

Systeme d'interpretation mixte : reseaux de neurones/systeme-expert applique aux images aeriennes

A Mixed System of Interpretation : Neural Networks/Expert-System Applied to Aerial Images



Jean-Pierre COCQUEREZ,
ENSEA - ETIS,
1 Allée des Chênes Pourpres,
95014 Cergy-Pontoise Cedex, France,

J. P. Cocquerez est ingénieur de l'ENSEA. Après avoir fait sa recherche en électronique rapide (chronométrie sub-nanoseconde) pour sa thèse de docteur-ingénieur, il s'intéresse aux traitements bas-niveau en analyse d'images et à l'extraction de primitives dans les images aériennes pour soutenir une thèse d'état en 1984. Actuellement, il effectue sa recherche en interprétation d'images en approfondissant les approches « systèmes expert » et neuromimétiques. J. P. Cocquerez est responsable de l'Équipe Traitement des Images et du Signal de l'ENSEA (ETIS) et du DEA Traitement des Images et du Signal de l'Université de Cergy. Il est également animateur de l'opération « segmentation » du GDR 134 du CNRS.



Philippe GAUSSIER,
ENSEA - ETIS,
1 Allée des Chênes Pourpres,
95014 Cergy-Pontoise Cedex, France,

Ph. Gaussier est né en 1967. Après une maîtrise EEA et un DEA en électronique à l'INSTN, il termine un Doctorat en électronique à l'Université Paris XI. Ses travaux portent sur l'apprentissage en analyse d'images et sur la simulation des mécanismes de vision à l'aide de réseaux de neurones.



Sylvie PHILIPP
ENSEA - ETIS,
1 Allée des Chênes Pourpres,
95014 Cergy-Pontoise Cedex, France,

Après une maîtrise et un CAPES de Mathématiques, S. Philipp a enseigné les Mathématiques dans le secondaire de 1977 à 1984. Elle a obtenu son Doctorat d'Informatique de l'Université Paris VI, option Traitement des Images en 1988. Sa thèse, effectuée à la Direction des Études et Recherches d'EDF portait sur l'analyse de texture appliquée aux radiographies de pièces métalliques. Elle est actuellement maître de conférence et enseigne à l'ENSEA et dans le DEA Traitement des Images et du Signal de l'Université de Cergy. Depuis 1988, ses activités de recherche se font dans le laboratoire de l'ETIS, à l'ENSEA, et portent sur les traitements bas-niveaux (segmentation et analyse de texture) et sur l'interprétation d'image.

RÉSUMÉ

Dans cet article, nous proposons un système complet d'analyse d'images comprenant toute la chaîne de traitements depuis le bas-niveau jusqu'à l'interprétation. Il utilise à la fois un réseau de neurones et un système à base de règles. Nous montrons que la mise en œuvre d'un système-expert fournit des informations précieuses pour la conception des réseaux. La réalisation mixte permet d'utiliser au mieux les spécificités de chacune

des approches. Nous montrons également comment faire apprendre des configurations localement contradictoires à un réseau de neurones.

MOTS CLÉS

Analyse d'image, image aérienne, réseau de neurones, système à base de règles, système-expert, reconnaissance des formes, apprentissage.

SUMMARY

In this paper, we propose a complete system of analysis of images, which includes the whole sequence of treatments from the low level until the interpretation. It uses neural networks as well as a rule-based system. We show that the implementation of an expert-system gives useful information for the conception of the neural nets. The mixed realisation allows us to use at best the specificities of each approach. We also show how to make a neural network learn locally contradictory configurations.

KEY WORDS

Image analysis, aerial image, neural network, rule-based system, expert-system, pattern matching, learning.

1. Introduction

L'objectif de cet article est de montrer l'intérêt d'une approche mixte réseaux neuronaux-système expert pour l'interprétation d'images. Nous mettons en évidence un certain nombre de caractéristiques intéressantes et nous donnons des indications pour la mise en œuvre d'une telle approche.

Notre champ d'application est l'analyse d'images aériennes de zones urbaines et péri-urbaines dont l'interprétation est complexe du fait de l'ambiguïté des situations et des problèmes de segmentation rencontrés. Nous cherchons plus particulièrement à reconnaître les structures artificielles comme les bâtiments dans une des images d'un couple stéréoscopique, afin de les reconstruire par la suite.

Plusieurs travaux portent sur l'interprétation d'images aériennes. La plupart font appel à des modèles plus ou moins complexes de la scène observée.

Pour Nevatia et Price [20], le modèle de la scène est représenté par un réseau sémantique. A chaque nœud est associé un objet ou une partie d'objet, les arcs traduisent les relations entre objets. Les nœuds sont étiquetés avec les attributs des objets. Une description de l'image est élaborée sous la forme d'un graphe qui a la même structure que le modèle et qui intègre les résultats de deux techniques de segmentation utilisant l'approche frontière et l'approche région. Ensuite, une mise en correspondance est faite entre le modèle et la description par une technique de relaxation.

Nagao et Matsuyama [19] extraient les caractéristiques des régions de l'image. Ensuite, plusieurs modules dédiés à l'identification d'objets particuliers (route, bosquet...) sont mis en œuvre. Les résultats fournis par les modules dédiés sont rassemblés et chaque module examine à nouveau les cas ambigus jusqu'à ce qu'il n'y ait plus de nouvelle identification.

Huertas et Nevatia [8] traitent de la reconnaissance de bâtiments et extraient des primitives de types segments et angles. En tenant compte des ombres, ils tracent les frontières des structures en respectant des contraintes de compatibilité pour construire des formes correspondant à des modèles de bâtiments en L, T, U. Ensuite, ils affinent la reconnaissance par une procédure de vérification d'hypothèses. Les images qu'ils traitent sont de bonne qualité et les bâtiments sont isolés. Plus récemment les

mêmes auteurs [7] [17] utilisent une démarche voisine pour l'analyse d'images d'aéroport.

Mc Keown [14] [15] propose un système-expert d'analyse d'images aériennes d'aéroport. La base de connaissance comprend en plus des données images, des données cartographiques et architecturales. Plusieurs techniques de segmentation (approche frontière, région et texture) sont exploitées. Le nombre de règles est important, elles ont nécessité le réglage d'un grand nombre de paramètres.

Signalons aussi, les travaux de Irwin et Mc Keown [9] qui utilisent les caractéristiques des ombres pour reconnaître et reconstruire les bâtiments.

En ce qui concerne notre étude, comme la plupart des auteurs précités, nous effectuons une segmentation qui permet d'élaborer une description de l'image sous forme d'un graphe d'adjacence de régions. Plutôt que de spécialiser des opérateurs comme Nagao et Matsuyama, ou de mettre en correspondance la description avec un graphe modèle, nous effectuons une classification à partir des attributs associés à chaque sommet du graphe d'adjacence. Nous préférons cette approche car elle traite l'image dans son ensemble, sans employer d'opérateur spécialisé pour un type d'objet. Même si elle fournit un étiquetage multiple des régions, elle est plus générale parce que l'adaptation à d'autres catégories d'images s'opère par un nouveau réglage des seuils et parce qu'elle ne nécessite pas la mise en œuvre d'opérateurs dédiés délicats à ajuster. Ensuite pour résoudre le problème de l'étiquetage multiple, nous utilisons un algorithme de propagation de contraintes qui exploite les relations de voisinage.

Pour la classification et la propagation de contraintes, nous avons dans un premier temps suivi une approche de type système-expert. Malheureusement la généralisation à un nombre important d'images différentes rend nécessaire la fabrication de nouvelles règles de classification. Il est difficile d'imaginer ces règles et surtout de vérifier qu'elles ne sont pas en contradiction avec des règles déjà créées.

D'autre part, les caractéristiques utilisées pour la reconnaissance étant des valeurs analogiques, nous voulions que le système puisse déterminer automatiquement les différents seuils dont il avait besoin et ce de façon optimale en regard d'une catégorie d'images. Nous nous sommes donc tournés vers une approche neuronale qui utilise un apprentissage, supprimant ainsi la phase de recherche de nouvelles règles et de seuils.

Nous avons développé deux systèmes pour l'interprétation

(l'un à base de règles et l'autre connexionniste). Finalement, pour les raisons évoquées ci-dessus et déjà présentées dans les travaux de Gutknecht et Pfeifer [5] ou de Hillman [6] (adaptabilité, réglage des seuils, généralisation), nous avons opté pour un système mixte réseau de neurones (R.N.)/système-expert. Une première classification est réalisée par un réseau de neurones multicouche alors que la résolution des conflits est réglée par un système-expert. L'approche neuronale est intéressante car elle permet de traiter les connaissances implicites décrites par un ensemble d'exemples ; les gros problèmes sont ceux du dimensionnement du réseau et de l'apprentissage de configurations contradictoires (« multiple credit assignement problem »).

En résumé, notre approche propose des solutions pour la détermination des seuils nécessaires à la classification des régions et la partie propagation de contraintes comporte beaucoup moins de règles que le système de Mc Keown. Notre système est de ce fait plus simple à mettre en œuvre. Il permet d'identifier correctement plusieurs types d'objets dans l'image (bâtiments, pelouses, bosquets, ...). Nous avons utilisé ces résultats dans le cadre d'une autre étude

pour l'identification de bâtiments et leur reconstruction tridimensionnelle à partir d'un couple stéréoscopique.

2. Présentation générale

Nous proposons un système d'interprétation d'images dont la structure est indiquée figure 1. Dans un premier temps, nous présenterons les opérations bas-niveau effectuées sur l'image : extraction de contours, fermeture des contours, polygonalisation, extraction des composantes connexes et fabrication d'un graphe d'adjacence comportant les caractéristiques de chaque région. Ensuite, nous développerons la partie interprétation avec les différentes interactions entre un système à base de règles et un réseau de neurones. Une fois l'interprétation faite, le système peut reconstruire les bâtiments à partir d'un couple stéréoscopique [10] [28].

