un pixel. Celui-ci code une grandeur qui est usuellement le niveau de gris (mesure de la luminance) selon une longueur d'onde donnée.

L'architecture des **rétines artificielles** programmables actuelles est celle d'une machine parallèle SIMD synchrone. Chaque pixel est un processeur disposant d'un nombre de bits de mémoire limité qui peut effectuer certaines opérations élémentaires :

- Acquérir une image sur la rétine.
- Copier un plan sur un autre plan.
- Translater un plan d'un pixel dans l'une des quatre directions cardinales N, E, S, W.
- Effectuer le calcul d'une fonction booléenne à deux variables (ET, OU, OU exclusif, ...) entre deux plans.

Un **Opérateur de Rétine** est défini par la composition d'un nombre fini d'actions élémentaires.

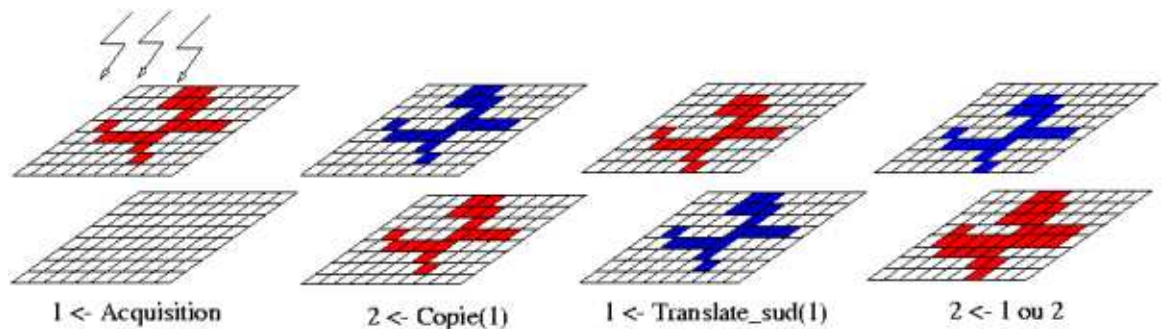


FIG. 14 – Une dilatation binaire sur la rétine décomposée en opérations élémentaires [Manzanera 2000].

Le processus d'acquisition sur la rétine produit une image binaire représentant les niveaux d'éclairement des pixels vis-à-vis d'un seuil prédéfini. Dans le circuit, on utilise une *photodiode* préchargée à une certaine tension, qu'on laisse ensuite se décharger. Pour obtenir une image en niveau de gris il suffit donc d'effectuer  $n$  lectures de la photodiode à des temps de référence correspondant à des tensions différentes et donc à des niveaux d'éclairement de la scène différents. [Manzanera 2000]

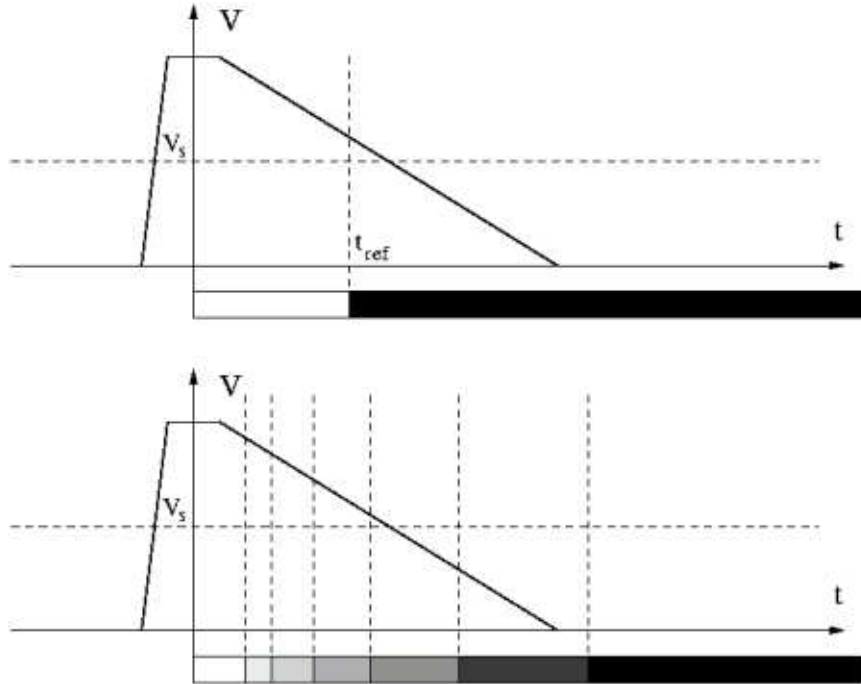


FIG. 15 – principe d'acquisition en niveau de gris par multiseuillage [Manzanera 2000].

