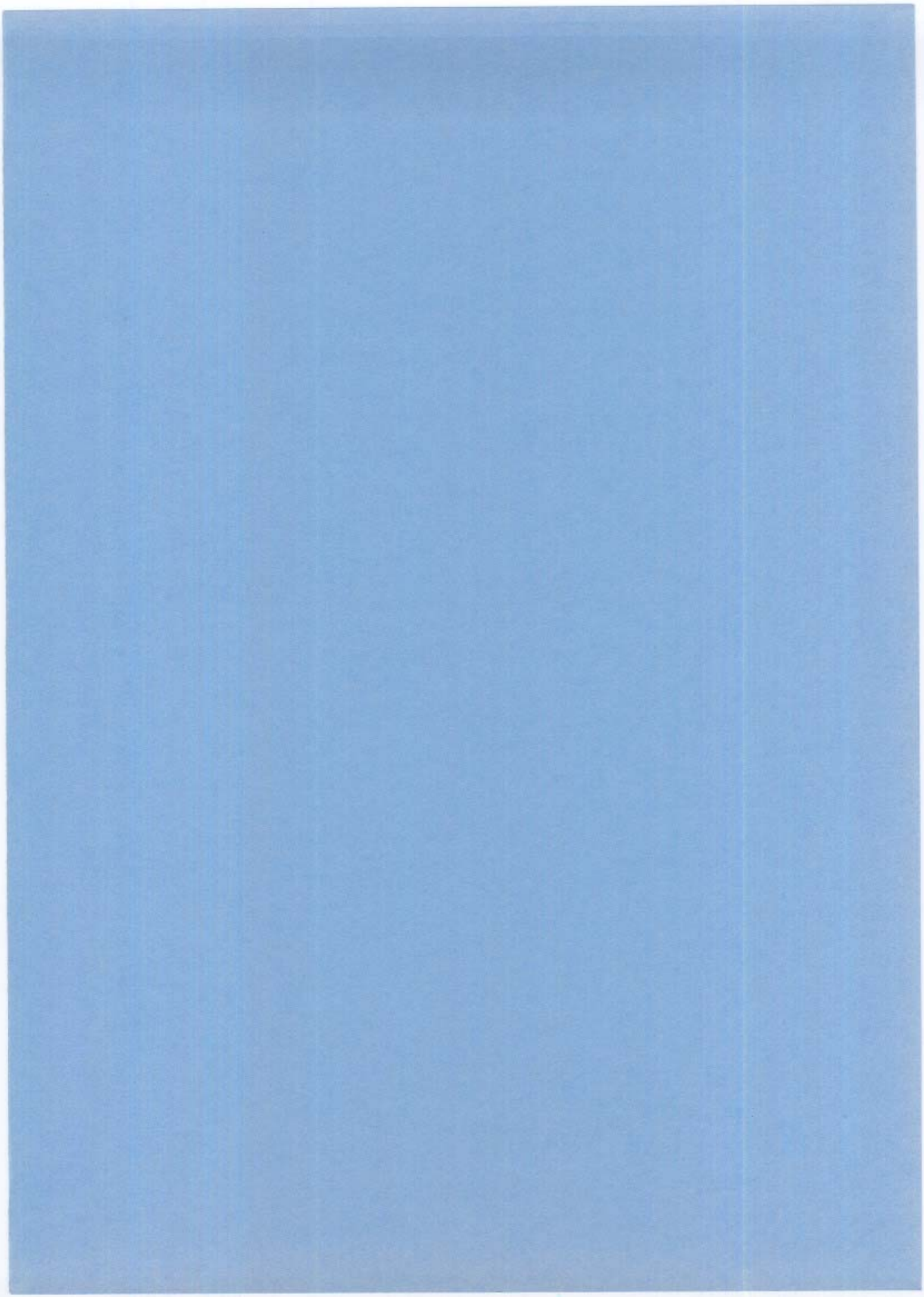
II

APPARIEMENTS DE GRAPHES ET RECONNAISSANCE DE CARACTÈRES MANUSCRITS



1 INTRODUCTION

L'une des questions fondamentales des neurosciences actuelles, qui est encore loin d'être résolue par l'expérimentation, est celle du code neuronal. S'il est désormais prouvé que le système nerveux périphérique et les aires primaires du système nerveux central font une abondante utilisation des *taux d'activité* (fréquence moyenne de décharge) pour coder des stimuli sensoriels ou moteurs, la nature de la représentation mentale dans les aires dites "associatives" est en revanche beaucoup moins claire. La plupart des chercheurs font la simple hypothèse que le codage en taux d'activité caractérise l'ensemble du cortex, et de ce fait ne pensent pas que la configuration temporelle des spikes présente un intérêt particulier dans la représentation de l'information. Remarquons que ce point de vue est largement encouragé par le type même des expériences électrophysiologiques, qui sont presque uniquement constituées d'enregistrements de cellules individuelles.

D'autres auteurs ont au contraire proposé que l'information essentielle se trouve dans les *relations temporelles précises entre spikes* (von der Malsburg, 1981; Abeles, 1982; von der Malsburg et Bienenstock, 1986; voir aussi Sejnowski, 1981) : ils pensent donc que la synchronisation des spikes, et plus généralement les relations temporelles précises entre spikes sur différents axones, sont des éléments fondamentaux du fonctionnement cortical. Dans ce cadre, l'activité individuelle des cellules n'a plus l'importance qu'on lui attribue d'ordinaire, particulièrement au niveau des aires associatives, l'information étant désormais codée principalement dans les *corrélations temporelles* entre activités neuronales (cette discussion sera reprise dans la partie III de cette thèse, chapitre 1). La technique d'enregistrement de cellules individuelles apparaît alors comme une projection plutôt étroite de l'information neuronale parvenant à l'observateur. Certes, il n'est pas aisé de mettre en évidence le rôle des configurations temporelles dans une activité multicellulaire, cependant les récentes expériences de Gray *et al.* (1989) sur le cortex visuel du chat (voir aussi Eckhorn *et al.*, 1988). semblent soutenir l'idée selon laquelle les corrélations d'activité joueraient un rôle important dans la représentation de l'information relationnelle, et en particulier dans le "liage" ("binding") des stimuli visuels élémentaires (ces observations ayant été faites au niveau du cortex visuel primaire).

Une autre question fondamentale est celle des synapses. Il est désormais admis que les synapses chimiques sont à la base de la mémoire à long terme et qu'elles constituent les sites principaux de modulation au cours de l'apprentissage : cette plasticité synaptique est par nature un processus long et irréversible. Cependant, les synapses pourraient être aussi le lieu d'un type de plasticité beaucoup plus rapide : la *plasticité synaptique rapide* (von der Malsburg, 1981), qui demeure pour l'instant spéculative, serait constituée par de changements transitoires et réversibles pouvant

affecter l'efficacité synaptique sur des périodes de temps allant de la milliseconde à la seconde (voir à ce propos l'hypothèse des "twitching synapses", Crick, 1982). Dans cette optique, les synapses pourraient jouer un rôle plus actif que celui qui leur est traditionnellement assigné : les poids synaptiques variant rapidement pourraient fournir le moyen de stocker et de manipuler l'information relationnelle à court terme. Remarquons encore une fois que l'existence de la plasticité synaptique rapide ainsi que son rôle possible dans le fonctionnement du cerveau ne sont pas aisément accessibles aux investigations expérimentales directes.

Il semble que l'utilisation des relations temporelles précises entre activités neuronales ainsi que des synapses à plasticité rapide peut s'avérer d'un grand intérêt dans la plupart des tâches cognitives accomplies par le cortex. En effet, l'information manipulée par notre cerveau est essentiellement une information de type *relationnel* : par exemple, lorsque nous analysons une scène visuelle ou un signal de parole, nous sommes particulièrement sensibles aux différents types de relations spatiales, temporelles ou spatio-temporelles entre primitives visuelles ou auditives (objets, parties d'objets, caractéristiques phonétiques, etc.). Or la synchronie des activités neuronales couplée à des synapses dynamiques forment un système de variables relationnelles, et fournissent de ce fait un support naturel de représentation et de traitement de l'information relationnelle. Elles constituent également une solution directe au problème de la composition ("binding problem").

Cette idée a été avancée pour la première fois par Christoph von der Malsburg dans *The Correlation Theory of Brain Function* (1981), et a été développée ultérieurement avec Elie Bienenstock (von der Malsburg et Bienenstock, 1986; 1987; Bienenstock et von der Malsburg, 1987). Au centre de cette théorie se trouve l'utilisation de *"liens dynamiques"* : formellement, on est conduit à décrire les états de représentation mentale comme des *graphes* (variables, éventuellement étiquetés), au lieu de listes de caractéristiques codées sur des assemblées de cellules. Pour reprendre l'analogie de la mécanique statistique entre les réseaux de neurones et les systèmes thermodynamiques, on peut dire que les variables pertinentes sont ici les valeurs d'interaction $J_{ss'}$ plutôt que les spins x_s. Par conséquent, la dynamique du système dépendra d'une fonction-coût sur les connexions, H(J), au lieu de H(x).

Comme nous l'avons déjà signalé plus haut (chapitre I.4), cette nouvelle approche conduira à la définition d'une nouvelle métrique dans l'espace des exemples, différente de l'habituelle métrique de Hamming qui est à la base de l'ensemble des modèles connexionnistes classiques : la distance que nous utiliserons sera le résultat de la minimisation d'une fonction-coût associée à une opération d'*appariement de graphes*. Cette fonction-coût est destinée à mesurer le degré de déformation nécesssaire pour faire coïncider un pattern avec un autre (pour le traitement des caractères manuscrits, nous la choisirons de type quadratique, et nous l'appellerons "énergie élastique de déformation"). Nous montrerons alors que

l'utilisation de cette autre notion de distance, fondée sur la comparaison des *relations* entre éléments d'une image et éléments d'une autre image, constitue un moyen très naturel d'obtenir une généralisation par rapport à des déformations modérées.