Partant d'une image de niveaux de gris prise par un avion, notre but est d'identifier les différents objets contenus dans la scène. Comme T. Matsuyama [16], nous différen-

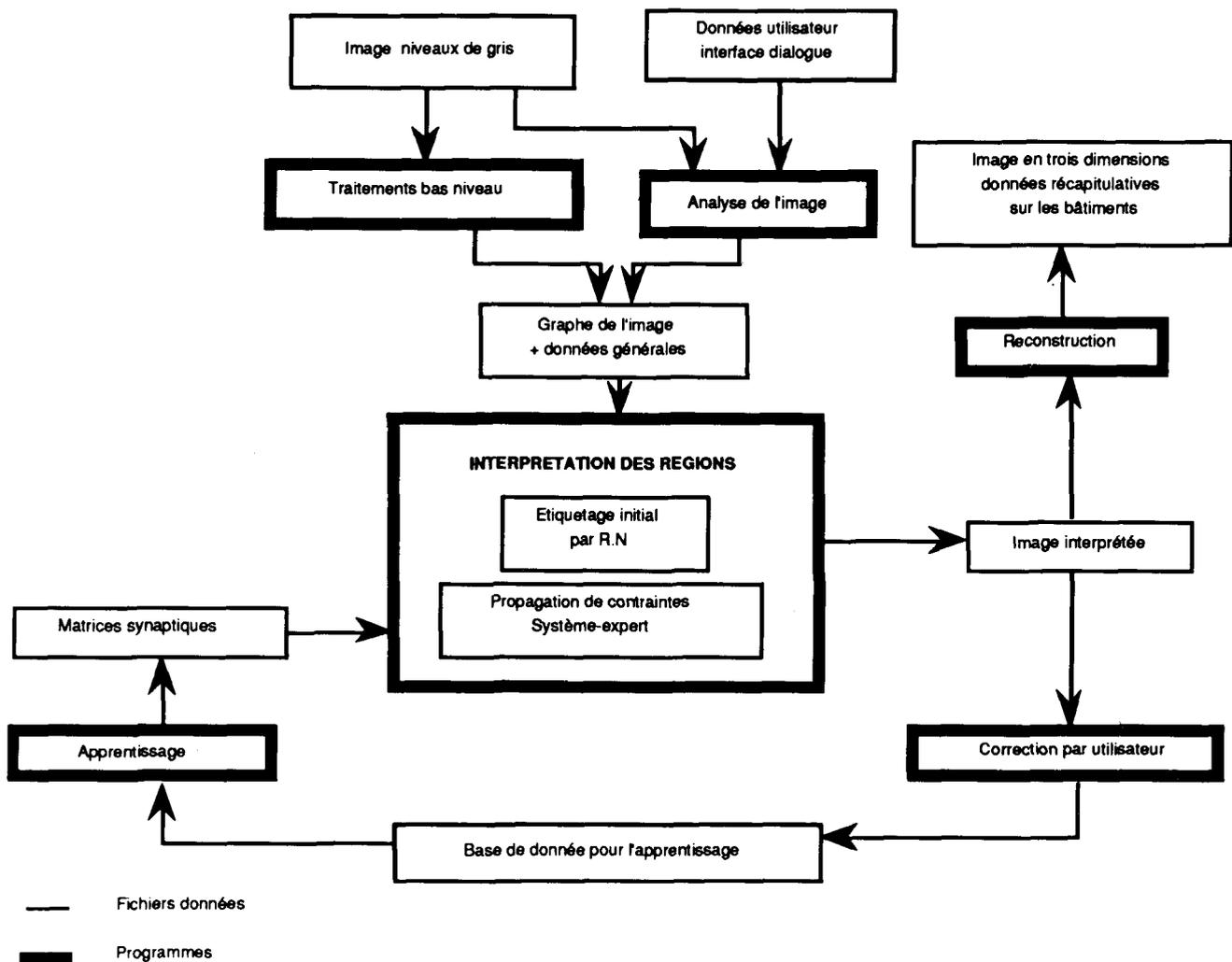


Figure 1. — Diagramme général du système d'interprétation.

cions le **domaine de la scène** du **domaine de l'image**. Le domaine de la scène comprend les objets réels ayant un contenu sémantique riche tels que bâtiments, routes, arbres, ... ; les relations entre ces données sont d'ordre topologique. Dans le domaine de l'image, on utilise des primitives telles que contours ou régions sur lesquelles on calcule des paramètres de forme, taille, position relative, ... ; leur contenu sémantique est pauvre, mais il est quantifiable.

2.1. TRAITEMENTS BAS-NIVEAU : SEGMENTATION PAR FERMETURE DE CONTOURS

Les algorithmes utilisés pour segmenter les images ont été exposés dans un précédent article [2]. Nous appliquons en premier lieu un **détecteur de contour** comprenant un lissage du type Nagao et une dérivation par un masque de Prewitt [18] [22] [29]. Le gradient est calculé en chaque point mais on ne retient que le maximum du gradient dans la direction du vecteur gradient afin d'obtenir des contours fins.

Ce détecteur de contours donne une image binaire. Les points de contour n'étant pas toujours connexes, pour obtenir la segmentation en régions, nous procédons à une **fermeture des contours** en sélectionnant les points qui appartiennent au chemin de gradient cumulé maximal issu de l'extrémité d'un contour.

Ces traitements opèrent en aveugle dans le domaine de l'image, la segmentation obtenue ne peut donc pas être optimale par rapport à l'interprétation effectuée dans le domaine de la scène. Les échecs d'interprétation nous conduiront à revoir la segmentation des régions non classifiées.

2.2. DESCRIPTION DE L'IMAGE : GRAPHE D'ADJACENCE DES RÉGIONS

Après la détection et la fermeture des contours, un algorithme d'extraction des composantes connexes élabore une image des régions dans laquelle tous les pixels situés à l'intérieur d'un contour fermé reçoivent la même étiquette, les petites régions sont fusionnées. Nous utilisons une méthode [3] développée à partir des travaux de Rosenfeld [23]. L'image est parcourue de la droite vers la gauche et du haut vers le bas avec un motif en forme de « L ». Cet algorithme est très rapide et permet de fabriquer le **graphe d'adjacence des régions** en même temps (fig. 2).

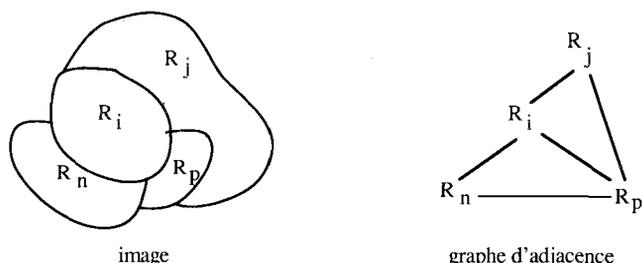


Figure 2. — Exemple de graphe d'adjacence.

Dans ce graphe, les nœuds représentent les régions et les arêtes les relations d'adjacence entre régions. Chaque région et chaque arête est caractérisée par un certain nombre d'attributs qui sont utilisés par le système d'interprétation [21] [26]. Les contours des régions sont approximatés par des segments de droite selon l'algorithme de Chebaro *et al.* [1]. A partir de ces segments de droite, qui ne sont pas nécessairement jointifs, on détermine le meilleur parallélogramme approximant la région.

Pour chaque région, sont calculés le niveau de gris moyen, la surface, le périmètre, la compacité ($4\pi \cdot \text{surface}/\text{périmètre}^2$), les coordonnées du centre de gravité, la direction de l'axe principal d'inertie, le pourcentage optimal d'approximation par un parallélogramme ($100 \cdot \text{surface}/(\text{surface meilleur parallélogramme approximant})$), la longueur, la largeur et l'élongation du meilleur rectangle approximant, le nombre et la liste des régions adjacentes. Les attributs d'une région **R** sont les composantes du vecteur attribut :

$$A_R = [a_1, \dots, a_M]^T.$$

Pour chaque adjacence, sont calculés la longueur et le gradient moyen de la frontière commune.

Le graphe d'adjacence est représenté par un ensemble d'assertions PROLOG exploitable aussi bien par le système-expert que par le simulateur de réseaux de neurones (planche 1). Les valeurs indiquées seront normalisées lors de leur exploitation (§ 3.2.2.1).

[région 19]

/* La région 19 est indiquée sur la figure 10 */

moyenne= 58 , surface= 33 ,

périmètre= 26 , compacité = 0,61 ,

approx. parallélogramme = 100 ,

élongation = 1,29

centre de gravité = (229,21 ; 5,4) ,

4 arêtes :

longueur = 5,	angle = 59
longueur = 5,	angle = 0
longueur = 7,	angle = -106
longueur = 5,	angle = 180

4 adjacences :

région 1	frontière = 6,	gradient = 7,75
région 24	frontière = 9,	gradient = 20,44
région 20	frontière = 5,	gradient = 3,40
région 18	frontière = 2,	gradient = 3,50

Planche 1. — Exemple d'attributs associés à une région du graphe d'adjacence.

2.3. SYSTÈME D'INTERPRÉTATION

Le problème d'interprétation d'images aériennes auquel nous étions confrontés, nous a conduit à imaginer un système simple permettant à un ensemble de modules de système-expert ou de réseaux de neurones de communiquer entre eux uniquement grâce à des assertions PROLOG (bases de données). On peut ainsi facilement interchanger des modules système-expert et des modules

réseau de neurones ou les faire fonctionner ensemble (fig. 3). Un gestionnaire de réseau permet même de distribuer les traitements sur des machines différentes et de les faire fonctionner en parallèle lorsque cela est possible [27].

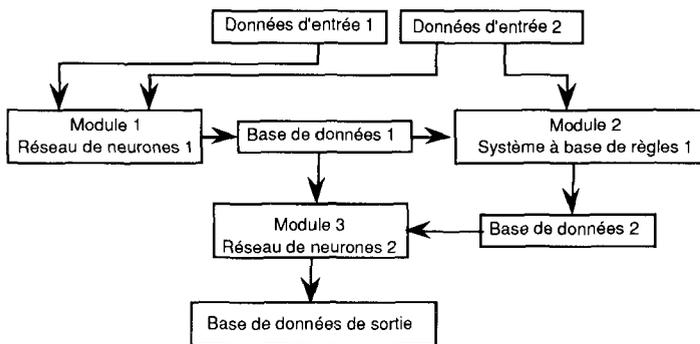


Figure 3. — Exemple de possibilités d'interconnexions entre modules de différents types fonctionnant avec des bases de données PROLOG.

Lors de l'**étiquetage initial**, un ensemble d'étiquettes possibles, dans le domaine de la scène, est assigné à chaque région en fonction de ses caractéristiques bas-niveau. Ces caractéristiques n'étant pas assez discriminantes, il est nécessaire de prendre en compte les relations topologiques et les connaissances du domaine de la scène ; un système de **propagation de contraintes** fabrique un étiquetage cohérent entre régions voisines (ex. : une face verticale au milieu de la pelouse, une ombre mal positionnée par rapport au soleil seront supprimées). L'interprétation utilise le graphe d'adjacence et des informations générales sur la scène permettant de réduire l'espace de recherche. Chacune des étapes de l'interprétation peut être réalisée par un système-expert ou par un réseau de neurones.

2.4. SYSTÈME MIXTE : RÉSEAU DE NEURONES/SYSTÈME-EXPERT

Comme nous l'avons déjà signalé, notre système d'interprétation comprend deux parties : un **classifieur** pour l'étiquetage initial et un **système à propagation de contraintes** pour l'étiquetage final. Nous en proposons deux réalisations : une basée sur des systèmes à base de règles et une employant des réseaux de neurones.

Pour les deux réalisations, le classifieur utilise le vecteur attribut $A_R = [a_1, \dots, a_M]$ de chaque région. Le système à propagation de contraintes exploite les caractéristiques des adjacences entre régions et des données globales sur l'image et sur la scène qui sont :

- position du soleil (approximative en degrés),
- position de la caméra (approximative en degrés),
- type d'objets que l'on peut rencontrer dans la scène (bâtiments, route, eau, ...),
- taille du pixel,
- relations entre objets dans la scène.