Nous supposons une machine de  $P$  processeurs. Nous définissons  $P = \{P_i : i \in [0, P - 1]\}$  comme l'ensemble de ses processeurs. La topologie du réseau est a priori quelconque. Nous définissons  $\varphi = (P, \varepsilon)$  le graphe d'interconnexion physique interprocesseurs, avec  $\varepsilon$  l'ensemble des arcs de ce graphe. Nous supposons  $\varphi$  symétrique et de degré  $D$ .

Nous supposons qu'un ensemble de données réparties sur les processeurs définit une variable parallèle (ou *pvar*). Si  $x$  est une pvar, alors  $x_i$  représente la valeur de cette pvar sur le processeur  $P_i$ , et  $x_i^k$  son  $k$ ième bit de codage.

Les communications s'expriment à l'aide d'un sous-graphe de  $G$  appelé *mgraphe*. Un graphe  $g$  est un mgraphe si ses noeuds sont les noeuds de  $G$  et ses arêtes un sous-ensemble des arêtes de  $G$ .



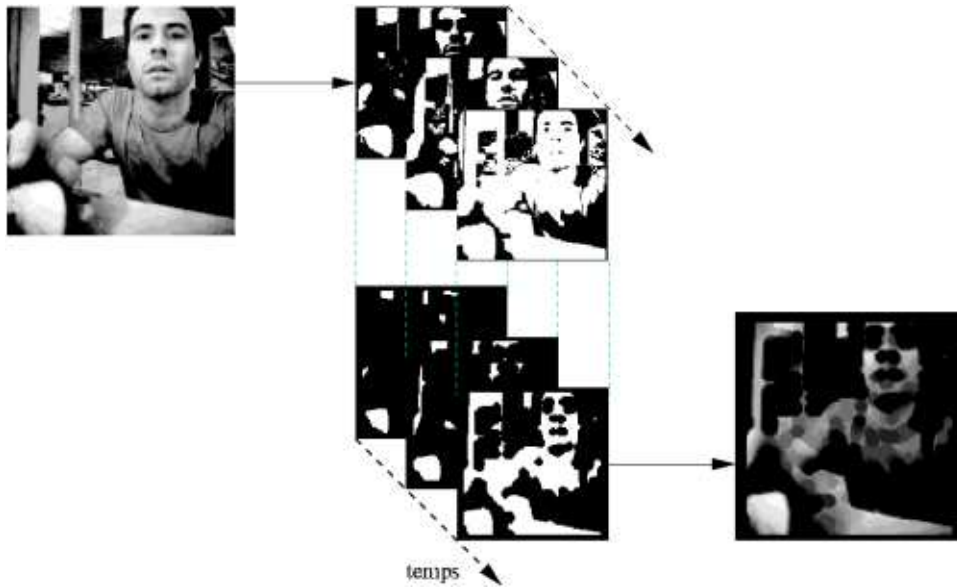


FIG. 16 – Principe de traitement en niveaux de gris par combinaison des traitements binaires effectués sur chaque seuil [Manzanera 2000].

L'opération de base réalisant des communications à l'aide d'un mgraphe est une *association*. Soit un opérateur associatif et communicatif  $\Theta$ , un mgraphe non orienté (symétrique)  $g$  et une pvar  $p$ , la  $\Theta$  – *association*( $g, p$ ) est une fonction  $(M, N^P) \mapsto N^P$  :

$$\Theta - association(g, p) = \{a_i : 0 \leq i < P \text{ et } a_i = \Theta_{j \in \gamma_g(i)} p_j\}$$

avec  $\gamma_g(i)$  sont les ancêtres d'un processeur dans un graphe  $g$ . [Dulac 1996]

En introduisant de tels opérateurs, nous pouvons réaliser alors des association non élémentaires qui peuvent ou doivent être réalisées sur une machine asynchrone.