Plus précisément, voici la manière dont l'opération d'appariement de graphes peut être réalisée neurobiologiquement, à travers les corrélations temporelles et les poids synaptiques rapides. La figure 0 représente un détail local de l'application d'un graphe sur un autre, dans deux situations différentes (l'une "favorable" et l'autre "défavorable") : il s'agit de deux nœuds, s et t, d'un graphe G, liés à deux autres nœuds, s´ et t´, d'un graphe G'. On considère alors que les nœuds de G représentent des neurones d'une aire du cortex visuel, et que les nœuds de G' sont des neurones situés dans une autre aire du cortex visuel : un *"lien dynamique"* entre s et s´ est réalisé par une corrélation temporelle forte entre l'activité du neurone s et celle du neurone s´, et, simultanément, par l'augmentation de l'efficacité synaptique rapide entre ces neurones (idem pour t et t´). Pour expliquer le mécanisme d'appariement de graphes, c'est à dire la façon dont les nœuds voisins de G peuvent être connectés à des nœuds voisins de G', il faut alors faire l'hypothèse que *les activités de neurones voisins sont corrélées*, tandis que les activités de neurones éloignés tendent à être décorrélées (cette supposition trouve des éléments de justification par exemple dans les expériences de Mastronarde, 1983). Cette structure de corrélation est donc inhérente au graphe G ainsi qu'au graphe G'.

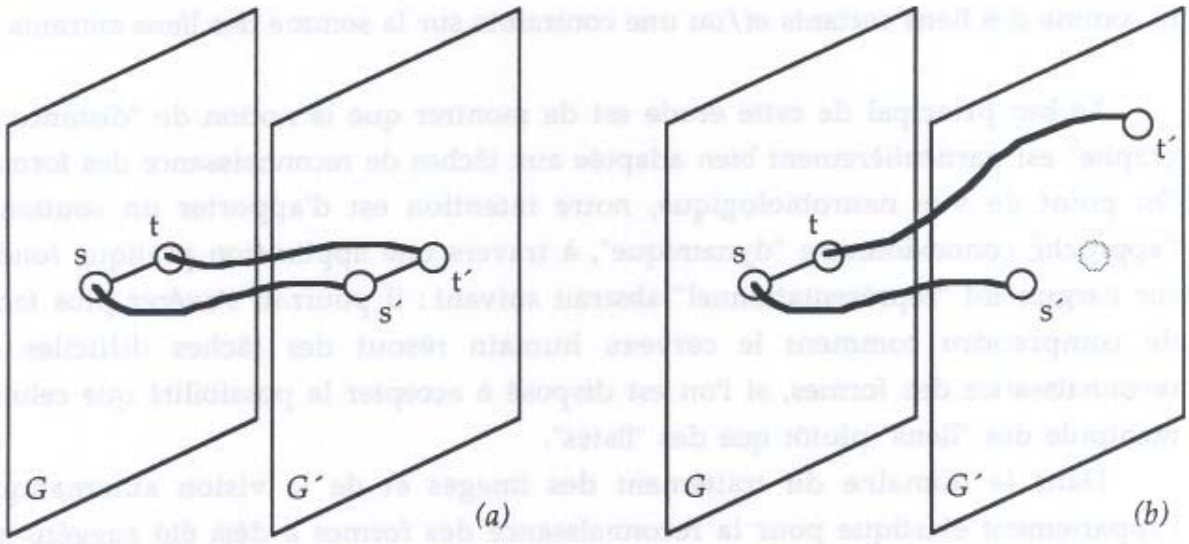


Figure 0 : Appariement de deux nœuds voisins d'un graphe G sur *(a)* deux nœuds voisins d'un autre graphe G', ou bien *(b)* deux nœuds non-voisins de G'. Dans la situation (a), les deux liens (s,s´) et (t,t´) coopèrent, i.e. se renforcent mutuellement à travers les corrélations des nœuds voisins et un mécanisme de plasticité hebbienne rapide (voir les explications du texte).

Dans ce cadre, considérons la situation où s est voisin de t, et s´ voisin de t´ ("cas favorable" de la figure 0a) : s'il existe un lien dynamique de s à s´, c'est à dire si l'activité de s et celle de s´ sont corrélées, alors, par transitivité, t et t´ seront également corrélés, et donc la connexion de t à t´ sera renforcée. Ainsi, l'effet de la

plasticité hebbienne est que *des liens qui connectent des nœuds voisins dans G à des nœuds voisins de G´ coopèrent*, c'est à dire se renforcent mutuellement. Inversement, des liens qui connectent des nœuds (s,t) voisins à des nœuds (s´,t´) non-voisins ("cas défavorable" de la figure 0b), ne coopéreront pas.

Dans ce mécanisme de coopération, on aura reconnu le principe de base des modèles de développement rétinotopique (projection des fibres nerveuses de la rétine au tectum ou de la rétine au cortex visuel) qui utilisent un "marquage temporel" des cellules à travers les corrélations de leurs activités (Willshaw et von der Malsburg, 1976; voir à ce propos la partie III de cette thèse, §2.2.e). Il y a cependant une importante différence entre l'échelle de temps du développement rétinotopique et celle de l'opération d'appariement de graphes postulée ici : tandis que le premier s'inscrit dans le vaste processus de l'ontogenèse, nous supposons que la deuxième est effectuée en une seconde ou une fraction de seconde, c'est à dire le temps nécessaire à le reconnaissance d'une forme. Le principe de base est donc le même (application préservant la topologie, i.e. les relations de voisinage), mais l'appariement de graphes repose sur une plasticité hebbienne beaucoup plus rapide, agissant sur l'échelle de temps psychologique.

Par ailleurs, comme dans les modèles de développement rétinotopique, il est nécessaire d'envisager une forme de *compétition* entre liens dynamiques, pour contrebalancer leurs interactions coopératives (afin que le système ne tombe pas dans un état trivial où toutes les efficacités synaptiques sont à leur valeur maximale). Ceci doit donc se traduire au niveau des graphes par une contrainte sur la somme des liens sortants et/ou une contrainte sur la somme des liens entrants.

Le but principal de cette étude est de montrer que la notion de "distance de graphe" est particulièrement bien adaptée aux tâches de reconnaissance des formes. Du point de vue neurobiologique, notre intention est d'apporter un soutien à l'approche connexionniste "dynamique", à travers une application pratique fondée sur l'argument "représentationnel" abstrait suivant : il pourrait s'avérer plus facile de comprendre comment le cerveau humain résout des tâches difficiles de reconnaissance des formes, si l'on est disposé à accepter la possibilité que celui-ci manipule des "liens" plutôt que des "listes".

Dans le domaine du traitement des images et de la vision automatique, l'appariement élastique pour la reconnaissance des formes a déjà été suggéré par plusieurs auteurs (Burr, 1980; 1981; Tappert, 1982). Cependant, on considère généralement que cette méthode est difficilement utilisable en pratique dans les applications réelles. Les résultats qui seront présentés ici apportent au contraire un soutien à cette approche du point de vue computationnel : par l'utilisation d'une fonction d'énergie qui se prête facilement à une minimisation rapide, nous montrerons que des résultats très satisfaisants peuvent être obtenus avec une quantité raisonnable de calculs dans une tâche de reconnaissance mettant en jeu des déformations de patterns.

2 ALGORITHME D'APPARIEMENT ÉLASTIQUE

Comme nous nous proposons d'utiliser le format de représentation *relationnel* pour traiter un problème de reconnaissance de caractères manuscrits, nous exposerons dans ce chapitre les adaptations du principe général d'"appariement de graphes" qui ont permis de produire une méthode de classification pratique. Par souci de performance, nous avons choisi une formalisation schématique, relativement éloignée du contexte neurobiologique et nous avons été conduits à simplifier plusieurs aspects concernant les appariements : nous avons abouti de la sorte à une fonction-coût formulée comme une *énergie élastique*, dont le principe est d'associer à la configuration "défavorable" de la figure 0b une pénalité égale au carré de la distance entre le nœud t et l'emplacement voisin de s. Cette formulation met clairement en évidence le fait que la ressemblance entre deux images se trouve essentiellement dans le *degré de déformation du plan* qui permet de les superposer (voir Bienenstock et Doursat, 1989, 1991; une formulation similaire a été utilisée par Buhmann *et al.*, 1989, dans un problème de reconnaissance des visages). Ainsi, la minimisation de l'énergie d'appariement de deux caractères fournira directement une *distance* entre ces caractères, puisqu'il s'agit d'une valeur qui mesure leur degré de dissimilitude, et nous utiliserons cette distance au chapitre 3 comme base du processus de classification, en nous appuyant également sur un critère de décision simple, tel que celui des plus proches voisins.

Nous commencerons par développer au §2.1 la manière dont les images qui représentent les caractères manuscrits seront transcrites en *graphes*. Le §2.2 sera consacré à l'élaboration d'une *énergie* H associée à l'appariement de ces graphes, qui tiendra compte à la fois des différences vectorielles sur les arêtes, et des différences picturales sur les nœuds : cette énergie sera composée de la somme de deux termes : une énergie de structure E (quadratique) et une énergie d'images Γ. Il est alors possible de donner une analogie mécanique décrivant cet appariement comme un unique *système déformable*. Nous verrons alors au §2.3 les conditions d'équilibre de ce système, en analysant les deux types de forces locales qui s'exercent sur les nœuds, *force élastique* et *force d'image*, ainsi que les pénalités locales qui leur correspondent et qui représentent le travail de ces forces lors du déplacement d'un nœud. La minimisation de l'énergie sera effectuée par itérations séquentielles sur les nœuds. Le §2.4 s'occupera d'adapter le processus de déformation élastique aux conditions pratiques des simulations. Il faudra alors tenir compte de l'aspect doublement discret des données, c'est à dire du fait que les images utilisées sont composées de pixels binaires, et qu'elles sont définies sur un réseau de points de coordonnées entières. Mais tout d'abord, le résumé préliminaire du §2.0 offre une présentation succincte du modèle.

2.0 Résumé synthétique préliminaire

On dispose d'une base de M caractères manuscrits : $D = \{\Phi^1, \Phi^2, \ldots, \Phi^M\}$ (M=1200). Les images numérisées qui représentent ces caractères ont déjà subi deux prétraitements : une *renormalisation* et un *seuillage* (cf. figure 21, chapitre 3). Donc, pour tout $\alpha = 1...M$, Φ^α est un tableau nxn de pixels noirs ou blancs :

$$\forall\, i=1...n,\ \forall\, j=1...n,\ \Phi^\alpha(i,j) = 1 \text{ ou } 0$$

qui sera complétée par des pixels blancs aux coordonnées inférieures à 1 et supérieures à n.