Les classes sont définies par un seuillage des composantes du vecteur attribut de chaque région. Le problème du choix des seuils est toujours difficile à résoudre. Ces seuils ne sont heureusement pas fondamentaux, il est même préférable de les laisser assez lâches pour autoriser un étiquetage multiple qui sera réduit par la propagation de contraintes. La réalisation connexionniste est une solution pour résoudre, entre autres, le problème du réglage des seuils.

Le système proposé est un **réseau de neurones structuré**, comme ceux de Feldman [4], qui comprend, en fait, deux réseaux multicouches de type propagation vers l'avant (« feed-forward »). L'un effectue une classification initiale, l'autre la propagation de contraintes, car nous verrons qu'un seul réseau ne permet pas de résoudre le problème de l'apprentissage de configurations contradictoires. L'implémentation connexionniste de la propagation de contraintes a présenté quelques difficultés mais la réalisation proposée est simple et cohérente avec la démarche de la propagation de contraintes classique.

3. Étude d'un classifieur pour l'étiquetage initial

Le classifieur à base de règles est très classique. Son intérêt est de nous donner des indications pour le dimensionnement du réseau.

3.1. ÉTIQUETAGE INITIAL PAR SYSTÈME-EXPERT

La classification est faite à partir de certaines composantes a_i du vecteur attribut A_R de chaque région R . A ces composantes (attributs) sont associés un ou plusieurs prédicats qui prennent les valeurs VRAI ou FAUX. Pour notre application, nous utilisons les attributs et les prédicats associés suivants :

- surface de la région R (en nombre de pixels) avec trois prédicats, *petit*(R), *moyen*(R) et *grand*(R),
- moyenne des niveaux de gris de R , avec quatre prédicats, *très_sombre*(R), *sombre*(R), *gris*(R) et *clair*(R),
- approximation par un parallélogramme, avec trois prédicats, *parallelo1*(R), *parallelo2*(R) et *parallelo3*(R),
- compacité, avec deux prédicats, *compact*(R) et *pas_compact*(R).

Les prédicats font appel à des seuils qui ont été fixés empiriquement après examen de quelques statistiques simples élaborées sur plusieurs images. Pour la surface par exemple, ils ont été fixés en fonction de la résolution de nos images. Quelques exemples sont donnés dans la planche 2.

Le système-expert affecte des étiquettes à une région R selon des règles de la forme :

SI *condition*(R) ALORS *étiqueter*(R).

Le terme *condition*(R) est une disjonction de conjonctions sur les prédicats ci-dessus (cf. planche 2).

Exemples d'assertions PROLOG introduisant les seuils:

```
valeur(grand,300).
valeur(apparal2,70).
valeur(sombre,80).
```

Exemples de règles PROLOG pour l'étiquetage initial:

```
/* la region R est grande si sa surface est superieure a "grand" */
grand(R) :-
```

```
  region(R,L_ATTRIBUT),
  nth0(2,L_ATTRIBUT,Surf),
  valeur(grand,V),
  Surf > V.
```

```
parallelo2(R) :-
```

```
  region(R,L_ATTRIBUT),
  nth0(5,L_ATTRIBUT,Apparal),
  valeur(apparal2,V),
  Apparal > V.
```

```
/* la region R est un batiment possible si elle est grande et
"parallelogramme" */
```

```
choix(R,batiment) :-
  parallelo2(R),
  grand(R).
```

```
/* la region R est un bosquet possible si elle est sombre et non
petite */
```

```
choix(R,bosquet) :-
  sombre(R),
  +\petit(R).
```

Planche 2. — Règles PROLOG utilisées pour l'étiquetage initial.

Les classes possibles sont au nombre de 6 : {bâtiment, pelouse, ombre, face verticale de bâtiment, arbre, bosquet}.

Les règles sont peu nombreuses (une vingtaine); les valeurs des seuils, fixées empiriquement ne sont pas critiques et autorisent un étiquetage multiple des régions.

3.2. ÉTIQUETAGE INITIAL PAR RÉSEAU DE NEURONES

La structure connectionniste bien adaptée pour réaliser l'étiquetage initial est celle d'un réseau de neurones multicouche de type perceptron avec une ou plusieurs couches cachées. Chaque neurone possède une fonction de sortie dérivable de type sigmoïde ce qui permet d'employer un apprentissage par rétropropagation du gradient [11] [12] [25]. La fonction d'entrée est de type affine : $e_i = \sum_{j=1}^N w_{ij} x_j - w_{i0} x_0$, où w_{ij} représente le poids

de la connexion liant l'entrée x_j au neurone i .

Plusieurs aspects doivent être étudiés pour la réalisation d'un tel réseau :

a) nature des entrées : binaires ou analogiques,

b) taille du vecteur d'entrée,

c) nombre de couches cachées et nombre de neurones par couche,

d) fabrication d'un ensemble d'exemples d'apprentissage (corrigé).

Les points b) et c) concernent le dimensionnement du réseau.

La taille du vecteur de sortie est égale au nombre de classes (ce sont les mêmes que pour le système-expert). Le vecteur de sortie est noté : $\mathbf{S} = [s_1, \dots, s_6]^T$; chaque s_i est associé à une classe et vaut +1 ou -1 suivant que l'étiquette est possible ou non pour la région.

Notre réalisation définitive est un réseau à entrées analogiques. Cependant dans un premier temps, nous présentons l'étude d'un réseau à entrées binaires car elle nous permet d'exposer un certain nombre de considérations sur le dimensionnement du réseau et sur l'apprentissage.

3.2.1. Un réseau de neurones multicouche avec des entrées binaires pour l'étiquetage initial

La réalisation en programmation logique du classifieur nous donne des indications précieuses pour le dimensionnement du réseau.

Il est facile de transformer une règle (en logique d'ordre 0) en un R.N. ayant trois couches qui peut réaliser n'importe quelle fonction logique. Par exemple, la figure 4 indique comment on peut réaliser un ET logique avec une couche de neurones.

Si région petite,
très_sombre,
compacte,
Alors la région peut être une ombre

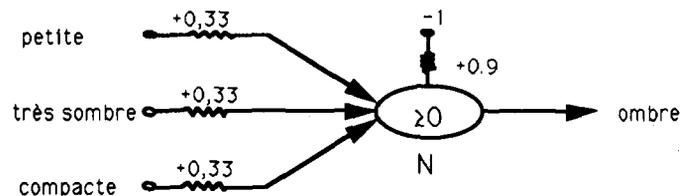


Figure 4. — Transposition d'une règle du système-expert sous forme neuronale.

3.2.1.1. Dimensionnement du réseau : taille du vecteur d'entrée et nombre de couches

Soit le vecteur d'entrée binaire $\mathbf{B} = [b_1, \dots, b_M]^T$. A chaque b_i correspond un prédicat du classifieur à base de règles (cf. § 3.1). Nous avons donc un vecteur d'entrée à 12 composantes pour chaque région; b_i vaut -1 ou +1 suivant que le prédicat associé est faux ou vrai (cf. planche 3). **La taille du vecteur d'entrée est donc égale au nombre de prédicats.**

Nous avons vu que la condition testée par une règle d'étiquetage était une disjonction de conjonctions. Par

	exemple 19	exemple 88	exemple 303	exemple 385	
très_sombre	-1	-1	-1	-1	
sombre	+1	+1	+1	+1	
gris	-1	-1	-1	-1	
clair	-1	-1	-1	-1	
parallelo1	+1	+1	+1	+1	
vecteur parallelo2	+1	+1	+1	+1	
d'entrée parallelo3	+1	+1	+1	+1	
binaire compact	+1	+1	+1	+1	
pas_comp.	-1	-1	-1	-1	
petit	-1	-1	-1	-1	fusion des sorties
moyen	+1	+1	+1	+1	
grand	-1	-1	-1	-1	
bâtiment	-1	-1	-1	+1	+1
pelouse	-1	-1	-1	-1	-1
vecteur ombre	-1	-1	-1	-1	-1
de sortie face_vert	-1	-1	-1	-1	-1
arbre	-1	-1	+1	-1	+1
bosquet	-1	+1	-1	-1	+1

Planche 3. — Exemples pour l'apprentissage du RN : les trois premiers appartiennent à l'image 1 (fig. 10), le dernier à l'image 2 (fig. 16).

exemple, une région R sera étiquetée ombre avec la règle suivante :

SI ((très_sombre(R), petite(R), compacte(R))
OU
(très_sombre(R), NON(petite(R)), NON(compacte(R))))
ALORS etiqueter(R, ombre)
(cf. fig. 5 a et 5 f).

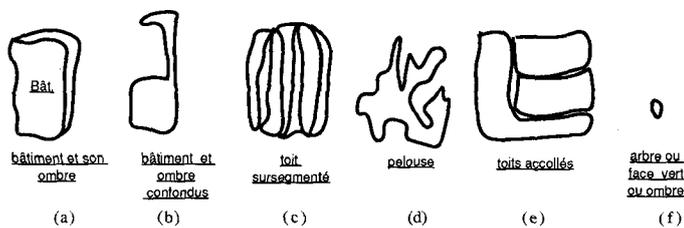


Figure 5. — Exemples d'objets à reconnaître.

Nous savons qu'il est aisé de simuler un ET logique ou un OU logique avec un neurone de type perceptron (cf. fig. 4). En conséquence, en cherchant à reproduire la structure des règles d'étiquetage utilisées en programmation logique, nous aurons besoin pour le réseau classifieur d'une seule couche cachée dont chaque neurone réalisera

un ET. Les 6 neurones de la couche de sortie réaliseront un OU. Cette organisation est en accord avec l'étude de Lippmann [13].

Il reste le dimensionnement de la couche cachée. Pour cela, nous utilisons le fait que, dans le classifieur à base de règles, une règle disjonctive utilise au maximum 3 conjonctions de prédicats. Nous affectons donc 3 neurones de la couche cachée à chaque neurone de sortie. Ce qui donne 18 neurones, plus un neurone à « -1 » pour pouvoir réaliser une fonction d'entrée affine pour chaque neurone de la couche de sortie. La couche cachée aura donc 19 neurones. Finalement, nous avons une structure 12-19-6.

Le classifieur à base de règles donne donc des informations très utiles sur la structure du R.N. (en particulier le nombre de neurones nécessaires sur chaque couche).

Cependant, le réseau de neurones offre une plus grande richesse que le système à base de règles. En effet, en connectant chaque neurone d'une couche à tous les neurones de la couche suivante, chaque neurone de sortie prend en compte toutes les composantes du vecteur d'entrée et les codages internes que représentent les états de la couche cachée. L'algorithme d'apprentissage, par l'ajustement des poids synaptiques, pondère l'influence de tous ces éléments. Ceci est difficile à réaliser explicitement en programmation logique.