Considérons que nous disposons d'une *ou-association* définie sur une machine synchrone (par exemple) comme suit :

```

ou-association(p,g)
répéter
ancien-p  $\leftarrow$  p
  pour tous les noeuds v voisins du noeud courant dans la maille
    si le noeud courant est connecté à v dans le graphe g
      p  $\leftarrow$  p—ancien-p[v]
    fin si
  fin pour
tant que p  $\neq$  ancien-p dans au moins un noeud de la maille
renvoyer p

```

Nous pouvons alors définir les  $\Theta$  – association suivantes :

$et - association(g, p) = \overline{ou - association(g, \bar{p})}$   
 $non - et - association(g, p) = \overline{ou - association(g, \bar{p})}$   
 $non - ou - association(g, p) = \overline{ou - association(g, p)}$   
 $min - association(g, p) = \overline{max - association(g, \bar{p})}$   
 [Dulac 1996]

```

max-association(g,p){
  pvar resultat Pour i=b à i=0 faire
    resultati = ou - association(g, pi)
    où(resultati == 1 et pi == 0) faire pi = 0
Fin Pour
  Retourner resultat
}
  
```

La mise en oeuvre matérielle de l'opérateur d'addition est un peu plus complexe que celle de l'opérateur ou-logique, notamment pour les opérations portant sur des *pvars* codées sur  $n$  bits. La réalisation d'une plus-association-élémentaire-1bit peut se faire à partir des bits transmis par chaque voisin : il suffit de compter et d'accumuler le résultat.

[Dulac 1996]

## A.2 Filtrage linéaire

La “convolution” du traitement d’images est la transformation d’images qui consiste à donner comme nouvelle valeur du pixel courant une combinaison linéaire des valeurs de gris d’un voisinage de ce pixel dans l’ancienne image.

$$f * g(x) = \int f(x+t)g(t)dt$$

La convoluée de  $f$  par  $g$  au sens du traitement d’images est en réalité le produit de convolution (au sens mathématique) de  $f$  par le symétrique  $S(g)$  de  $g$ .

$$I * C(c, y) = \sum_i \sum_j I(x+i, y+j)C(i, j)$$

On dit que  $I * C$  est l’image convoluée de  $I$  par le **convolution** de noyau  $C$ .  $C(i, j)$  est également appelée la réponse impulsionnelle du filtre.

[Louchet 2002]

	Linéaires	Non linéaires
Lissages	directionnels ou non-directionnels	filtre médian, min ,max, de Nagao
Gradients	Gradient EW, NS, etc..	norme du gradient, Sobel, Roberts, Kirsh, etc...
Laplacien	sur 4-voisins, 8-voisins, etc...	

On considère une convolution 3x3. Etant donné une image  $p$  et une matrice 3x3 :  $m = \{m_{ij} : -1 \leq i \leq 1, -1 \leq j \leq 1\}$ , la convolution de  $p$  par  $m$  donne une image  $q$  :

$$q_{l,m} = \sum_{-1 \leq i \leq 1} \sum_{-1 \leq j \leq 1} m_{ij} x_{pl+i, m+j}$$

Les différents coefficients  $m_{ij}$  sont usuellement représentés sous la forme d’une matrice correspondant au masque de convolution physiquement applicable au pixel et à son voisinage :

$$\text{Masque de convolution} = \begin{vmatrix} m_{1,1} & m_{0,1} & m_{-1,1} \\ m_{1,0} & m_{0,0} & m_{-1,0} \\ m_{1,-1} & m_{0,-1} & m_{-1,-1} \end{vmatrix}$$

Nous supposons que toutes les données sont codées sous la forme d’entiers non signés ; en particulier  $m$  sera codé sur  $b$  bits, et  $p$  codé sur  $c$  bits. On peut donc écrire :

$$\begin{aligned} q_{l,m} &= \sum_{-1 \leq i \leq 1} \sum_{-1 \leq j \leq 1} \sum_{r=0}^{b-1} 2^r m_{ij}^r x_{pl+i, m+j} \\ &= \sum_{r=0}^{b-1} 2^r \sum_{-1 \leq i \leq 1} \sum_{-1 \leq j \leq 1} m_{ij}^r x_{pl+i, m+j} \\ &= \sum_{r=0}^{b-1} 2^r \text{plus-association-elementaire}(g_r, p) \end{aligned}$$

où  $g_r$  est défini sur chaque PE par le  $i$ ème bit des éléments de  $m$ . Le masque local correspondant est  $g_r = \{m_{ij}^r : -1 \leq i \leq 1, -1 \leq j \leq 1\}$ .