Pour la représentation relationnelle de ces caractères on utilisera des graphes réguliers, qui ont la forme de *grilles planes à maille carrée* (cf. figure 3) : ainsi, chaque nœud est relié à ses 4 plus proches voisins (sauf 3 sur les bords et 2 dans les coins). La procédure d'appariement élastique consistera alors à déformer un graphe étiqueté représentant l'image Φ^α sur une autre image Φ^β, et à calculer une *énergie* associée à cette déformation (cf. figure 12).

Un graphe sera en général noté G : il est composé d'un ensemble de *nœuds*, O, et d'un ensemble *d'arêtes* A, partie de $O \times O$. Les nœuds seront notés : s, t, u, ... une arête entre s et t sera notée <s,t> (les arêtes sont orientées). Les nœuds portent des *étiquettes* (X_S) noir-et-blanc, qui seront déterminées au départ sur le modèle de l'image Φ^α : pour cela, le maillage $G = (O, A)$ sera placé dans un état non-déformé sur Φ^α et chaque nœud s_{ij} prendra alors la couleur du pixel (i,j) sur lequel il se trouve :

$$X_{Sij} = \Phi^\alpha(i,j)$$

Dans toute la suite, on notera ces étiquettes ($X_S^{\alpha 0}$). Remarquons que, pour les simulations, on restreindra en réalité le graphe G à un *sous-graphe* $G^\alpha = (O^\alpha, A^\alpha)$ dont les contours seront adaptés à la forme du caractère Φ^α : G^α contiendra tous les nœuds noirs, ainsi qu'une certaine couche de nœuds blancs autour des nœuds noirs (cf. figure 22, chapitre 3).

Le graphe étiqueté ainsi obtenu (G^α, $\mathbf{X}^{\alpha 0}$) est placé sur l'image Φ^β où il doit subir une déformation pour faire coïncider l'étiquetage qu'il porte avec les pixels de Φ^β. Chaque nœud s est repéré par des coordonnées (k_S, l_S), et le respect des étiquettes impose alors :

$$\Phi^\beta(k_S, l_S) = X_S^{\alpha 0}$$

(nœuds blancs sur pixels blancs et nœuds noirs sur pixels noirs). Dans toute la suite on désignera par V_S les coordonnées courantes du nœud s et par V_S^0 les coordonnées initiales qui ont permis de définir son étiquette, c'est à dire : $V_S = (k_S, l_S)$ et $V_S^0 = (i,j)$ pour $s = s_{ij}$. Donc $X_S^{\alpha 0}$ représente $\Phi^\alpha(V_S^0)$.

Le graphe se trouve alors dans un certain *état de déformation* V, auquel est associée une *énergie élastique* E(V) dont l'expression sera :

$$E(V) = \frac{1}{2} \sum_{<s,t> \in A^{\alpha}} \left| (V_t - V_s) - (V_t^0 - V_s^0) \right|^2$$

Remarquons d'autre part que la contrainte de respect des étiquettes peut aussi s'exprimer de façon équivalente comme l'annulation d'une deuxième fonction-coût, ou *pénalité d'étiquettes* $\Gamma^{\alpha\beta}$, définie par :

$$\Gamma^{\alpha\beta}(V) = \sum_{s \in O^{\alpha}} \delta\left(\Phi^{\alpha}(V_s^0), \Phi^{\beta}(V_s)\right), \quad \text{avec} \quad \begin{array}{l} \delta(X, X') = 0 \text{ si } X = X' \\ \delta(X, X') = 1 \text{ si } X \neq X' \end{array}$$

c'est à dire que l'on impose : $\Gamma^{\alpha\beta}(V) = 0$. Par ailleurs, on définira une troisième fonction-coût pour cette déformation, notée S(V), dont le rôle sera de pénaliser les *superpositions* de nœuds sur les même pixels (S est une somme sur les pixels) :

$$S(V) = \sum_{(k,l)} (\rho(k,l) - 1)^{+2}, \text{ avec } \rho(k,l) = \text{card}\{s \in O^{\alpha}; V_s = (k,l)\}$$

où $x^+ = x$ si $x \geq 0$, et $x^+ = 0$ si $x < 0$ ($\rho(k,l)$ représente le degré "d'encombrement" sur le pixel (k,l)). Finalement, *l'énergie totale* H(V) associée à la déformation V et devant être minimisée, s'écrira en général :

$$H^{\alpha\beta}(V) = E(V) + \lambda \Gamma^{\alpha\beta}(V) + \kappa S(V)$$

(où λ et κ sont des coefficients strictement positifs), et, en particulier, dans la version stricte concernant le respect des étiquettes, on aura :

$$H(V) = E(V) + \kappa S(V), \text{ avec } \Gamma^{\alpha\beta}(V) = 0$$

(ce qui revient à choisir $\lambda = +\infty$). Par conséquent, étant donné les images Φ^{α} et Φ^{β}, on cherche la déformation V la moins coûteuse, c'est à dire la valeur d'énergie minimale suivante :

$$H^{\alpha\beta}{}_{min} = \min_{V;\, \Gamma^{\alpha\beta}(V)=0} H(V)$$

En pratique, il n'est pas possible d'atteindre le minimum absolu $H^{\alpha\beta}{}_{min}$, à moins d'envisager une recherche exhaustive très longue (les graphes contiennent environ une centaine de nœuds, et l'image étendue environ un millier de pixels). C'est pourquoi on se contentera d'un *minimum local* : la relaxation du graphe vers un état d'équilibre sera obtenue en visitant *séquentiellement* les nœuds, et en

améliorant leurs positions individuellement, les autres nœuds étant maintenus immobiles. L'amélioration se fera directement dans le sens d'une diminution de H, c'est à dire "à température 0".

Ainsi, étant donné un nœud s, on cherchera à améliorer sa position V_s, les autres positions $(V_t)_{t\neq s}$ étant fixées. Si on ne tenait compte ni des étiquettes ni de la pénalité S, la meilleure position pour un nœud s du point de vue de l'énergie élastique E prise seule, serait donnée par la formule suivante :

$$V_s^b = \frac{1}{n_s} \sum_{\substack{t \in O^\alpha; \\ <s,t> \in A^\alpha}} \left(V_t + V_s^0 - V_t^0\right)$$

Il s'agit du *barycentre* des positions suggérées par les nœuds voisins de s, c'est à dire les nœuds reliés à s par une arête, qui sont en nombre n_s (dans la plupart des cas, $n_s = 4$) (remarquons que ce point doit être rapporté à des coordonnées entières pour correspondre à un pixel du réseau).

En réalité, puisqu'on doit tenir compte de Γ et S, la position optimale ne se trouve pas en V_s^b, et il faut ajouter une petite déviation v_s, de telle sorte que le pixel de coordonnées $V_s = V_s^b + v_s$, qui sera choisi par le nœud s, ait même étiquette que ce nœud et offre également un encombrement $\rho(V_s)$ faible (à minimser conjointement avec la pénalité élastique). Finalement la position optimale pour s sera notée : $V_{s\,min} = V_s^b + v_{s\,min}$ (où $v_{s\,min}$ réalise le minimum d'un supplément local, dérivé de H, dont la formule est : $\Delta H(v_s) = n_s \mid v_s \mid^2 + \kappa\,(2\rho(V_s^b+v_s) - 1)^+$, sous la contrainte $\Gamma = 0$).

L'optimisation séquentielle de l'énergie s'écrit alors :

$$V_s(\tau) = V_s^b(\tau) + v_{s\,min}(\tau), \quad \text{avec} \quad V_s^b(\tau) = \frac{1}{n_s} \sum_{\substack{t \in O^\alpha; \\ <s,t> \in A^\alpha}} \left(V_t(\tau-1) + V_s^0 - V_t^0\right)$$

où τ est l'indice des temps, qui sera incrémenté de 1 à chaque visite de nœud. Une *itération* est définie comme une visite complète de tous les nœuds du graphe G^α, dans un ordre aléatoire fixé à l'avance. La minimisation comprendra N itérations : si N est suffisamment grand (en pratique N≥10), le graphe finira par s'immobiliser complètement dans un minimum local (où à la rigueur tombera dans un cycle court caractérisé par une permutation incessante de quelques nœuds). Cependant, on peut vouloir écourter le temps de calcul, et ne pas chercher à atteindre le miminum local : dans ce cas, on prendra N<10, voire N=0. Il reste alors à définir l'*initialisation* des positions des nœuds du graphe, ou *itération-0*, qui fournit V(0) : cette étape est assez délicate, et sera commentée en détail au §2.4.d.

2.1 Représentation des caractères par des graphes

2.1.a Rappels sur les graphes étiquetés

Un graphe G est composé d'un ensemble de *nœuds* O, et *d'arêtes* A, ce qu'on écrira : $G = (O, A)$. L'armature A est un sous-ensemble particulier de couples de nœuds, c'est à dire $A \in O \times O$, et ses éléments seront notés <s,t>. Par ailleurs, on attribue des *étiquettes* à tous les composants de G, nœuds et arêtes, c'est à dire des caractéristiques numériques ou symboliques : les étiquettes portées par les nœuds, appelées *descripteurs*, forment un vecteur $\mathbf{X} = (X_s)_{s \in O}$, et les étiquettes portées par les arêtes, appelées *relateurs*, forment un tableau bidimensionnel $\mathbf{C} = (C_{st})_{<s,t>}$ (les composantes C_{st} pour $(s,t) \notin A$ ne sont pas définies) (figure 1).