3.2.1.2. Procédure d'apprentissage

Pour l'apprentissage, nous avons appliqué un algorithme classique de rétropropagation du gradient [11], [25]. Avant apprentissage, les poids des liaisons synaptiques sont initialisés aléatoirement avec des valeurs comprises entre -1 et +1. La fonction de sortie de tous les neurones est une sigmoïde de la forme :

$$\frac{e^{-ax} - 1}{e^{-bx} - 1}$$

L'apprentissage se fait à partir d'un « corrigé » C qui est un ensemble de L couples (vecteur d'entrée, vecteur de sortie) appelés exemples :

$$C = \{(B_1, S_1), \dots, (B_h, S_h), \dots, (B_L, S_L)\} .$$

L'apprentissage s'effectue en présentant un grand nombre de fois au réseau des exemples (B_h, S_h) choisis aléatoirement dans C. Au départ, le pas du gradient ϵ de l'algorithme de rétropropagation vaut 0,1, et nous le faisons décroître jusqu'à 0,001 ; le nombre de présentations d'exemples varie entre 10^5 et 10^6 .

3.2.1.3. Fabrication du corrigé

Dans un premier temps, nous avons fabriqué le corrigé de la manière suivante :

— l'expert humain étiquette un certain nombre de régions résultant de la segmentation d'images choisies comme référence (image 1, figure 9 ; image 2, figure 15). Les étiquettes appartiennent à l'ensemble déjà cité : {bâtiment, pelouse, ombre, face verticale de bâtiment, arbre, bosquet}. A peu près un tiers des régions ne sont pas étiquetées parce qu'elles sont petites ou difficilement

identifiables à cause de l'imperfection de la segmentation. A chaque région R_h , on associe donc, dans un premier temps, un vecteur de sortie S_h dont au plus une seule composante est à « + 1 », celle correspondant à la classe de la région R_h . Pour les régions non étiquetées, toutes les composantes du vecteur de sortie sont à « - 1 » ;

— à chaque région R_h , on associe un vecteur binaire d'entrée B_h en testant les 12 prédicats énoncés précédemment et appliqués sur le vecteur d'attributs A_h associé à la région R_h du graphe d'adjacence.

On constitue ainsi des exemples (B_h, S_h) .

Lorsqu'on effectue l'apprentissage sur ce premier type de corrigé, les résultats sont mauvais lorsqu'on fait travailler le réseau (après apprentissage bien sûr) sur l'une ou l'autre des images d'apprentissage. En effet, seulement 40 % environ des régions sont correctement étiquetées. Les raisons en sont les suivantes :

- a) une région non étiquetée peut avoir le même vecteur binaire qu'une région étiquetée ;
- b) deux régions étiquetées différemment peuvent avoir le même vecteur binaire.

Par exemple, sur la planche 3, les régions 19, 88, 303 de l'image 1 et la région 385 de l'image 2 sont étiquetées respectivement « inconnu », « bosquet », « arbre » et « bâtiment » dans le corrigé, or, elles ont le même vecteur binaire.

Ces deux phénomènes entraînent une non-convergence du réseau ou une convergence vers des états non significatifs et donc l'ajustement des poids s'effectue mal à cause des exemples contradictoires. Ils peuvent amener le réseau à ne pas étiqueter une région correctement étiquetée dans l'image d'apprentissage.

Pour résoudre ces problèmes, nous avons modifié l'élaboration du corrigé comme suit :

quand deux régions R_h et R_p produisant deux exemples (B_h, S_h) et (B_p, S_p) ont le même vecteur binaire ($B_h = B_p$), on conserve ces deux exemples, mais on remplace leur vecteur de sortie S_h et S_p par un même vecteur :

$$S_h^f = S_p^f = S_h \vee S_p$$

en faisant le OU logique composante à composante des deux vecteurs de sortie (l'exposant f symbolise cette opération de « fusion » des vecteurs).

On remplace donc les deux exemples (B_h, S_h) et (B_p, S_p) par les deux exemples (B_h, S_h^f) et (B_p, S_p^f) (cf. planche 3).

Ce procédé est appliqué itérativement sur toutes les régions et permet de fabriquer un nouveau corrigé comportant un **étiquetage multiple des régions**. Dans ce nouveau corrigé, il y a, par principe, des exemples identiques. C'est une manière d'accorder plus d'importance à certaines configurations lors de l'apprentissage.

Après cette phase, il reste quelques exemples dont les vecteurs de sortie ont toutes leurs composantes à « - 1 ». Nous avons choisi de les supprimer du corrigé appris car pour une même sortie (la classe « inconnu »), ils peuvent avoir des vecteurs d'entrée très différents. Deux autres raisons sont qu'il n'existe pas de critères bien définis pour

ne pas étiqueter une région et que ces exemples sont très peu nombreux.

Le corrigé pour l'étiquetage initial est finalement :

$$C^f = ((B_1, S_1^f), \dots, (B_h, S_h^f), \dots, (B_K, S_K^f)) .$$

3.2.2. Un réseau de neurones multicouche avec des entrées analogiques pour l'étiquetage initial

3.2.2.1. Dimensionnement du réseau : taille du vecteur d'entrée et structuration

Le réseau précédent donne des résultats satisfaisants, mais il est toujours tributaire des seuils de chaque attribut. Pour résoudre ce problème, nous avons décidé d'entrer directement les valeurs analogiques des attributs. Nous avons donc ajouté une couche supplémentaire en tête du réseau précédent afin d'opérer le seuillage, elle reçoit un vecteur d'entrée de la forme :

$$E = [e_1, \dots, e_i, \dots, e_N]^t .$$

Pour une région R donnée, e_i est une des composantes a_k du vecteur $A_R = [a_1, \dots, a_M]^t$ que l'on a normalisée. Nous avons retenu les attributs qui sont traités par les prédicats du classifieur à base de règles, il y en a 4 :

la surface, la moyenne des niveaux de gris, l'approximation par un parallélogramme, la compacité.

Nous ajoutons un neurone à « - 1 » pour le seuillage.

Nous avons choisi de « structurer » le réseau en limitant les connexions entre les neurones des deux premières couches. Ainsi, le problème du seuillage est simplifié si chaque neurone de la première couche cachée est seulement connecté à une entrée analogique et à l'entrée constante. Cela permet un meilleur contrôle du comportement de ces neurones. On garantit ainsi un apprentissage indépendant pour le seuillage de chaque entrée et on préserve la capacité du réseau à généraliser (on lui empêche d'apprendre une configuration de l'ensemble des entrées trop particulière). Nous avons adopté une structure régulière, c'est-à-dire que les neurones de la première couche cachée reçoivent le même nombre de connexions quelle que soit la grandeur analogique qu'ils traitent. Ce nombre est le nombre maximum de seuils appliqués à une entrée analogique dans le système d'étiquetage initial à base de règles (au maximum 4 seuils par attribut). Nous aboutissons ainsi au motif de la figure 6.

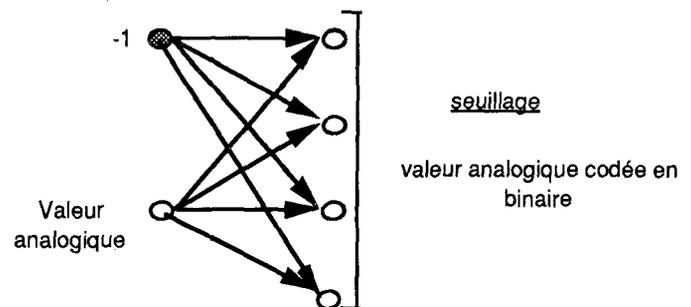


Figure 6. — Structure de la première couche cachée du réseau pour un paramètre d'entrée.

Chaque neurone de la première couche est ainsi relié à 4 neurones de la deuxième couche. La première couche cachée comprend donc 16 neurones traitant les données analogiques, plus un neurone à « -1 » pour la couche suivante, ce qui fait en tout 17 neurones. Les deux dernières couches du réseau sont inchangées, ce qui nous conduit à un réseau 5-17-19-6, représenté figure 7.

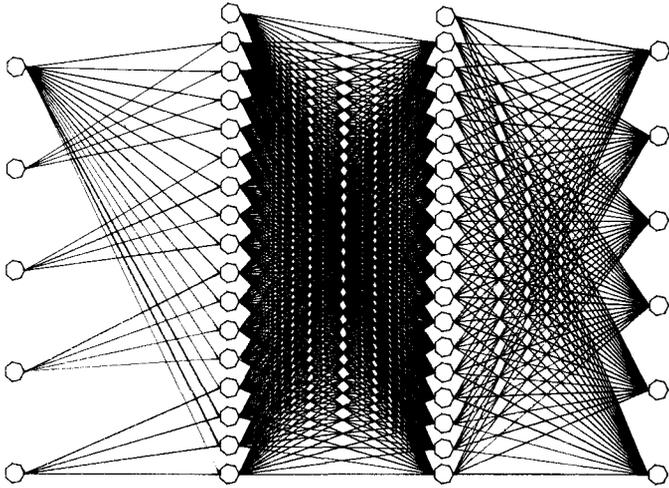


Figure 7. — Structure du réseau de neurones pour l'étiquetage initial.

3.2.2.2. Fabrication du corrigé du réseau à entrées analogiques.

Pour le réseau avec entrées analogiques, le corrigé est un ensemble d'exemples :

$$C^e = \{ (E_1, S_1), \dots, (E_h, S_h), \dots, (E_L, S_L) \} .$$

Dans un exemple (E_h, S_h) , E_h est un vecteur à composantes analogiques normalisées et S_h est un vecteur de sortie. Si S_h correspond à une région reconnue par l'expert, il possède une seule composante à « +1 » ; s'il correspond à une région non identifiée, toutes ses composantes sont à « -1 ».

Si nous utilisons directement ce type de corrigé, nous obtenons de mauvais résultats pour pratiquement les mêmes raisons que pour le réseau à entrées binaires :

- a) une région non étiquetée peut avoir un vecteur analogique voisin de celui d'une région étiquetée ;
- b) deux régions étiquetées différemment peuvent avoir des vecteurs analogiques voisins.

Ceci pose de nouveau le problème de l'étiquetage multiple. Nous l'avons résolu en associant à chaque vecteur analogique E_h , un vecteur binaire B_h élaboré avec les seuils employés pour le réseau à entrées binaires. Donc à partir du corrigé :

$$C^f = \{ (B_1, S_1^f), \dots, (B_h, S_h^f), \dots, (B_K, S_K^f) \}$$

nous avons construit le corrigé du réseau avec entrées analogiques :

$$C^{fe} = \{ (E_1, S_1^f), \dots, (E_h, S_h^f), \dots, (E_K, S_K^f) \} .$$

Dans le cas où l'on veut réaliser directement un réseau à entrées analogiques, il est nécessaire pour résoudre les problèmes a) et b) de créer arbitrairement des classes définies par un seuillage sur les valeurs analogiques. Les seuils peuvent être choisis de manière à découper le domaine de variation d'une donnée analogique en intervalles égaux. L'algorithme d'apprentissage ajustera ensuite les poids du réseau afin d'optimiser les valeurs des seuils.