[Dulac 1996]

### A.3 Morphologie mathématique

Dans la **morphologie mathématique** applicable aux images en niveaux de gris créée par Georges MATHERON, on considère les images comme une partie de  $E = R^N$  ou de  $E = Z^N$ . On lui applique alors des opérateurs “et”, “ou” afin d’obtenir des opérations d’érosions, de dilatations, etc...

[Louchet 2002]

#### Granulométrie

La granulométrie est l’étude de la taille des objets. En physique, elle repose principalement sur le tamisage (grains), et l’injection de mercure (milieux poreux). Or, les grains refusés par le tamis de taille  $\lambda$ , qui sont une partie de la population initiale, sont refusés aussi par tout nouveau tamisage de taille  $\mu \leq \lambda$ . Il faudra aussi prendre en compte la dépendance des concepts de taille et de forme et le rôle de l’homothétie. Elle consiste donc en une suite d’ouvertures ou de fermetures par des familles d’éléments structurants qui dépendent d’un paramètre positif, la taille. [Serra 2000]

L’axiomatique de MATHERON définit comme **granulométrie** toute famille :

- i) d’ouvertures  $\{\gamma_\lambda\}$  dépendant d’un paramètre positif  $\lambda$ ,
- ii) et fonctions décroissantes de  $\lambda : \lambda \geq \mu > 0 \Rightarrow \gamma_\lambda \leq \gamma_\mu$ .

Ce second axiome est équivalent au semi-groupe où la composition de deux opérations est égale à la plus sévère :

$$\gamma_\lambda \gamma_\mu = \gamma_\mu \gamma_\lambda = \gamma_{\sup(\lambda, \mu)}$$

(1)

$[\lambda \geq \mu > 0 \Rightarrow \gamma_\lambda \leq \gamma_\mu \Leftarrow \gamma_\lambda = \gamma_\lambda \gamma_\mu \leq (\gamma_\lambda \gamma_\mu \vee \gamma_\mu \gamma_\lambda) \leq \gamma_\mu]$

inversement,  $\gamma_\lambda = \gamma_\mu \gamma_\lambda$  et  $\gamma_\lambda \leq I \Rightarrow \gamma_\lambda \leq \gamma_\mu$  d’où le semi-groupe.]

Si  $B_\lambda$  et  $B_\mu$  désignent les invariants de  $\gamma_\lambda$  et de  $\gamma_\mu$ , on voit facilement que

$$(1) \Leftrightarrow B_\lambda \subseteq B_\mu$$

Enfin quand les  $\gamma_\lambda(X) = X \circ \lambda B$  sont les ouvertures par adjonction selon des éléments structurants **homothétiques**  $\lambda B$ , on obtient une **granulométrie** si et seulement si  $B$  est **convexe**.

#### Mesure granulométrique

Une **granulométrie** se calcule à l’aide d’une pyramide de filtres dont chaque élément (ouverture ou fermeture) agit selon une taille donnée  $\lambda$ ;  $\lambda$  est le facteur d’homothétie portant sur le ou les éléments structurants mis en jeu.

A la sortie de chaque filtre  $\lambda$  on mesure l’aire (ensembles) ou l’intégrale (fonctions), soit  $M_\lambda$ . la quantité :

$$F_\lambda = 1 - M_\lambda / M_0$$

est une **fonction de distribution**.

#### Boules

Soit  $X$  une image en niveaux de gris.

Soit  $(B_\lambda)_{\lambda \in N}$  une famille de boules euclidienne définies par :

$$B_e(x, n) = \{z \in Z \mid d_e(x, z) \leq n\}$$

on note  $\check{B} = -B$ .

en pratique afin, de discrétiser les boules on utilise le code suivant :

```
#define NMAX 15

int boule[NMAX][NMAX];

initBoule(int taille)
{
    double val;

    for (int i=0; i<NMAX; i++)
        for (int j=0; j<NMAX; j++)
            boule[i][j]=0;

    for (int i=0; i<NMAX; i++)
        for (int j=0; j<NMAX; j++)
            {
                val = sqrt(pow(((NMAX/2)-i),2)+pow(((NMAX/2)-j),2));
                if (val<n)
                    boule[i][j]=1;
                else
                    boule[i][j]=0;
            }

    if (n==1)
        boule[NMAX/2][NMAX/2]=1;
}
```

### opération de Minkowski

$X$  et  $B$  deux sous-ensembles de  $\mathbb{Z}^2$ .