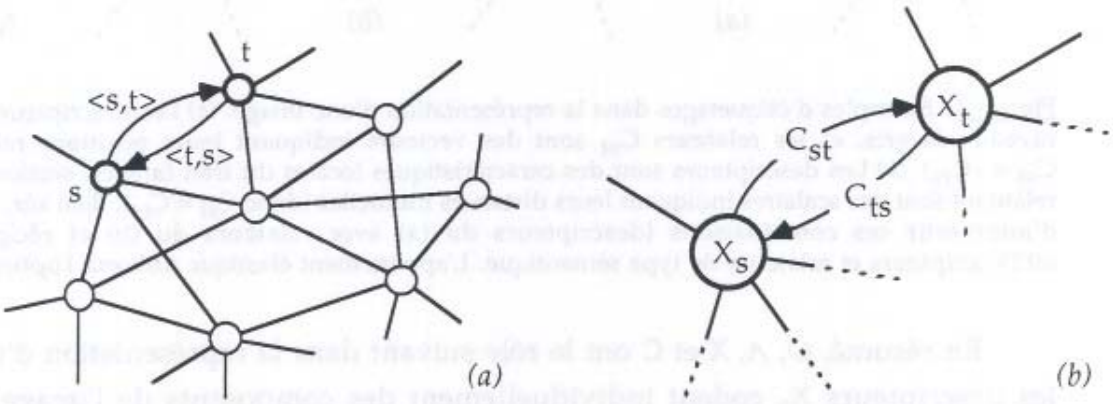


Figure 1 : Composants des graphes étiquetés. *(a)* Nœuds et arêtes : les arêtes sont en général orientées, donc <s,t> ≠ <t,s>. *(b)* Etiquettes de nœuds X_s (descripteurs) et étiquettes d'arêtes C_{st} (relateurs).

De façon très générale, ces étiquettes peuvent prendre des valeurs réelles ou vectorielles, ou encore représenter des objets non quantifiables, c'est à dire en général des symboles, ou des énoncés de relations pour les C_{st}. Par exemple, dans la représentation d'une image, X_s figure un élément local qui peut être un simple pixel ou une caractéristique plus élaborée (contour, forme géométrique élémentaire, etc.), et C_{st} code les relations géométriques entre les morceaux d'image X_s et X_t (distance, position relative, rapport de taille, etc.) (figure 2).

Mais quel que soit l'étiquetage particulier $(\mathbf{X}, \mathbf{C})$ choisi pour le graphe (O, A), les arêtes A et leurs étiquettes $\mathbf{C}$ ont toujours pour but d'exprimer les *relations* entre les différents éléments de $(O, \mathbf{X})$. Les graphes étiquetés qui codent des images ou des scènes sont en général des projections directes de l'objet réel, c'est à dire qu'il est possible d'assigner des positions spatiales aux nœuds qui portent chacun un morceau d'image, et, par un système de relateurs géométriques cohérents, d'obtenir une reproduction plus ou moins fidèle de la scène en deux ou trois dimensions. Bien sûr, cette reproduction n'est pas rigoureuse, c'est à dire exacte dans ses proportions ou homogène dans ses détails : par exemple, les descripteurs peuvent être le résultat d'une transformation locale par des convolutions (lissage gaussien,

Fourier, ondelettes, etc.). Mais l'image est rarement soumise à une transformation globale qui ferait interagir des composantes éloignées. Donc les représentations en graphe découpent la scène en composants élémentaires, et leur propriété essentielle est de préserver les relations de voisinage entre ces composants.

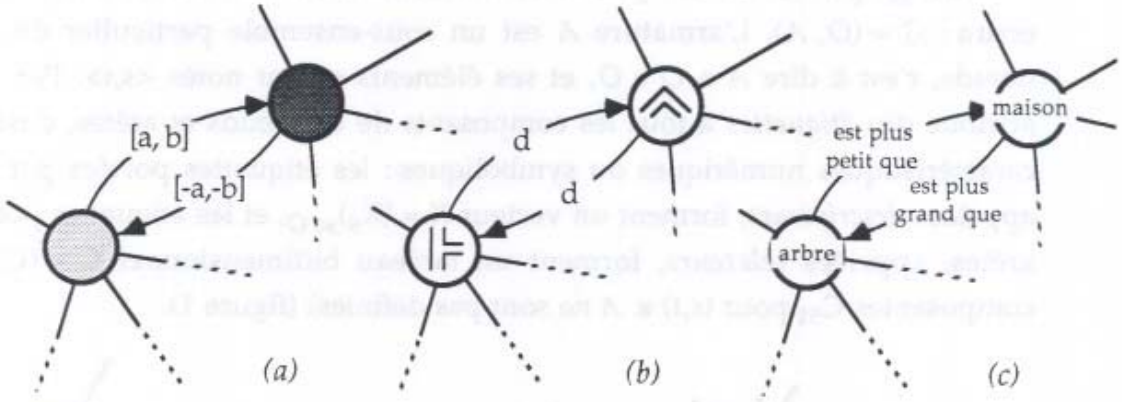


Figure 2 : Exemples d'étiquetages dans la représentation d'une image. *(a)* Les descripteurs X_s sont des niveaux-de-gris, et les relateurs C_{st} sont des vecteurs indiquant leurs positions relatives (donc $C_{st} = -C_{ts}$). *(b)* Les descripteurs sont des caractéristiques locales du trait (angles, sections, etc.) et les relateurs sont des scalaires indiquant leurs distances mutuelles (donc $C_{st} = C_{ts}$). Bien sûr, il est possible d'intervertir ces combinaisons (descripteurs du (a) avec relateurs du (b) et réciproquement). *(c)* Descripteurs et relateurs de type sémantique. L'appariement élastique utilisera l'option (a).

En résumé, O, A, $\mathbf{X}$ et $\mathbf{C}$ ont le rôle suivant dans la représentation d'une image : les descripteurs X_s codent individuellement des composants de l'image, le graphe $G = (O, A)$ sert de support formel à ces composants en attribuant chaque X_s à un nœud s et en le reliant à d'autres X_t par les arêtes <s,t> (G figure un *réseau d'interactions* entre composants qui servira de base à la définition d'une énergie d'appariement), et les relateurs C_{st} donnent une signification géométrique à ce support en précisant sur chaque arête <s,t> la relation réelle qu'entretiennent X_s et X_t. Dans la suite, on dira que la réunion du graphe et de ses relateurs, c'est à dire le couple $(G, \mathbf{C})$, forme le *support géométrique* de l'image $\mathbf{X}$. Nous décrivons à présent les images et les supports qui ont été choisis pour représenter les caractères manuscrits.

2.1.b Représentation des caractères par des grilles régulières

Les caractères seront schématisés par des *images binaires* portées par des *grilles planes régulières à maille carrée* (figure 3). Les nœuds du graphe se trouvent aux intersections de m lignes et p colonnes disposées à intervalles réguliers sur le plan : ils forment donc un tableau de points de coordonnées (i,j), avec $1 \le i \le m$ et $1 \le j \le p$. Les étiquettes des nœuds reproduisent directement la couleur du pixel qui se trouve au même endroit, et qui est déterminée par rapport à un seuil de luminance : elle est blanche si le pixel est plus clair que le seuil, et noire si ce pixel est plus foncé; donc $X_s \in \{blanc, noir\}$. D'autre part, les arêtes ne relient que les nœuds *plus proches*

voisins : elles forment par conséquent un maillage composé de segments unitaires orthogonaux du plan : les segments horizontaux reliant (i,j) à (i,j+1), et les segments verticaux reliant (i,j) à (i+1,j).

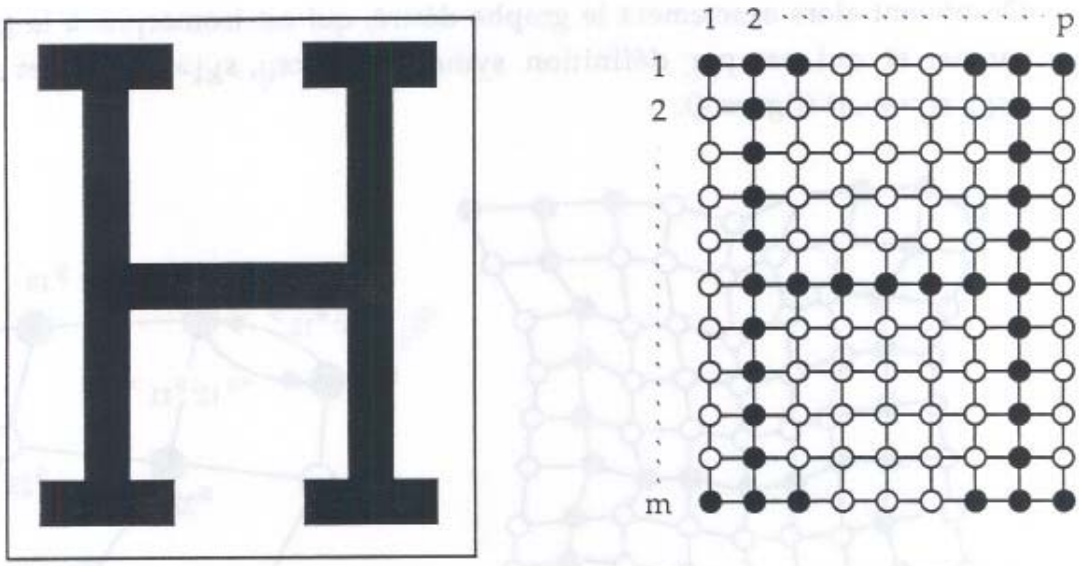


Figure 3 : Représentation relationnelle d'un caractère. Le graphe est une grille plane à maille carrée, composée de mp nœuds, reliés par m(p–1) arêtes horizontales et p(m–1) arêtes verticales. Donc la connectivité est de 4 en chaque nœud (sauf sur les bords où elle vaut 3, et dans les coins où elle vaut 2). Les étiquettes des nœuds sont noires ou blanches.

Le support de grille à maille carrée sera donc le même pour tous les caractères, et leurs différences ne résideront que dans les images **X** étiquetant les nœuds. Cependant, cet état n'est que provisoire, car, comme on le verra dans la suite, la grille sera amenée à *se déformer* au cours du processus de comparaison des caractères. A partir de cette rapide description, nous allons formaliser avec plus de précision le support vectoriel (*G*, **C**) qui définit la grille à maille carrée, en séparant la structure formelle *G* de sa valeur géométrique **C**. On verra alors que ce support orthonormal correspond seulement à l'état *standard* des caractères, c'est à dire leur état *non-déformé* : en d'autres termes, on verra comment les déformations ultérieures de ces images sont justement équivalentes à un changement de **C** sur le même *G*.