3.3. RÉSULTATS EXPÉRIMENTAUX POUR L'ÉTIQUETAGE INITIAL

Nous avons utilisé deux images d'apprentissage (fig. 9 et 15), qui sont des images aériennes de dimension 256×256 et pour lesquelles un pixel représente un carré de 1 m de côté. L'image 1 présente des bâtiments isolés, de forme rectangulaire, induisant chacun une ombre ; un bâtiment dans le bas de l'image possède un toit texturé. L'image 2 représente une zone industrielle où les bâtiments se jouxtent, ils n'ont donc pas tous une ombre ; ils sont de formes beaucoup plus irrégulières et de luminance très variable. La segmentation de la première donne 358 régions (fig. 10), celle de la seconde 458 régions. Sur un total de 816 régions, l'opérateur humain en interprète 540 (fig. 11 et 16), ce qui représente environ 75 % des pixels. Mais dans certains cas, nous ne sommes pas sûrs que notre interprétation soit la meilleure !

Le corrigé pour l'étiquetage initial comporte 804 exemples (seules 12 régions n'ont pas d'étiquette après l'étape de fusion).

Le réseau se stabilise après environ un million de présentations d'exemples, ce qui prend plusieurs heures sur une station SUN/4. Le taux de régions bien classifiées (par rapport au corrigé fabriqué pour l'étiquetage initial) est de 86 %.

Nous avons aussi testé un réseau à trois couches avec entrées analogiques et d'autres réseaux à quatre couches, comportant notamment moins de connexions entre la troisième et la dernière couche (un neurone de la dernière couche relié seulement à 4 neurones de la couche précédente par exemple), mais le taux de succès n'est pas aussi bon et la convergence est plus longue.

Le réseau est d'abord testé sur les deux images d'apprentissage puis il est appliqué à une nouvelle image (fig. 19) de même résolution et comportant des bâtiments isolés. Les résultats de cet étiquetage initial sont donnés par les figures 12, 17 et 20. Les régions étiquetées « inconnu » sont celles qui ont reçu zéro ou plusieurs étiquettes. Les taux de succès par rapport aux corrigés sont donnés par les planches 4, 6 et 8 (nous avons donc fabriqué dans ce but le corrigé de l'image 3). Ils sont calculés, pour chaque classe, comme le rapport du nombre de régions étiquetées uniquement avec cette classe sur le nombre de régions de cette classe dans le corrigé. Seules sont prises en compte les régions traitées par l'expert humain. Après une analyse fine des résultats, il apparaît que certaines régions non reconnues par l'expert ont été correctement étiquetées par le système. Ces régions n'interviennent pas dans le calcul des taux de succès. Les régions comportant un étiquetage multiple sont considérées comme fausses par rapport au

corrigé, ce qui explique les faibles taux de succès obtenus. De plus, le taux de pixels bien étiquetés est très nettement supérieur au taux de régions bien étiquetées que nous donnons (les grandes régions étant dans leur grande majorité bien reconnues).

Compte-tenu de ces restrictions, on constate que le réseau étiquette en général correctement et avec une seule étiquette les bâtiments, la pelouse et certaines ombres. Par contre, on a très souvent un étiquetage multiple pour les arbres, les bosquets, les faces verticales des bâtiments et certaines ombres. En effet, il n'existe pas de paramètres vraiment discriminants. Cet étiquetage multiple sera traité par la propagation de contraintes.

A titre de comparaison, nous présentons également le résultat de l'étiquetage initial par le système-expert sur l'image 1 (fig. 13 et planche 4) et sur l'image 3 (planche 8). Les résultats de ces deux étiquetages initiaux sont assez proches, le réseau de neurones donne toutefois plus d'étiquettes « bâtiment » que le système-expert.

4. Propagation de contraintes

A la fin de l'étiquetage initial, qu'il soit fait par le système-expert ou le réseau de neurones, chaque région possède zéro, une ou plusieurs étiquettes. Le problème est d'affiner cet étiquetage et d'affecter une seule étiquette à chaque région en tenant compte de l'étiquetage des régions voisines. C'est le rôle de la propagation de contraintes.

4.1. PROPAGATION DE CONTRAINTES PAR SYSTÈME-EXPERT

L'algorithme utilisé est dérivé de celui présenté par N. C. Rowe dans « Relaxation method » [24]. Les contraintes d'adjacence sont exprimées par des prédicats impliquant deux régions.

Dans un premier temps, les étiquettes attribuées à une région qui ne sont pas cohérentes avec les étiquettes des régions adjacentes sont supprimées (**contraintes d'interdiction**). On utilise des contraintes d'inclusion telles que :

- un arbre ne peut pas être inclus dans un bâtiment,
- une ombre ne peut pas être incluse dans de la pelouse, etc...

et des contraintes d'adjacence telles que :

- une ombre
 - doit être adjacente à une structure susceptible de faire ombre (un bâtiment, un arbre, un bosquet ou une région non étiquetée) située entre elle et le soleil,
 - doit respecter certaines contraintes concernant sa taille par rapport à cette structure, la longueur de la frontière commune par rapport aux périmètres, etc...

Puis, pour étiqueter les régions qui n'ont plus d'étiquette, on se sert de **contraintes d'obligation**. Cette étape a pour but de forcer l'étiquetage, soit en utilisant les critères d'adjacence, soit en employant des seuils plus précis. Une région reçoit (si possible) une étiquette unique qui tient compte des interprétations des régions qui l'entourent. Par exemple, si une région non étiquetée est adjacente à une région étiquetée face_verticale et si elle se trouve du bon côté par rapport à cette face verticale (en fonction de la position de la caméra) alors la région est étiquetée bâtiment.

L'algorithme utilisé est le suivant :

Soit $\{R_1, \dots, R_n\}$ l'ensemble des régions et $\{e_1, \dots, e_m\}$ l'ensemble des étiquettes. La liste des régions adjacentes à R_i est notée LA_i et la liste des étiquettes possibles pour R_i est notée LL_i . Par exemple sur la figure 2, la région R_j appartient à LA_i .

Les différentes étapes de l'algorithme sont :

1. Marquer chaque région « active »
2. Tant qu'il reste des régions actives faire :
Choisir une région active R_i .
Pour chaque étiquette l_j appartenant à LL_i ,
si l_j n'est consistante avec aucune étiquette d'aucune région adjacente appartenant à LA_i (en utilisant les contraintes d'interdiction)
supprimer l_j de la liste LL_i ,
marquer active chaque région de LA_i qui est inactive.
Marquer R_i « inactive ».
3. Marquer « active » toutes les régions qui n'ont plus aucune étiquette ou qui en ont plusieurs après l'étape 2.
4. Tant qu'il reste des régions actives faire :
Choisir une région active R_i .
Affecter à R_i une étiquette si les contraintes d'obligation sont satisfaites.
Marquer R_i « inactive ».
5. Donner l'étiquetage final.

A l'initialisation, toutes les régions du graphe d'adjacence sont marquées « active ».

La boucle 2 de l'algorithme se termine toujours. En effet :

- une région examinée est systématiquement rendue « inactive » ;

- une région est « réactivée » si elle est « inactive » et si elle est adjacente à une région dont la liste d'étiquettes a été modifiée. Comme le traitement de cette boucle n'ajoute pas d'étiquette, quand les contraintes ne modifient plus l'étiquetage des régions, il n'y a plus d'activation de région et la boucle se termine.

En pratique, à la fin de cet algorithme, il subsiste peu de cas d'étiquetage multiple ou d'étiquetage « vide ».

4.2. UN RÉSEAU DE NEURONES POUR LA PROPAGATION DE CONTRAINTES

L'étude d'un réseau de neurones pour la propagation de contraintes n'est pas immédiate. Les problèmes sont liés au dimensionnement du vecteur d'entrée, à l'élaboration du corrigé, au fonctionnement itératif qui est calqué sur celui réalisé en programmation logique. Nous avons conservé une structure de réseau multicouche avec un apprentissage par rétropropagation du gradient. Nous présentons dans les paragraphes suivants l'étude de ce deuxième réseau noté RN2 (fig. 8).

4.2.1. Vecteurs d'entrée et de sortie, élaboration du corrigé

La propagation de contraintes utilise les attributs d'une région, les attributs des régions adjacentes et les attributs des adjacences. Un des problèmes est le dimensionnement du vecteur d'entrée car le nombre de régions adjacentes est variable. Pour prendre en compte explicitement toutes les données il faudrait un vecteur d'entrée dont la taille serait égale à : (nombre maximal de régions adjacentes + 1) × nombre d'attributs. De plus, cela introduirait un problème pour la généralisation de l'apprentissage : un nombre prohibitif d'exemples serait nécessaire pour que l'apprentissage soit fiable et complet (le nombre des configurations d'adjacence varie exponentiellement avec le nombre d'adjacences).

Nous avons donc choisi de fabriquer un vecteur d'entrée P_h composé de trois parties :

— les interprétations de la région étudiée, S_h

— les interprétations des régions adjacentes fusionnées par un OU logique $(\bigcup_j S_j)$ comme exposé dans le paragraphe 3.2.1.3,

— les paramètres relatifs aux contraintes d'adjacence et également fusionnés par un OU logique $(\bigcup_j F_{hj})$.

Le vecteur binaire F_{hj} caractérise la cohérence d'étiquetage entre deux régions adjacentes R_h et R_j .

Il possède 8 composantes qui traduisent les possibilités d'étiquetage de R_h en fonction des étiquettes de R_j (on utilise les mêmes contraintes que pour le système expert) : ombre possible, face verticale possible, bâtiment possible, bosquet possible, ombre impossible, face verticale impossible, bâtiment impossible, bosquet impossible.

Les unions logiques des étiquettes des régions adjacentes et des contraintes d'adjacence sont suffisantes puisqu'il suffit qu'une étiquette soit consistante avec l'une des étiquettes d'une des régions adjacentes pour qu'elle soit conservée (cf. § 4.1).