L'addition de Minkowski de  $X$  et  $B$  est l'ensemble :

$$X \oplus B = \bigcup_{b \in B} X_b.$$

La soustraction de Minkowski de  $X$  par  $B$  est l'ensemble :

$$X \ominus B = \bigcap_{b \in B} X_b.$$

où  $X_b$  désigne le translaté de  $X$  par  $b$ .

[Manzanera 2000]

### Dilatation

Dilaté de  $X$  par l'élément structurant  $B$  :

$$D(X, B) = \{x \in E \mid B_x \cap X \neq \emptyset\} = X \oplus \check{B}$$

Propriétés de la dilatation

$$X \subset Y \Rightarrow D(X, B) \subset D(Y, B)$$

$$\begin{aligned}
D(X, B \cup C) &= D(X, B) \cup D(X, C) \\
D(X, B \cap C) &= D(X, B) \cap D(X, C) \\
D(X \cup Y, B) &= D(X, B) \cup D(Y, B) \\
D(X \cap Y, B) &= D(X, B) \cap D(Y, B) \\
D(D(X, B), C) &= D(X, B \oplus C)
\end{aligned}$$

**Erosion**

Erodé de  $X$  par l'élément structurant  $B$  :

$$E(X, B) = \{x \in E \mid B_x \subset X\} = X \ominus \check{B}$$

Propriétés de l'érosion

$$\begin{aligned}
E(X, B) &\subset X \text{ si et seulement si origine} \in B \\
X \subset Y &\Rightarrow E(X, B) \subset E(Y, B) \\
E(X, B \cup C) &= E(X, B) \cup E(X, C) \\
E(X, B \cap C) &= E(X, B) \cap E(X, C) \\
E(X \cup Y, B) &\subset E(X, B) \cup E(Y, B) \\
E(X \cap Y, B) &= E(X, B) \cap E(Y, B) \\
E(E(X, B), C) &= E(X, B \oplus C) \\
D(E(X, B), C) &\subset E(D(X, C), B) \\
X \ominus B &= (X^c \oplus B)^c
\end{aligned}$$

**Ouverture**

Ouverture d'élément structurant  $B$  = érosion par  $B$  puis dilatation par  $-B$  :

$$X_B = D(E(X, B), -B) = (X \ominus (-B)) \oplus B = \gamma_B(X)$$

Propriétés de l'ouverture

$$\begin{aligned}
\text{si } \forall \lambda \in NB_{\lambda+1} = B_\lambda \oplus B_1 \text{ alors } X_{\lambda+1} &= \gamma_{B_\lambda}(X_\lambda) \\
X_B &\subset X \\
X \subset Y &\Rightarrow X_B \subset Y_B \\
(X_B)_B &= X_B
\end{aligned}$$

**Fermeture**

Fermeture d'élément structurant  $B$  = dilatation par  $B$  puis érosion par  $-B$  :

$$X^B = E(D(X, B), -B) = (X \oplus (-B)) \ominus B = \varphi_B(X)$$

Propriétés de la fermeture

$$\begin{aligned}
X_B &\subset X \subset X^B \\
X \subset Y &\Rightarrow X^B \subset Y^B \\
(X^B)^B &= X^B \\
\neg(X^B) &= (\neg X)_B
\end{aligned}$$

**Transformé en Tout-ou-rien**

Soient  $H$  et  $M$  deux sous-ensembles de  $Z^2$  tels que  $H \cap M = \emptyset$ . La **transformé en tout-ou-rien** (TTR) de  $X$  par le couple  $(H, M)$  est l'ensemble :

$$X \otimes (H, M) = (X \ominus \check{H}) \cap (X^c \ominus \check{M})$$

**Top-hat**

La transformation top-hat ou chapeau haut-de-forme de l'image  $f$  par l'élément structurant binaire  $B$  est définie comme  $f - f_B$ . L'ouverture de  $f_B$  de  $f$  par  $B$  élimine les pics plus étroits que l'élément structurant. La transformée en chapeau haut-de-forme détecte donc les pics étroits.[Louchet 2002]

$$\tau_B(X) = X - \gamma_B(X)$$

**Gradient morphologique**

Le gradient morphologique est défini habituellement par :

$$g_B(X) = (X \oplus \check{B}) - (X \ominus \check{B})$$

que l'on peut décomposer en gradient interne :

$$g_B^i(X) = X - (X \ominus \check{B})$$

et externe :

$$g_B^e(X) = (X \oplus \check{B}) - X$$

[Manzanera 2000]