En l'absence d'étiquetage, la structure du graphe *G* est un *maillage* qui doit pouvoir être défini sans faire appel à la géométrie (*A* est purement un ensemble de couples de nœuds). Dans ce but, l'ensemble des mxp nœuds sera numéroté par deux indices, l'un variant de 1 à m, l'autre de 1 à p :

$$O = \{s_{11}, s_{12}, \dots, s_{1p}, s_{21}, s_{22}, \dots, s_{2p}, \dots, s_{m1}, s_{m2}, \dots, s_{mp}\}$$

(par la suite, cette double indexation ne sera pas systématiquement utilisée, et les nœuds seront notés plus simplement s ou t : ici, elle sert seulement à clarifier la

définition de A). Puis l'ensemble des arêtes A est rempli de la façon suivante :

$$A = \{<s_{ij}, s_{kl}>,\ 1 \le i,k \le m,\ 1 \le j,l \le p;\ (i = k \text{ et } |j-l| = 1) \text{ ou } (|i-k| = 1 \text{ et } j = l)\}.$$

On obtient alors exactement le graphe désiré, qui est isomorphe à la grille à maille carrée, et qui est par définition symétrique ($<s_{ij}, s_{kl}> \in A$ si et seulement si $<s_{kl}, s_{ij}> \in A$) (figure 4).

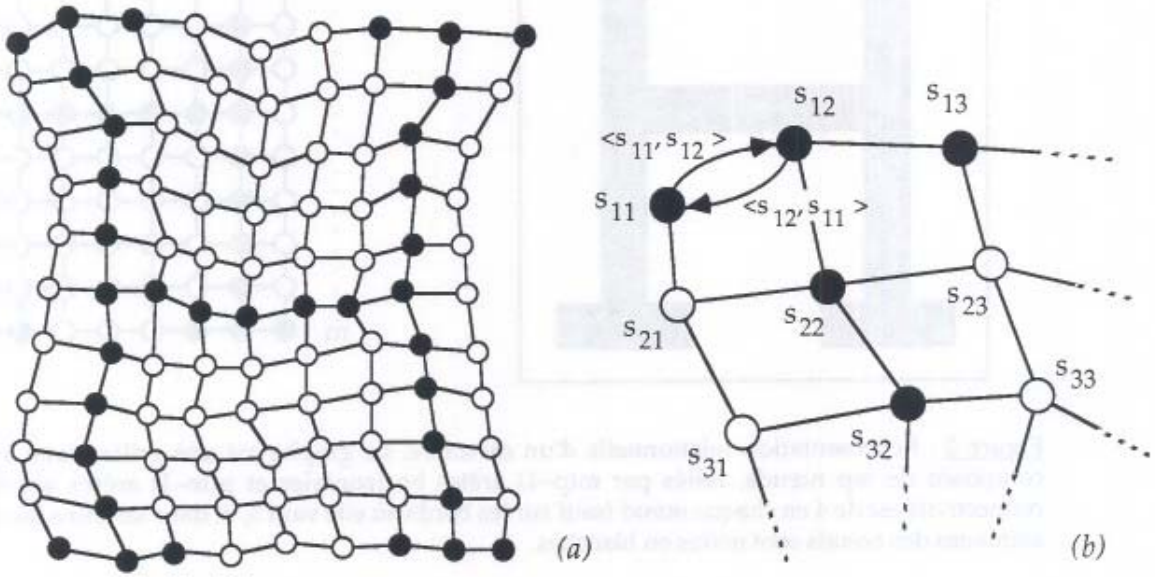


Figure 4 : Maillage G. Les nœuds portent des étiquettes noires ou blanches, mais les arêtes n'ont pas encore de signification géométrique (G est une "trame souple"). *(a)* Vue d'ensemble. *(b)* Détail supérieur gauche : étant donné la double indexation des nœuds, les arêtes ne relient que les paires d'indices distantes d'une unité (l'orientation des arêtes est précisée ici entre les nœuds s_{11} et s_{12}).

En adoptant cette grille parmi les multiples variantes de support qui se présentaient, on s'est donc restreint à la classe des graphes réguliers constitué d'un *motif* géométrique répété périodiquement (il n'y avait en effet aucune raison de prendre un support a priori non-homogène). La maille carrée n'est bien sûr pas la seule possibilité : il serait possible de prendre une connectivité plus forte (par exemple, chaque nœud est relié à ses 8 plus proches voisins, donc : $<s_{ij}, s_{kl}> \in A \Leftrightarrow |i-k| \le 1$ et $|j-1| \le 1$), ou encore une maille triangulaire, hexagonale, etc. (figure 5). Cependant, la connectivité aux 4 plus proches voisins reste l'option la plus simple en deux dimensions.

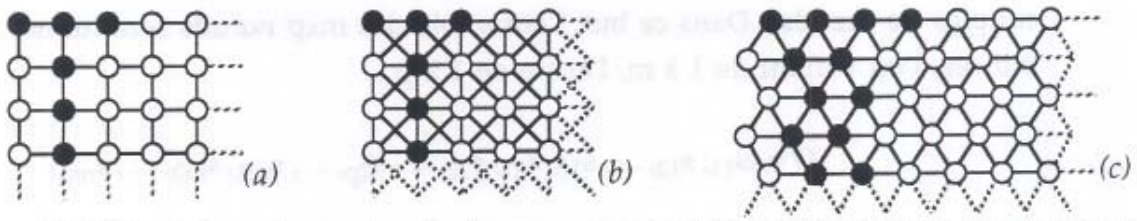


Figure 5 : Différents types de graphes réguliers pouvant servir à la représentation des caractères. Pour plus de clarté, ces maillages sont montrés dans un état non-déformé *(a)* Connectivité à 4 : maille carrée. *(b)* Connectivité à 8. *(c)* Connectivité à 6 : maille triangulaire.

Les étiquettes d'arêtes C_{st} seront quant à elles des *vecteurs* du plan de l'image : chacune est composée de deux coordonnées, $C_{st} = (a_{st}, b_{st})$ (figure 2a), et la structure complète $(G, \mathbf{C})$ sera appelée *support vectoriel* de l'image. Donc, pour représenter la structure rigide avec arêtes unitaires et angles droits définie plus haut (figure 3), les (C_{st}) devront être des vecteurs unitaires othogonaux : (1, 0), (0, 1) et (–1, 0), (0, –1). En reprenant la double indexation des nœuds qui a servi à construire G, on peut alors donner une description exacte et complète de l'étiquetage relationnel de la façon suivante :

$$\forall \langle s_{ij}, s_{kl} \rangle \in A,\ C_{sij\ skl} = (k-i, l-j).$$

Munie de cet étiquetage $\mathbf{C}$, la grille à maille carrée G sera appelée *grille orthonormée* (figure 6). Remarquons que l'on pourrait se contenter de relations *scalaires* (donc symétriques) qui signaleraient les distances entre nœuds (figure 2b) : dans le cas de la grille, on appliquerait alors l'unique relation *« est à distance 1 de »* sur toutes les arêtes de G, c'est à dire : $\forall \langle s,t \rangle \in A,\ C_{st} = 1$. Cependant, cette représentation relationnelle scalaire serait alors invariante par rotation, pour tous les angles de –180° à +180°, ce qui n'est pas souhaitable avec les caractères (penser à "9" et "6" ou "d" et "p"). C'est pourquoi on préfère la précision apportée par les deux coordonnées d'un vecteur à la simple norme de ce vecteur : dans ce cas, l'invariance par rotation sera "faible", c'est à dire approximative pour des petits angles.

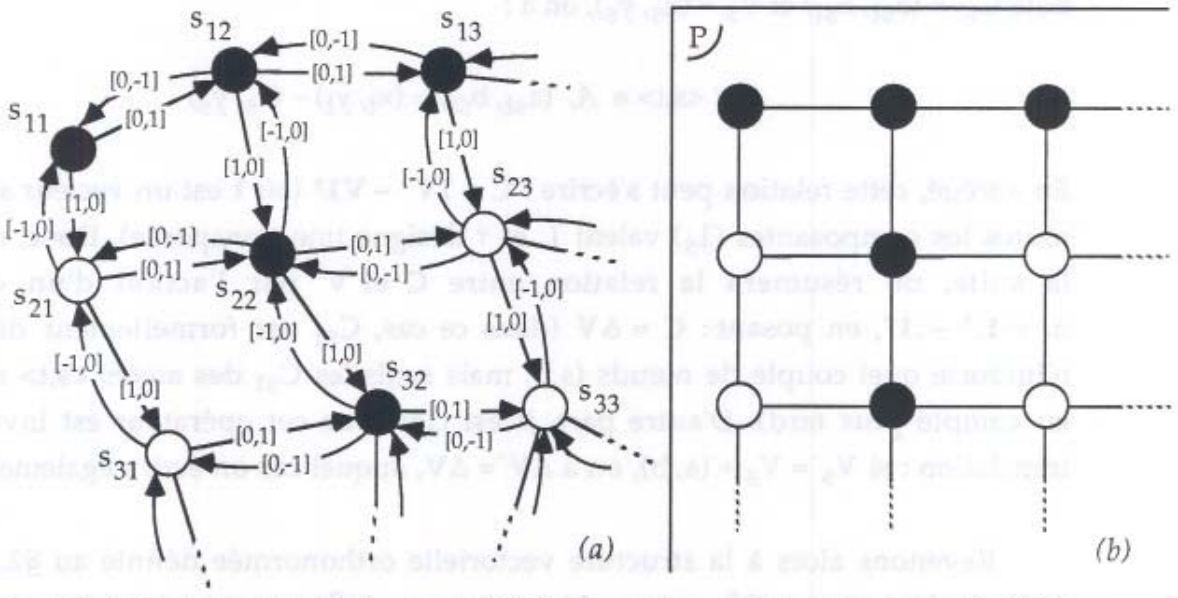


Figure 6 : Etiquetage vectoriel du graphe G (détail supérieur gauche de la figure 4). *(a)* Chaque arête $\langle s_{ij}, s_{kl} \rangle$ porte un vecteur unitaire issu de la différence entre les indices de nœuds : (k–i, l–j). *(b)* Muni de cet étiquetage $\mathbf{C}$, le graphe G forme alors une grille orthonormée sur le plan (le couple $(G, \mathbf{C})$ est une "trame rigide" qui sert de support à l'image).