Par ailleurs, le réseau doit savoir traiter les étiquetages multiples, les cas d'étiquetage unique et de non-étiquetage. On doit donc lui faire apprendre d'une part des exemples (P_h, S_h) issus du corrigé C (cf. § 3.2.1.1) fabriqué par l'expert et qui comportent, au plus, une étiquette par région et d'autre part des exemples (P'_h, S_h) issus du corrigé pour l'étiquetage initial C^I (cf. § 3.2.1.3) et qui peuvent comprendre plusieurs étiquettes. Le corrigé est

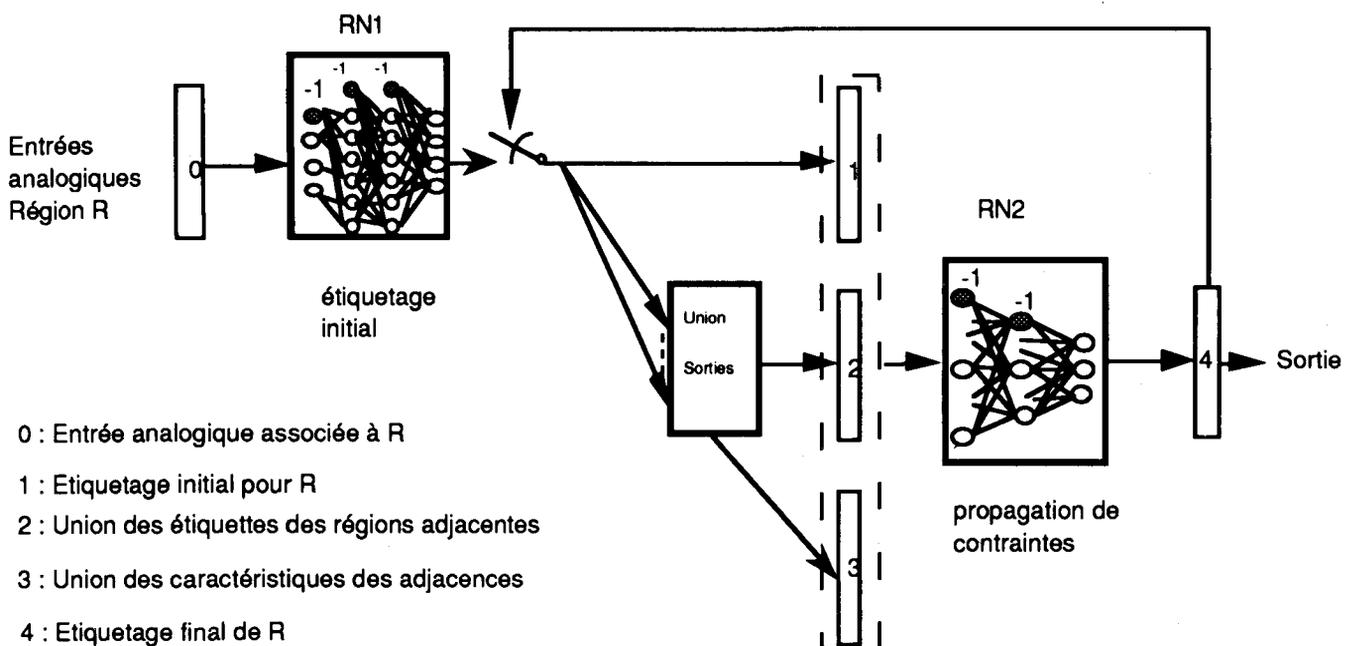


Figure 8. — Un système de réseaux de neurones pour l'interprétation d'images.

donc de la forme :

$$C^p = \{ (P_1, S_1), \dots, (P_h, S_h), \dots, (P_K, S_K), \dots, (P'_1, S_1), \dots, (P'_h, S_h), \dots, (P'_L, S_L) \}$$

avec :

$$P_h = \begin{bmatrix} S_h \\ \bigcup_j S_j \\ \bigcup_j F_{hj} \end{bmatrix} \quad \text{et} \quad P'_h = \begin{bmatrix} S'_h \\ \bigcup_j S'_j \\ \bigcup_j F'_{hj} \end{bmatrix}.$$

4.2.2. Apprentissage

La structure du réseau de neurones a été déterminée empiriquement car il est difficile de trouver des critères à partir des règles du système-expert.

Il comporte une couche d'entrée de 20 (6 + 6 + 8) neurones plus 1 neurone à « - 1 » pour le seuil et une couche de sortie de 6 neurones (les 6 étiquettes possibles). Différents essais ont été faits avec zéro et une couche cachée, les meilleurs résultats ont été obtenus avec une couche cachée de 21 neurones plus 1 neurone à « - 1 » (87 % des régions correctement classifiées par rapport au corrigé de l'expert).

4.2.3. Fonctionnement de la propagation de contraintes par réseau de neurones

Pour chaque région R_h de l'image à interpréter, dans un premier temps, le deuxième réseau de neurones reçoit en entrée le vecteur d'entrée P_h^1 construit à partir des résultats de l'étiquetage initial :

$$P_h^1 = \begin{bmatrix} S_h^0 \\ \bigcup_j S_j^0 \\ \bigcup_j F_{hj}^0 \end{bmatrix}.$$

(Les composantes de ces vecteurs sont fabriquées comme indiqué § 4.2.1).

Il fournit un vecteur de sortie S_h^1 .

Dans un deuxième temps, il doit « propager » des contraintes d'étiquetage, ce qui ne peut se faire en une seule itération. La sortie est alors réinjectée vers l'entrée. C'est ce qu'indique l'interrupteur de la figure 8. Pour l'itération n , on fabrique le vecteur d'entrée P_h^n à partir des vecteurs de sortie S_h^{n-1} de la région R_h et S_j^{n-1} des régions adjacentes R_j :

$$P_h^n = \begin{bmatrix} S_h^{n-1} \\ \bigcup_j S_j^{n-1} \\ \bigcup_j F_{hj}^{n-1} \end{bmatrix}.$$

Comme le réseau a appris des exemples issus du corrigé C, un bon état de sortie (par exemple égal au corrigé) sera conservé si on le réinjecte en entrée. Le réseau RN2 réagit comme une mémoire **pseudo-associative**.

4.3. RÉSULTATS

Les différentes simulations réalisées montrent que les performances du deuxième réseau RN2 ne dépendent pas trop des résultats de l'étiquetage initial (il compense les erreurs commises par le premier réseau en se servant des informations relatives aux régions adjacentes). En fait, il n'est pas très intéressant d'obliger le premier réseau RN1 (classifieur) à résoudre de nombreux cas particuliers car d'une part cela nécessiterait un réseau plus complexe et d'autre part, on prendrait le risque de lui faire apprendre uniquement des cas particuliers, ce qui pourrait provoquer des résultats aberrants pour des types de régions qu'il n'a jamais rencontrés. Le problème est que le réseau RN2 nécessite un apprentissage important et que l'on ne peut jamais être sûr d'avoir présenté tous les types de contraintes possibles.

Les résultats du système mixte (réseau de neurones pour l'étiquetage initial suivi de propagation de contraintes par relaxation) sont donnés figures 14, 18 et 21 ainsi que planches 5, 7 et 9. Les taux de succès sont calculés comme dans le paragraphe 3.3 en pourcentage de régions et de pixels bien étiquetés par rapport au corrigé. Sur les images ayant servi à l'apprentissage, le pourcentage de pixels étiquetés comme dans le corrigé est de 85 % et 86 %. Il est de 69 % pour l'image 3 qui n'a pas servi à l'apprentissage du réseau de neurones. Les bâtiments sont dans l'ensemble bien reconnus. Les problèmes rencontrés sur l'image 3 sont dus au fait que certaines régions ont des formes qui n'avaient jamais été rencontrées dans les images d'apprentissage (les ombres et les bosquets sont nettement plus grands et souvent confondus).

A titre de comparaison, nous donnons également les taux de succès du système complet à base de règles (étiquetage initial et propagation de contraintes) sur les images 1 et 3. Les deux systèmes donnent sensiblement les mêmes résultats (80 % contre 85 % de succès pour l'image 1, 71 % contre 69 % pour l'image 3). L'étiquetage multiple fourni par le réseau de neurones sur l'image n'ayant pas servi à l'apprentissage est donc équivalent à celui obtenu par le système-expert, la suppression des étiquettes non cohérentes avec l'environnement donnant à peu près les mêmes taux de succès.

Le système-expert utilise des règles strictes pour l'interprétation d'une région. Par contre le réseau de neurones, de par la structure des connexions entre la deuxième et la troisième couche (cf. § 3.2.1.1), prend mieux en compte les composantes du vecteur d'entrée. De ce fait, il se comporte mieux sur des régions moins « typées » et est moins tributaire de la forme d'une région ou de la présence de texture qui conduisent à des erreurs de segmentation. Par exemple, il reconnaît des bâtiments aux formes irrégulières.

Par ailleurs, puisque les règles du système-expert pour la propagation de contraintes ne sont pas trop compliquées

et que le cadre d'utilisation est bien défini, il nous est apparu plus judicieux de continuer à réaliser la propagation de contraintes avec un système à base de règles.

5. Conclusion

Nous avons présenté un système complet d'analyse d'image, sur un cas réel et complexe. Ce système intègre toute la chaîne des traitements, depuis le bas-niveau (segmentation en régions) jusqu'à l'interprétation.

Le choix d'un système mixte réseau de neurones (pour l'étiquetage initial multiple)/système-expert (pour la propagation de contraintes) se justifie par les trois avantages suivants :

- la détermination automatique des seuils pour l'étiquetage initial,
- la simplicité des règles du système-expert pour la propagation de contraintes qui ne dépendent pas de l'image ni de sa résolution,
- la robustesse aux erreurs de segmentation.

Nous avons montré qu'il était possible de réaliser un système d'apprentissage complexe pour la reconnaissance des formes. Les réseaux de neurones utilisant la rétropropagation du gradient représentent une solution attractive pour résoudre certains problèmes de reconnaissance des formes car ils sont robustes, adaptables, ils supportent des erreurs lors de l'apprentissage et travaillent en général beaucoup plus vite que des systèmes à bases de règles (ils peuvent être très facilement utilisés dans des architectures massivement parallèles). Ce sont des systèmes dans lesquels on peut facilement ajouter des connaissances et enfin c'est sûrement le moyen le plus simple de réaliser un système d'apprentissage interactif dédié à des utilisateurs non spécialisés en informatique et qui voudraient faire apprendre de nouveaux cas à leur système (corriger les erreurs apparues en phase d'utilisation et intégrer les corrections pour la suite de l'utilisation). Quant aux modules à base de règles, ils sont très utiles chaque fois que l'univers analysé est bien défini.

La réalisation mixte assure le contrôle du fonctionnement du réseau de neurones par un système à base de règles.

Manuscrit reçu le 4 juillet 1991.