## A.4 Opérateurs géodésiques

### Distance géodésique

Soit  $X \subset E$ . Pour tout  $(x, y) \in X^2$ , la distance géodésique de  $x$  à  $y$  dans  $X$  est définie par :

$$\delta_X(x, y) = \inf\{\ell(\gamma); \gamma \in \Gamma(x, y), \gamma \subset X\}.$$

Un arc  $\gamma \in \Gamma(x, y)$  est une *géodésique* de  $x$  à  $y$  dans  $\ell(\gamma) = \delta_X(x, y)$ .

### Boule géodésique

La boule géodésique dans  $X$ , de centre  $x \in X$  et de rayon  $r$  est définie par :

$$B_X(x, r) = \{y \in X; \delta_X(x, y) \leq r\}.$$

### Dilatation géodésique

Soit  $X$  et  $Y$  deux sous-ensembles de  $E$ . La dilatation géodésique de  $X$  dans  $Y$ , de rayon  $r$ , est l'ensemble :

$$d_Y^r(X) = \{y \in Y; X \cap B_Y(y, r) \neq \emptyset\}.$$

### Reconstruction géodésique

La reconstruction géodésique de  $X$  dans  $Y$  est l'ensemble :

$$R_Y(X) = \bigcup_{r>0} d_Y^r(X).$$

on notera en outre :

$$E_Y(X) = R_{Y \cup X}(X).$$

Notons que si  $X \subset Y$ , alors  $E_Y(X) = R_Y(X)$ .

### Seuillage par hystérésis

Soit  $I : Z^2 \rightarrow N$  une image en niveaux de gris. Soient deux entiers  $a$  et  $b$  tels que  $0 < a < b$ . Le seuillage de niveau  $n \in N$  est l'image binaire :

$$I_n = \{z \in Z^2; I(z) \geq n\}$$

Le seuillage par hystérésis de  $I$  sur  $[a, b]$  est l'image binaire suivante :

$$I_a^b = R_{I_a}(I_b)$$

[Manzanera 2000]



### A.5 Histogramme

L'histogramme se définit comme une fonction  $F(n)$  du niveau de gris  $n$  qui renvoie, pour chaque niveau de gris, le nombre de pixels ayant la même valeur :

$$H(n) = \sum_{n=0}^{\text{niveau de gris maximum}} \text{Pixels de l' image ayant le niveau de gris } n$$

```

int resultat[niveau de gris];
pvar image, graphe, candidat, racine-centrée;
where (self-address-grid()==taille-image/2) racine-centrée ← 1
graphe ← recouvrement-par-un-arbre( $\mathcal{G}$ ,racine-centrée);
for i=0 to niveau-de-gris-maximum
candidat ← 0;
where (image==i) candidat ← 1;
resultat[i] ← Plus-association(candidat,graphe);
end for
retourner resultat[]

```

[Dulac 1996]

## A.6 Réseaux de neurones

Les méthodes **connexionnistes** sont inspirées par M.DORÉ sur la reconnaissance de caractères. Le principe des **réseaux de neurones** est de transmettre des informations en entrée d'un réseau de processeurs élémentaires organisés en couches superposées interconnectées et dont on compare la couche de sortie avec la sortie désirée (dans le cas d'apprentissage supervisée) afin d'ajuster les poids des différents neurones du réseau.[Belaïd 1992]

### L'algorithme du Widrow-Hoff

Règle du WIDROW-HOFF - gradient *stochastique* :

Initialiser  $W(0)$   
 Itérer  
 Choisir aléatoirement un exemple  $(x^t, d^t)$   
 $W(t+1) = W(t) - \epsilon(t) \cdot [W(t)x^t - d^t]x^t$   
 Critère d'arrêt

### Le perceptron de Rosenblatt

Le **Perceptron** est utilisé pour la discrimination.