2.1.c Coordonnées de nœuds

Comme nous venons de le montrer, la définition de G est *indépendante* de la définition de $\mathbf{C}$, c'est à dire que la structure du graphe est toujours séparable de sa

valeur géométrique. Donc ce n'est pas parce que la structure formelle (O, A) est isomorphe à une grille plane à maille carrée, que les vecteurs orthonormés définis ci-dessus sont les seules étiquettes possibles pour les arêtes : en réalité, ils ne constituent qu'un cas particulier parmi une infinité de structures vectorielles **C** ajustables de façon cohérente à G, et l'ensemble de tous ces étiquetages représente précisément l'ensemble de toutes les *déformations du graphe* dans le plan vectoriel.

De façon générale, les relateurs vectoriels (C_{st}) peuvent s'écrire comme des différences entre descripteurs vectoriels (V_s) :

$$\forall \langle s,t \rangle \in A,\ C_{st} = V_t - V_s$$

Les (V_s) sont donc des vecteurs rattachés aux nœuds, qui représentent les *coordonnées* de ces nœuds sur le plan : elles peuvent être réelles et varier continûment (figure 7), le cas défini au §2.1.b n'étant qu'un cas particulier dans lequel les nœuds sont placés sur des positions entières (figure 8). Ainsi, la relation C_{st} qu'entretiennent deux nœuds s et t est déduite des propriétés individuelles de ces nœuds (ici, une position sur le plan), par l'application d'une opération simple (ici, la soustraction). On rappelle que **C** est un tableau bidimensionnel (composantes doublement indexées), que **V** a le format d'un vecteur (composantes simplement indexées), et que leurs composantes sont des couples de coordonnées. Donc, si on note $C_{st} = (a_{st}, b_{st})$ et $V_s = (x_s, y_s)$, on a :

$$\forall \langle s,t \rangle \in A,\ (a_{st}, b_{st}) = (x_t, y_t) - (x_s, y_s)$$

En abrégé, cette relation peut s'écrire : $\mathbf{C} = \mathbf{1}\mathbf{V}^\dagger - \mathbf{V}\mathbf{1}^\dagger$ (où **1** est un vecteur sur O dont toutes les composantes (1_s) valent 1, et † désigne une transposée). Donc, dans toute la suite, on résumera la relation entre **C** et **V** par l'action d'un opérateur $\Delta. = \mathbf{1}.^\dagger - .\mathbf{1}^\dagger$, en posant : $\mathbf{C} = \Delta\mathbf{V}$ (dans ce cas, C_{st} est formellement défini pour n'importe quel couple de nœuds (s,t), mais seuls les C_{st} des arêtes <s,t> seront pris en compte plus tard). D'autre part, il est clair que cet opérateur est invariant par translation : si $V_s' = V_s + (a, b)$, on a $\Delta\mathbf{V}' = \Delta\mathbf{V}$, auquel cas on écrira également $\mathbf{V} \equiv \mathbf{V}'$.

Revenons alors à la structure vectorielle orthonormée définie au §2.1.b : on la notera désormais $\mathbf{C}^0$, et on désignera par $\mathbf{V}^0$ les coordonnées des nœuds correspondantes. Donc, à une translation uniforme près, ces coordonnées valent :

$$\forall\, i = 1...m,\ \forall\, j = 1...p,\ V^0_{Sij} = (i,j)$$

Le support vectoriel $(G, \mathbf{C}^0) = (G, \Delta\mathbf{V}^0)$, représente la grille orthonormée et sera appelé *support standard* des caractères. On constate donc que la position standard d'un nœud est exactement dictée par les deux indices qu'il a reçus lors de la définition de A : $V^0_{Sij} = (i,j)$ (figure 8).

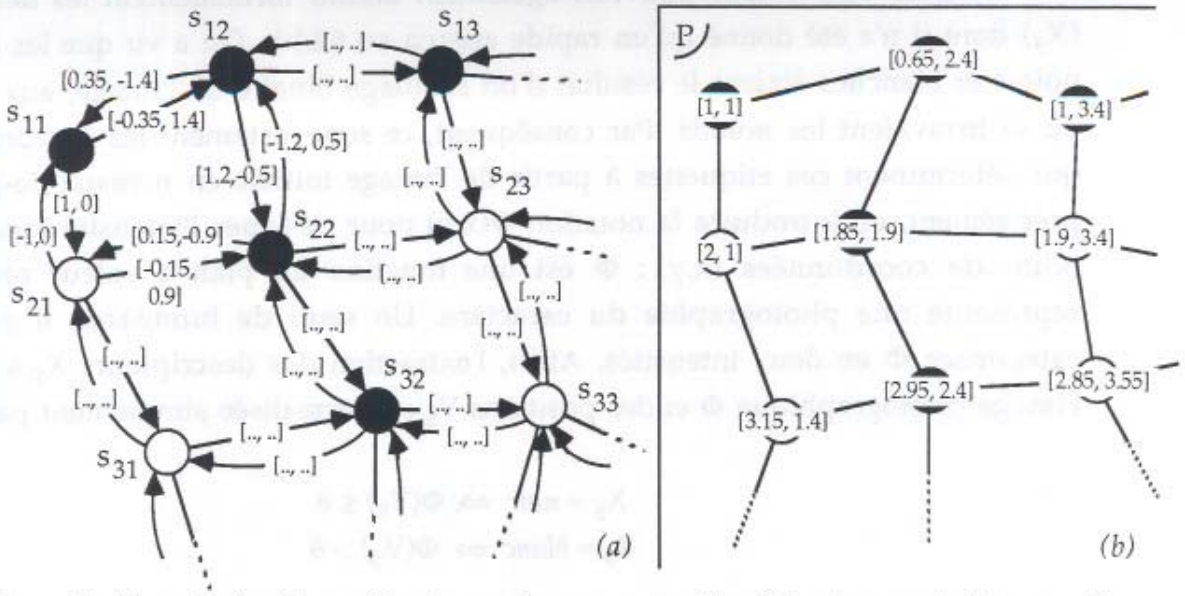


Figure 7 : Exemple de déformation du graphe au moyen d'un étiquetage vectoriel non-orthonormé (comparer avec la figure 6). *(a)* Le graphe est toujours la grille plane à maille carrée *G*, mais les vecteurs d'arêtes C_{st} ne sont en général pas unitaires, ni orthogonaux. *(b)* Le support vectoriel (*G*, **C**) ainsi obtenu est une *déformation* de la grille orthonormée. Il est paramétré de façon équivalente par des vecteurs V_s attachés au nœuds, qui représentent les coordonnées de ces nœuds sur le plan, et dont les différences mutuelles fournissent les étiquettes d'arêtes C_{st} (par exemple, pour $s=s_{11}$ et $t=s_{12}$, on a ici : $V_s=(1,1)$ et $V_t=(0.65, 2.4)$, donc : $C_{st}=V_t-V_s=(-0.35, 1.4)=-C_{ts}$).

En revanche, la variation de sa position V_{sij} en dehors de l'état standard est a priori indépendante de i et j et peut donc prendre des coordonnées réelles quelconques entre les positions entières initiales : $V_{sij} = (x, y)$ (en indexant par le nœud s_{ij} : $V_{sij} = (x_{sij}, y_{sij})$). Par conséquent, il est clair que si l'on impose aux nœuds un système de coordonnées différent du système standard, on contraint les arêtes à s'étirer ou à se contracter par rapport à la situation normale, et les angles entre arêtes à devenir plus obtus ou plus aigus que l'angle droit (tous ces mouvements sont codés dans **C**). Bref, toute transformation de $\mathbf{V}^0$ en **V** autre qu'une translation uniforme, provoquera une *déformation de la grille orthonormée.*

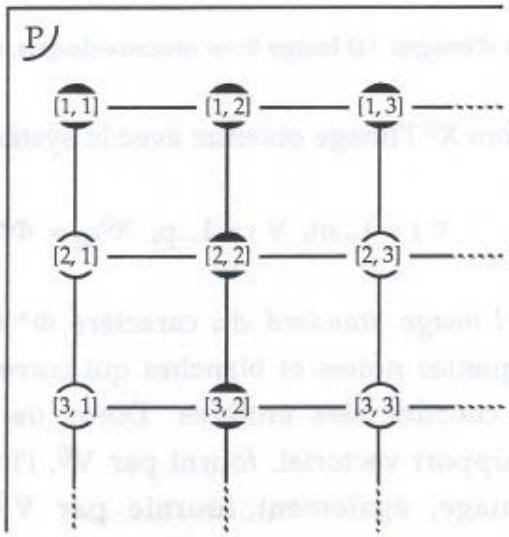


Figure 8 : Coordonnées des nœuds associées à la grille orthonormée de la figure 6 (à comparer avec la figure 7b). Dans cet état particulier, appelé *état standard*, les coordonnées reproduisent exactement les indices des nœuds (par exemple, pour $s=s_{12}$, on a : $V_s=(1,2)$).

Dans ce cadre, nous pouvons également définir formellement les descripteurs (X_S) dont il n'a été donné qu'un rapide aperçu au §2.1.b. On a vu que les étiquettes noires et blanches étaient le résultat d'un seuillage binaire de l'image, aux positions où se trouvaient les nœuds. Par conséquent, ce sont justement les coordonnées V_S qui déterminent ces étiquettes à partir de l'image initiale en niveaux-de-gris. Plus précisément, on introduira la notation $\Phi(x,y)$ pour désigner l'intensité lunineuse au point de coordonnées (x,y) : Φ est une fonction du plan à valeur réelles, qui représente une photographie du caractère. Un seuil de luminance θ permet de catégoriser Φ en deux intensités. Ainsi, l'extraction des descripteurs X_S à partir de l'image photographique Φ et des positions V_S est formalisée simplement par :

$$X_S = noir \Leftrightarrow \Phi(V_S) \leq \theta$$
$$X_S = blanc \Leftrightarrow \Phi(V_S) > \theta$$

En abrégé, on notera cette opération : $X_S = \Phi^*(V_S)$, c'est à dire : $\mathbf{X} = \Phi^*(\mathbf{V})$. La fonction Φ^* est donc la version seuillée de Φ (figure 9).