BIBLIOGRAPHIE

- [1] B. CHEBARO, L. MASSIP-PAILHES, S. CASTAN, « Description polygonale basée sur l'extraction d'îlots de confiance », 8e RFIA, Villeurbanne, novembre 1991.
- [2] R. DERICHE, J.-P. COCQUEREZ, « Extraction de composantes connexes basée sur une détection optimale des contours », MARI 87, Paris, mai 1987.
- [3] R. DERICHE, J.-P. COCQUEREZ, G. ALMOUZNI, « An efficient method to built early image description », 9th International Conference on Pattern Recognition, Rome, July 1988.
- [4] J. A. FELDMAN, « Connectionist Models and Parallelism in High Level Vision », CVGIP 31, 1985, p. 178-200.
- [5] M. GUTKNECHT, R. PFEIFER, « An approach to integrating expert systems with connectionist network », AICOM, Vol. 3, n° 3, September 1990, p. 116-127.
- [6] D. V. HILLMAN, « Integrating neural nets and expert systems », *IA expert*, June 1990, p. 54-59.
- [7] A. HUERTAS, W. COLE, R. NEVATIA, « Detecting runway in complex airport scenes », CVGIP 51, 107-145, 1990.
- [8] A. HUERTAS, R. NEVATIA, « Detecting buildings in aerial images », CVGIP, 41, 131-152, 1988.
- [9] R. B. IRVIN, D. McKEOWN, « Methods for exploiting the relationship between buildings and their shadows in aerial imagery », *IEEE Trans. on SMC*, Vol. 19, n° 6, november/december 1989.
- [10] M. JORDAN, « Analyse stéréoscopique de vues aériennes. Elaboration d'une description volumique des scènes ». Thèse de Doctorat, Université Paris 11, 1992.
- [11] Y. LE CUN, « Modèles connexionistes de l'apprentissage », Thèse de Doctorat, Université Paris 6, 1987.
- [12] Y. LE CUN, B. BOSER, J. S. DENKER, D. HENDERSON, R. E. HOWARD, W. HUBBARD, L. D. JACKEL, « Handwritten digit recognition with a back-propagation network », In David Touretzky ed., *Neural Information Processing Systems*, Vol. 2, Denver, 1989-1990, Morgan Kaufman.
- [13] R. P. LIPPMANN, « An introduction to computing with neural nets », *IEEE Trans. on ASSP*, 4-22, April 1987.
- [14] D. McKEOWN, W. A. HARVEY, J. McDERMOTT, « Rule-based interpretation of aerial imagery », *IEEE Trans. on PAMI*, Vol. 7, n° 5, September 1985.
- [15] D. McKEOWN, « Building knowledge-based systems for detecting man-made structures from remotely sensed imagery », *Phil. Trans. R. Soc. Lond. A* 324, 423-435, 1988.
- [16] T. MATSUYAMA, « Expert systems for image processing : knowledge-based composition of image analysis processes », CVGIP 48, October 1989, p. 22-49.
- [17] R. MOHAN, R. NEVATIA, « Using perceptual organization to extract 3D structures », *IEEE Trans. on PAMI*, Vol. 11, n° 11, p. 1121-1139, November 1989.
- [18] M. NAGAO, T. MATSUYAMA, « Edge preserving smoothing », CVGIP 9, 1979, p. 394-407.
- [19] M. NAGAO, T. MATSUYAMA, Y. IKEDA, « Region extraction and shape analysis in aerial photographs », CVGIP 10, p. 195-223, 1979, Academic Press.
- [20] R. NEVATIA, K. PRICE, « Locating structures in aerial images », *IEEE Trans. on PAMI*, Vol. PAMI-4, n° 5, September 1982.
- [21] Y. OHTA, « Knowledge-based interpretation of outdoor natural color scenes », ed. Pitman, 1985.
- [22] J. M. S. PREWITT, « Object enhancement and extraction, Picture processing and Psychopictories », ed. B. S. Lipkin, A. Rosenfeld, Academic Press, N.Y. 1970.
- [23] A. ROSENFELD, J.-L. PFALTZ, « Sequential operations on digital image processing », *JACM*, 13, 4, p. 471-494, 1966.
- [24] N. C. ROWE, « Artificial intelligence through PROLOG », Prentice-Hall, 1988.
- [25] D. E. RUMELHART, G. E. HINTON, R. J. WILLIAMS, « Learning internal representations by error propagation ». *Parallel Distributed Processing : Explorations in the Microstructures of Cognition*, Vol. 1, 1986, Cambridge, MA : MIT Press.
- [26] Y. SHIRAI, « Three-dimensional computer vision », ed. Springer-Verlag, 1987.
- [27] R. THIBERT, J.-P. COCQUEREZ, M. LECLERC, « Langage graphique dédié au traitement d'images permettant d'exploiter la puissance et les ressources réparties », 8e congrès RFIA, Lyon, novembre 1991.
- [28] M. ZAHID, « Stéréovision en imagerie aérienne : reconstruction tridimensionnelle des bâtiments », Thèse de doctorat, Université Paris 11, avril 1992.
- [29] Rapport Segmentation, 1re partie, GDR 134, CNRS, décembre 1991.

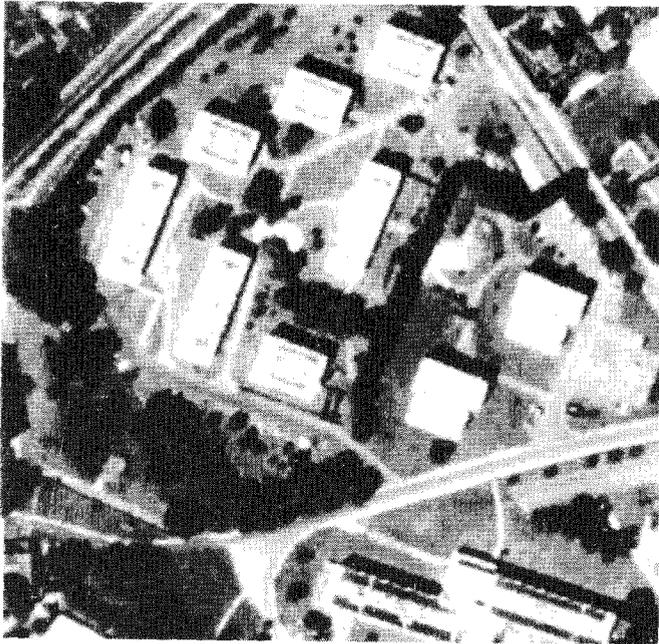


Fig. 9. — Image d'apprentissage 1.

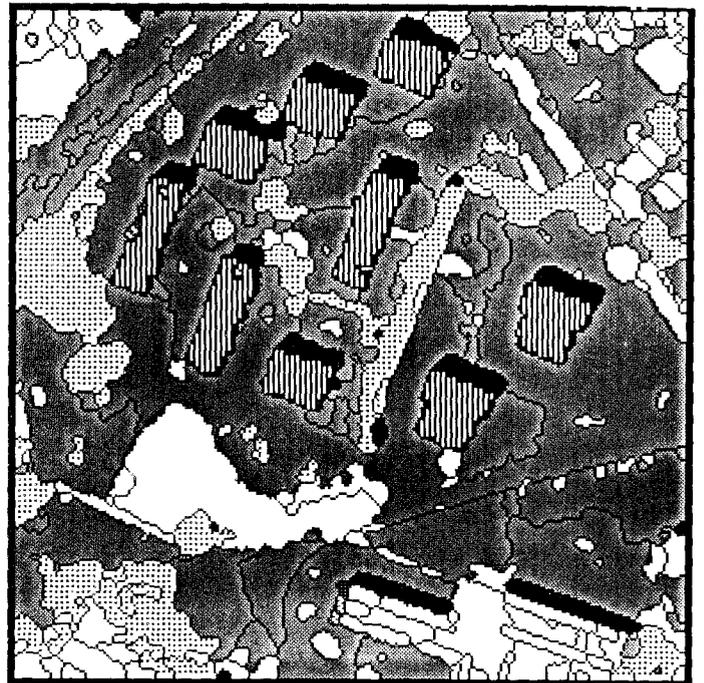


Fig. 11. — Corrigé de l'image 1.

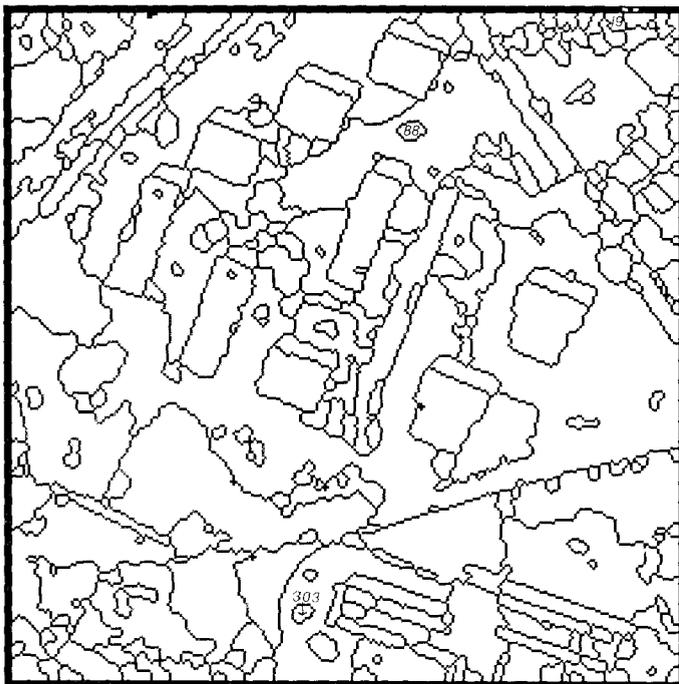


Fig. 10. — Image 1 segmentée.

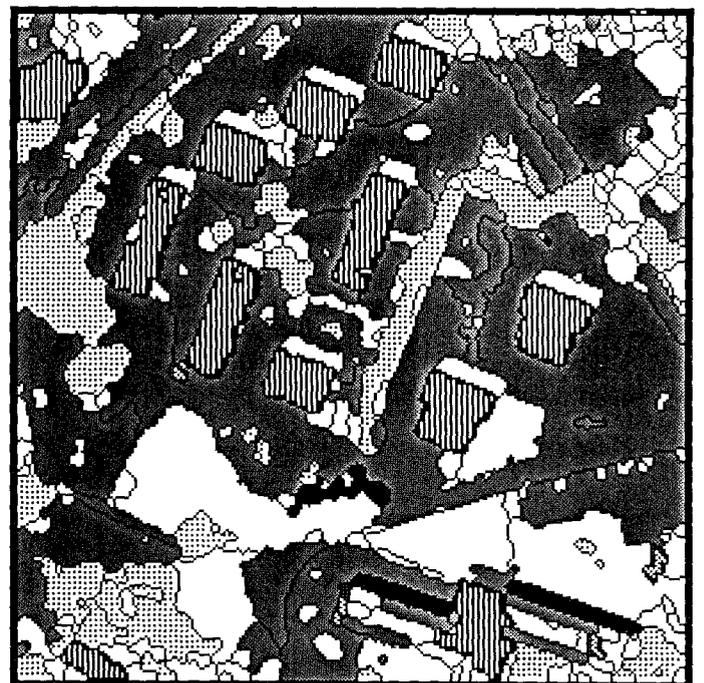
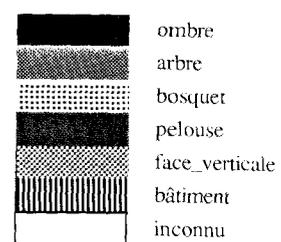


Fig. 12. — Etiquetage initial de l'image 1 par R. N.