Chaque cellule d'association  $f_i : S^n \rightarrow S$

La cellule de décision calcule une *fonction à seuil*  $F = 1[\sum_i w_i f_i]$

Soit

- $Classe1 = \{x : F[x] = 1\}$
- $Classe2 = \{x : F[x] = -1\}$

L'algorithme du **Perceptron** (cas à 2 classes)

Sortie désirée  $\pm 1$   
 Initialiser  $W(0)$   
 Répéter (t)  
 Pour  $i = 1$  à  $N$   
 Si  $d^i(W(t) \cdot x^i) \leq 0$  alors  $W(t+1) = W(t) + \epsilon D^i \cdot x^i$   
 Jusqu'à convergence  
 Décision :  $F(x) = \text{sign}(\sum_{i=1}^n w_i x_i)$

C'est un algorithme de correction d'erreur.

si  $\epsilon$  est constant : règle à incrément fixe

si  $\epsilon$  est fonction du temps : règle à incrément variable

### Le perceptron Multicouche

Les paramètres du PMC sont définies ainsi :

$$x_i = g[A_i]$$

avec

$$A_i = \sum_j \omega_{ij} x_j$$

$$y^k = F(x^k)$$

$$g(v) = \frac{1}{1+e^{-kv}} \text{ Sigmoide}$$

- $x_i$  : cellules d'association ;
- $A_i$  : structure du modèle ;
- $\omega_{ij}$  : poids  $i$  de la cellule  $j$  ;
- $y^k$  : sortie  $k$  ;
- $F(x^k)$  : fonction à seuil

## B Applications développées pour le projet

Tout au long du travail effectué pour ce projet, nous avons développé du code que nous avons porté sur un logiciel existant ou sur une application que nous avons maintenu. Nous présentons ici ces divers travaux par ordre chronologique.

### Contribution à inti

Tout d'abord, nous avons testé tous les algorithmes que nous imaginions pour ce projet sur les logiciels développés par Antoine MANZANERA. Nous avons ainsi ajouté certains outils sur son application *Inti*.

### TraitLib

Pour des soucis de confort, nous avons élaboré un logiciel personnel dans le même esprit que *Inti* mais intégrant certaines fonctionnalités que nous voulions tester. Nous avons donc développé le logiciel *TraitLib* qui est maintenant assez mûre et intègre la plupart des algorithmes classique en traitement d'images. En réalité ce logiciel existait déjà depuis un module MIR du DEA et nous n'avons fait que le faire évoluer vers une application plus "professionnelle".

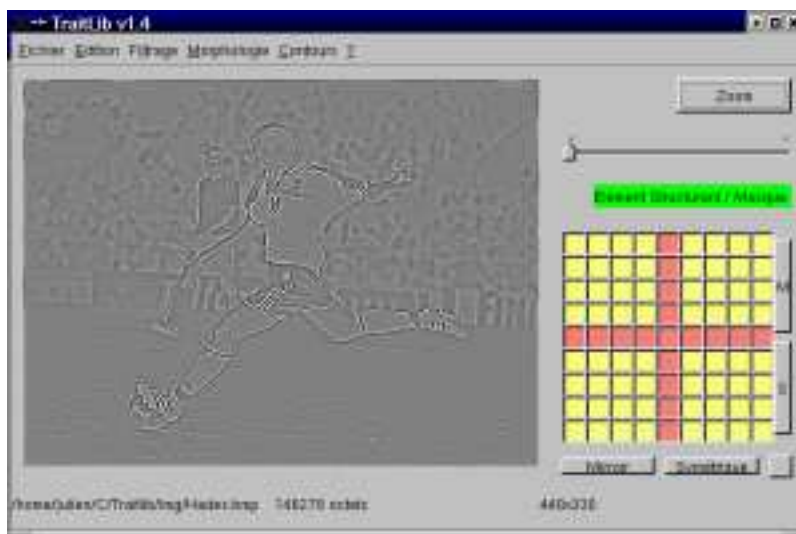


FIG. 17 – Logiciel TraitLib

## Granu

Enfin, nous avons développé une application spécifique au sujet du stage, qui nous permet de visualiser rapidement les résultats que nous obtenons avec les algorithmes implémentés en reconnaissance de plaques d'immatriculation. Il nous permet de suivre le cheminement que nous effectuons pour mener à bien la reconnaissance de plaques d'immatriculation.



FIG. 18 – Logiciel Granu