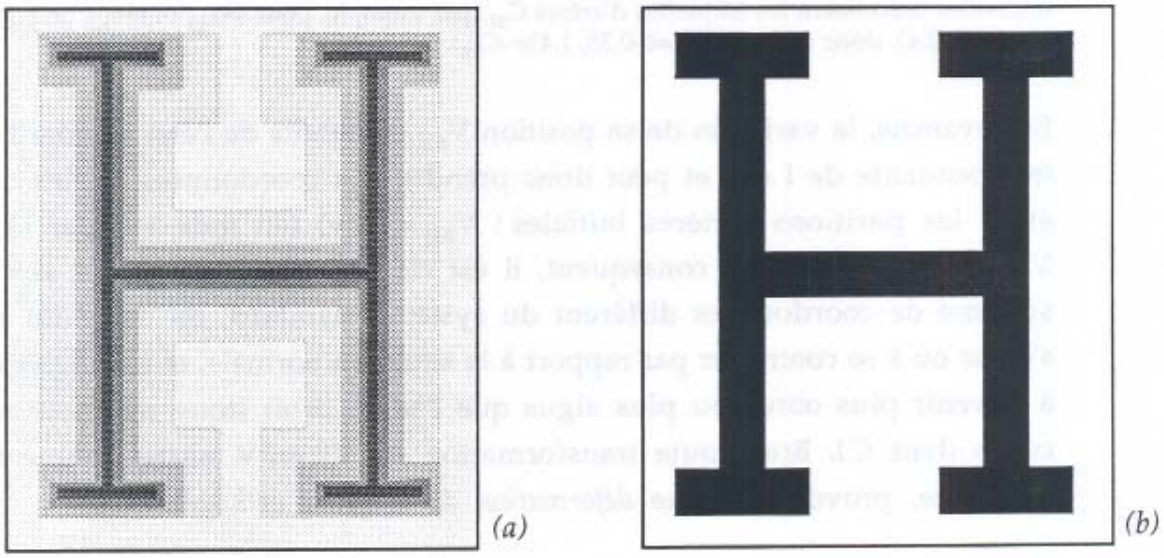


Figure 9 : Exemples d'images. *(a)* Image Φ en niveaux-de-gris. *(b)* Version seuillée Φ^* en noir-et-blanc.

On note alors $\mathbf{X}^0$ l'image obtenue avec le système standard $\mathbf{V}^0$, soit $\mathbf{X}^0 = \Phi^*(\mathbf{V}^0)$:

$$\forall\, i = 1...m,\ \forall\, j = 1...p,\ X^0{}_{Sij} = \Phi^*(V^0{}_{Sij}) = \Phi^*(i,j)$$

$\mathbf{X}^0$ sera appelé *l'image standard* du caractère Φ^* (image de la figure 3b) : elle est composée d'étiquettes noires et blanches qui correspondent exactement aux nœuds placés sur les coordonnées entières. Donc, de même que $\mathbf{C}^0$ n'est qu'un cas particulier de support vectoriel, fourni par $\mathbf{V}^0$, l'image standard $\mathbf{X}^0$ n'est qu'un cas particulier d'image, également fournie par $\mathbf{V}^0$. Ainsi, de façon générale, les étiquetages $\mathbf{C}$ et $\mathbf{X}$ peuvent prendre une infinité de valeurs différentes pour

représenter le même caractère Φ^*, la cohérence de l'ensemble étant assurée par leur variation solidaire à travers le paramètre **V** (figure 10).

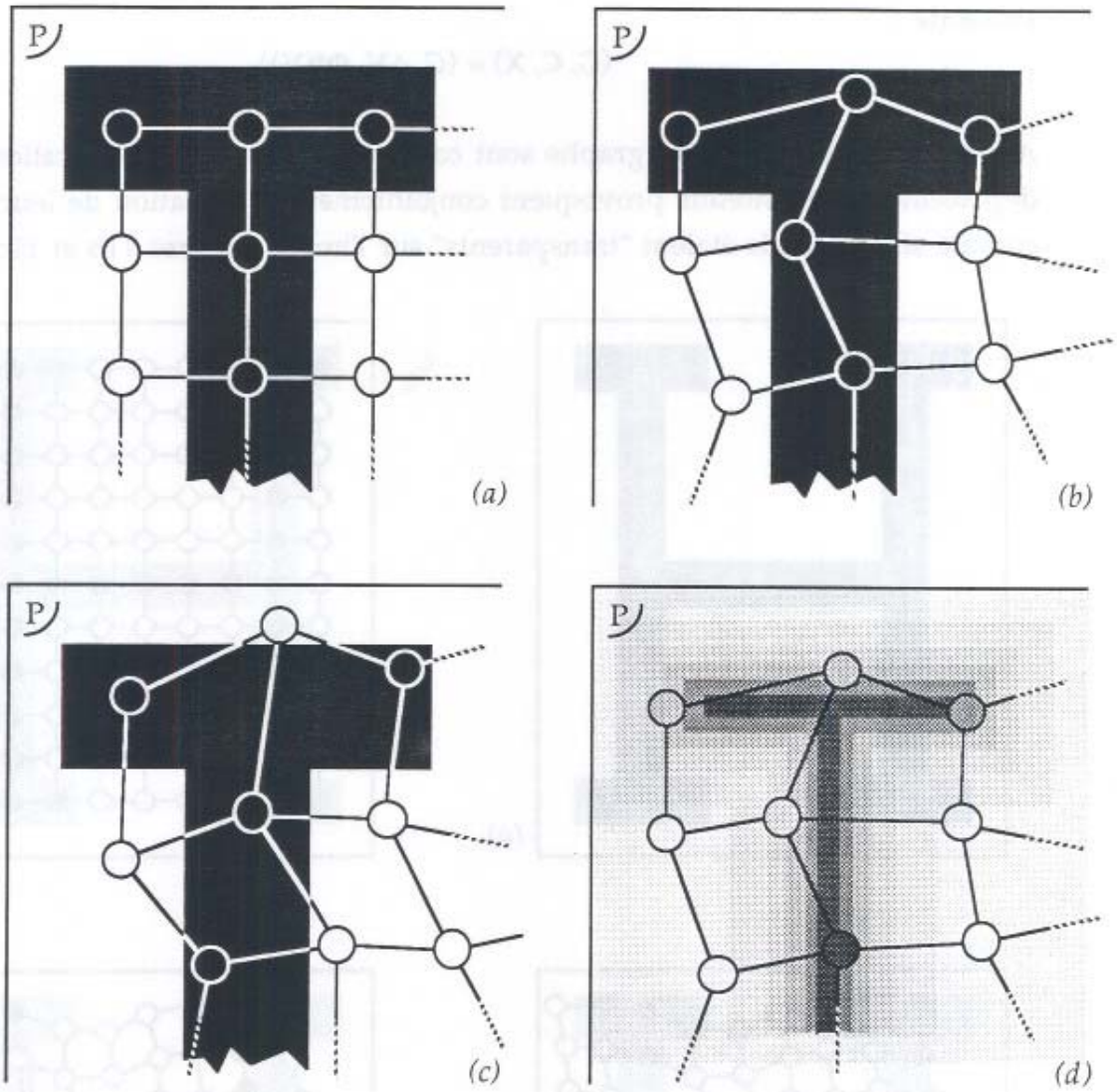


Figure 10 : Différents étiquetages des nœuds en fonction de leurs coordonnées. Les trois premiers cas utilisent l'image seuillée Φ^*, et le quatrième montre un exemple avec l'image réelle Φ (détail supérieur gauche des figures 9b et 9a). *(a)* Etat standard (coordonnées entières de la figure 8). *(b)* Etat déformé (coordonnées de la figure 7b) : les nœuds ne sont pas sortis de leurs zones noires et blanches respectives, et ils ont gardé leur couleur standard. *(c)* Autre état déformé : ici, l'étiquette de s_{12} est devenue blanche et celle de s_{31} est devenue noire. *(d)* Etat déformé du (b) sur l'image réelle : les étiquettes des nœuds adoptent les niveaux-de-gris des zones sur lesquelles elles se trouvent.

2.1.d Représentations conformes et représentations difformes

En résumé, la grille à maille carrée *G* reçoit une structure géométrique au moyen d'un étiquetage vectoriel des arêtes (C_{st}), lequel peut se déduire d'un étiquetage vectoriel des nœuds (V_s) par différences mutuelles : $C_{st} = V_t - V_s$. Les (V_s) représentent les coordonnées des nœuds sur le plan, et leurs variations provoquent des déformations du graphe. Dans le cas particulier où $V_s = V_s^0$ (à une translation près), le graphe *G* est une grillle orthonormée. D'autre part, les

coordonnées (V_S) fournissent les étiquettes des nœuds (X_S) à travers la fonction-image du plan Φ^* : $X_S = \Phi^*(V_S)$. Donc, au total, dans ces graphes étiquetés, les descripteurs **X** et les relateurs **C** sont liés à travers le paramètre **V** de la façon suivante :

$$(G, \mathbf{C}, \mathbf{X}) = (G, \Delta\mathbf{V}, \Phi^*(\mathbf{V})).$$

Ainsi, les déformations du graphe sont compensées par des modifications de **X** : les déplacements des nœuds provoquent conjointement la variation de leurs étiquettes, comme si les nœuds étaient "transparents" sur l'image (figures 11b et 11c).