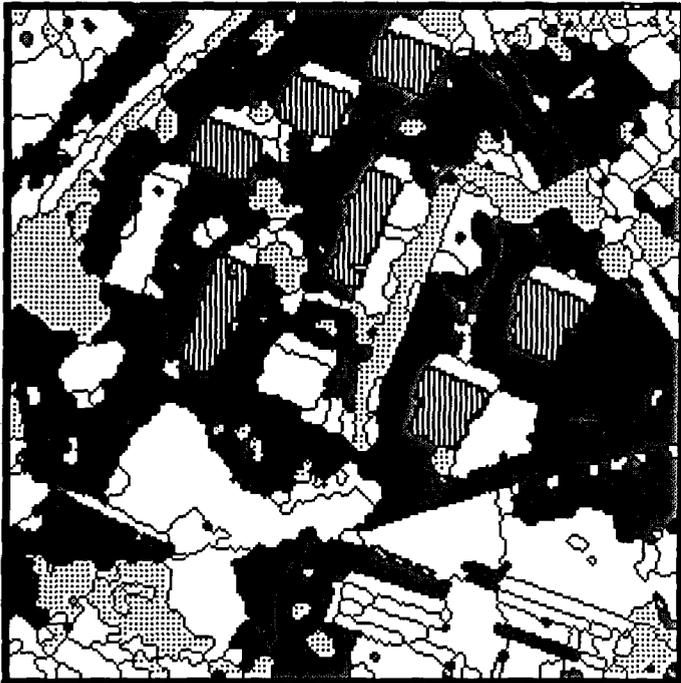


Fig. 13. — Etiquetage initial de l'image 1 par système-expert.

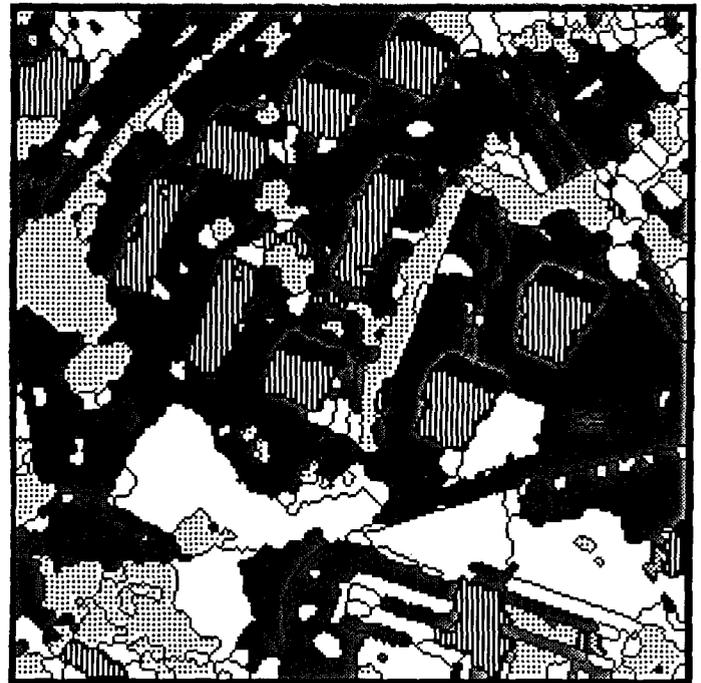


Fig. 14. — Etiquetage final de l'image 1 par le système mixte.

classes	nombre de régions dans le corrigé	étiquetage initial du réseau		étiquetage initial système expert	
		nombre de régions	pourcentage	nombre de régions	pourcentage
bâtiment	9	9	100%	7	77%
ombre	20	2	10%	8	40%
face verticale	3	1	33%	1	33%
arbre	28	18	64%	18	64%
bosquet	52	36	69%	33	63%
pelouse	110	31	28%	45	41%
TOTAL	222	97	43%	112	50%

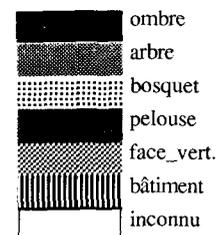


Planche 4. — Résultats de l'étiquetage initial, par rapport au corrigé, de l'image 1.

classes	nombre de régions dans le corrigé	étiquetage système mixte		étiquetage système-expert	
		nombre de régions	pourcentage	nombre de régions	pourcentage
bâtiment	9	9	100%	9	100%
ombre	20	10	50%	12	60%
face verticale	3	0	0%	0	0%
arbre	28	19	68%	18	64%
bosquet	52	43	83%	46	88%
pelouse	110	54	49%	49	44%
TOTAL	222	135	61%	134	60%
nbre de régions					
pourcentage de pixels bien étiquetés par rapport au corrigé			85%		80%

Planche 5. — Résultats de l'étiquetage final de l'image 1.

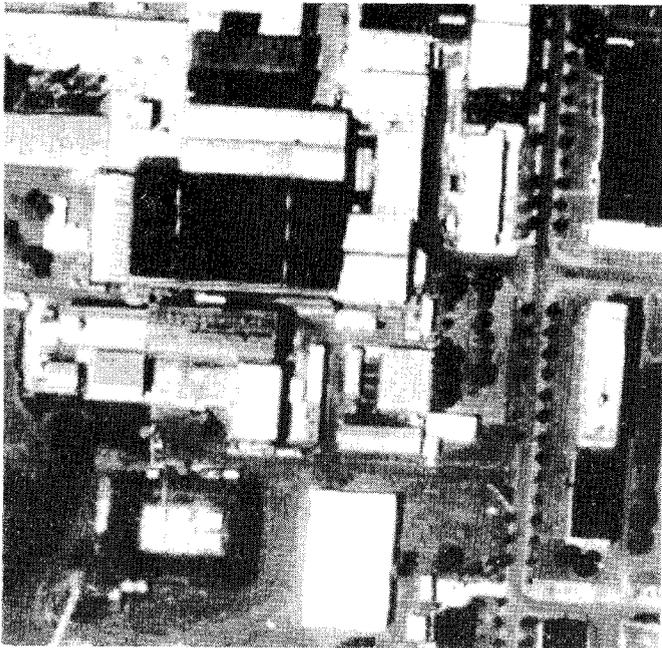


Fig. 15. — Image d'apprentissage 2.



Fig. 16. — Corrigé de l'image 2.

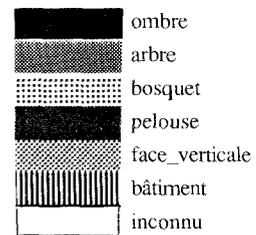


Fig. 17. — Etiquetage initial de l'image 2 par R. N.

Classes	nombre de régions dans le corrigé	étiquetage initial du réseau	
		nombre de régions	pourcentage
bâtiment	22	17	77%
ombre	27	8	30%
face verticale	15	1	7%
arbre	36	28	78%
bosquet	26	14	54%
pelouse	100	39	20%
TOTAL	318	107	33%

Planche 6. — Résultats de l'étiquetage initial sur l'image 2.

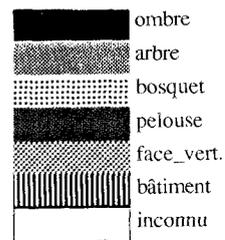
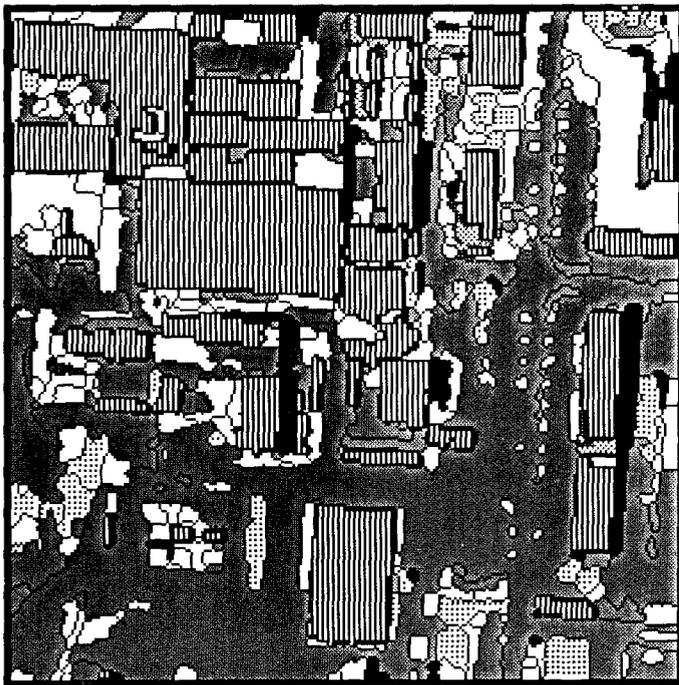


Fig. 18. — Etiquetage final de l'image 2 par le système mixte.

classes	nombre de régions dans le corrigé	étiquetage système mixte	
		nombre de régions	pourcentage
bâtiment	22	19	86%
ombre	27	10	37%
face verticale	15	1	6%
arbre	36	32	89%
bosquet	26	17	65%
pelouse	100	100	52%
TOTAL	318	179	56%
nbre de régions			
pourcentage de pixels bien étiquetés par rapport au corrigé			86%

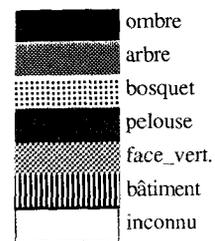
Planche 7. — Résultats de l'étiquetage final de l'image 2.



Fig. 19. — Image 3.



Fig. 20. — Etiquetage initial de l'image 3 par réseau de neurones.



classes	nombre de régions dans le corrigé	étiquetage initial du réseau		étiquetage initial système expert	
		nombre de régions	pourcentage	nombre de régions	pourcentage
bâtiment	16	13	81%	11	69%
ombre	7	1	14%	1	14%
face verticale	0	0	-	0	-
arbre	10	0	0%	0	0%
bosquet	62	28	45%	39	63%
pelouse	67	20	30%	32	48%
TOTAL	162	62	38%	83	51%

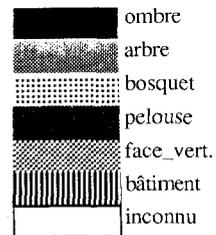
Planche 8. — Résultats de l'étiquetage initial de l'image 3.



Fig. 21. — Etiquetage final de l'image 3 par le système mixte.



Fig. 22. — Etiquetage initial et propagation de contraintes par système-expert de l'image 3.



classes	nombre de régions dans le corrigé	étiquetage système mixte		étiquetage système-expert	
		nombre de régions	pourcentage	nombre de régions	pourcentage
bâtiment	16	15	94%	11	69%
ombre	7	1	15%	0	0%
face verticale	0	0	-	0	-
arbre	10	0	0%	0	0%
bosquet	62	29	47%	41	66%
pelouse	67	37	55%	36	54%
TOTAL	162	82	51%	88	54%
pourcentage de pixels bien étiquetés par rapport au corrigé			69%		71%

Planche 9. — Résultats de l'étiquetage final de l'image 3.