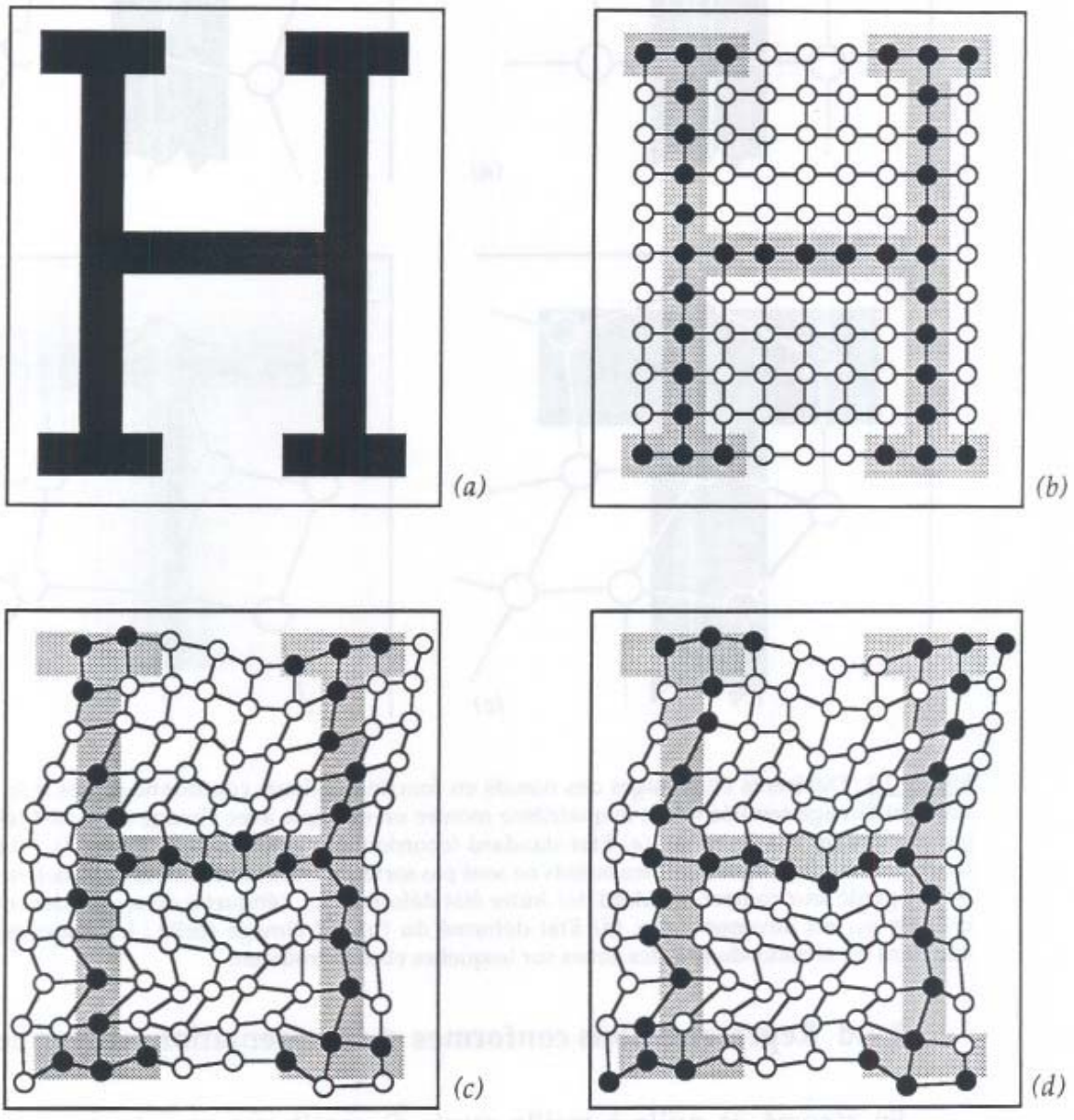


Figure 11 : Différentes représentations relationnelles d'une même image Φ^*. *(a)* Image seuillée de la figure 9b. *(b)* Représentation standard par la grille orthonormée (pour plus de lisibilité, l'image est reproduite en gris au lieu de noir). *(c)* Représentation conforme : la grille est déformée et, parallèlement, les étiquettes de nœuds ont changé pour respecter les zones d'images (le point de référence étant situé au centre du nœud). *(d)* Représentation difforme : la grille est déformée exactement comme au (c), mais les nœuds ont gardé leurs étiquettes standard du (a).

Dans une telle représentation la disposition relative des éléments de Φ^* est toujours conservée, et on dira qu'il s'agit d'une *représentation conforme* à Φ^* (tous les étiquetages de la figure 10 sont des étiquetages conformes). La grille orthonormée du §2.1.b est un cas particulier de représentation conforme, obtenu pour $\mathbf{V} = \mathbf{V}^0$, et on l'appellera *représentation standard* de Φ^* : $(G, \mathbf{C}^0, \mathbf{X}^0) = (G, \Delta\mathbf{V}^0, \Phi^*(\mathbf{V}^0))$.

Par conséquent, pour produire une véritable déformation de l'image Φ^* à l'aide du graphe, il faut changer les coordonnées des nœuds sans changer leurs étiquettes. Par exemple, à partir de la représentation standard, on obtient un état non-conforme en remplaçant $\mathbf{V}^0$ par $\mathbf{V}$, mais en gardant l'image standard $\mathbf{X}^0$, soit : $(G, \Delta\mathbf{V}, \Phi^*(\mathbf{V}^0))$. Donc, de façon générale, on sort des représentations conformes dès que les étiquettes portées par les nœuds ne correspondent plus à leurs positions, comme si les nœuds étaient "opaques" (figure 11d). La disposition relative des éléments de Φ^* n'est plus conservée, et ces représentations seront appelées *difformes*, par opposition aux représentations conformes.

A l'aide des représentations conformes et difformes, nous allons formuler une énergie d'appariement entre caractères au paragraphe 2.2 : cette énergie sera une *mesure* de la déformation qu'il faut faire subir à un caractère Φ^* pour qu'il ressemble au mieux à un autre caractère Φ'^*. Donc, il ne s'agira pas d'un d'appariement de nœuds et d'arêtes $G = (O, A)$, puisque ce sont les mêmes pour tous les caractères (et dans tous leurs états de déformation), mais d'un appariement d'étiquetages, c'est à dire d'une comparaison entre descripteurs $\mathbf{X}$ et $\mathbf{X}'$ et relateurs $\mathbf{C}$ et $\mathbf{C}'$.

2.2 Energie d'appariement

On dispose d'un ensemble de M caractères manuscrits, sous forme d'images en niveaux-de-gris, c'est à dire des fonctions réelles du plan : $D = \{\Phi^1, \Phi^2, \ldots, \Phi^M\}$. Ces images Φ peuvent être transformées en images noir-et-blanc Φ^*, qui sont des fonctions binaires du plan, par un seuillage de part et d'autre d'une valeur θ. La base des caractères prétraités s'écrit alors : $D^* = \{\Phi^{1*}, \Phi^{2*}, \ldots, \Phi^{M*}\}$ (voir par exemple la figure 21, chapitre 3). De plus, pour les besoins des simulations numériques, le plan de l'image devra être discrétisé. Cependant, avant d'aborder ces problèmes d'ordre pratique au paragraphe 2.4, nous resterons pour l'instant (au §2.2 et au §2.3) dans le cadre théorique des fonctions réelles définies sur un plan continu, en utilisant les images de la base D.

Dans ce paragraphe, nous allons définir une *fonction d'énergie* H associée à la comparaison de deux caractères de D, Φ^α et Φ^β. Formellement, il s'agit de l'énergie d'appariement de deux graphes étiquetés, l'un représentant Φ^α et l'autre Φ^β. Ces graphes sont des représentations conformes de Φ^α et Φ^β, la première étant placée dans l'état standard orthonormé pour servir de référence à la deuxième, soit : $(G, \Delta\mathbf{V}^0, \Phi^\alpha(\mathbf{V}^0))$ et $(G, \Delta\mathbf{V}, \Phi^\beta(\mathbf{V}))$. L'énergie H est donc une fonction de $\mathbf{V}$

paramétrée par Φ^α et Φ^β : elle sera conçue pour mesurer le degré de dissimilitude entre ces deux caractères, et s'exprimera comme la somme de deux termes, E et Γ, le premier fondé sur la structure vectorielle (différences entre ΔV et ΔV^0), et le deuxième sur l'image (différences entre $\Phi^\beta(V)$ et $\Phi^\alpha(V^0)$). Donc le but de l'appariement sera de *minimiser* la valeur de H par rapport à V, c'est à dire de trouver la représentation relationnelle de Φ^β qui approche le plus celle de Φ^α, tant du point de vue de la structure géométrique (déformation minimale de V par rapport à V^0) que du point de vue de l'image (recouvrement maximal entre $\Phi^\beta(V)$ et $\Phi^\alpha(V^0)$).

Cependant, dans toute la suite, il sera plus simple de voir cet appariement comme la *déformation d'un unique graphe* portant l'image de Φ^α sur le plan de Φ^β. En effet, on peut considérer de façon équivalente que l'on dispose d'un seul graphe étiqueté, dont la structure est déformable, mais dont les étiquettes de nœuds sont fixes : il s'agit d'une représentation difforme de Φ^α, $(G, \Delta V, \Phi^\alpha(V^0))$, qui se déplace sur l'image Φ^β. Le but est le même qu'avant : il s'agit de trouver les meilleures positions (V_S) pour les nœuds, c'est à dire celles qui restent à la fois proches des positions initiales $(V_S{}^0)$ pour déformer le moins possible les arêtes, et qui en même temps révèlent des étiquettes variables $\Phi^\beta(V_S)$ semblables aux étiquettes fixes $\Phi^\alpha(V_S{}^0)$ que les nœuds portent. La fonction-coût H est donc l'énergie d'un unique système mécanique déformable : elle sera composée à la fois d'un coût de déformation, E, et d'un coût de non-coïncidence des étiquettes, Γ.

2.2.a Energie élastique de déformation

Le problème est donc de trouver une déformation du caractère Φ^α qui le fasse ressembler au caractère Φ^β (figure 12) : à cette déformation correspond une énergie E, qui sera appelée *énergie élastique*. Si E est faible, cela signifie que la déformation nécessaire à la mise en coïncidence n'est pas importante, donc que Φ^α ressemble déjà à Φ^β dans son état normal, tandis que si l'énergie est grande, cela signifie que ces caractères ne se ressemblent pas naturellement. Donc, pour apprécier cette ressemblance, on demandera une identité entre les étiquetages des nœuds issus de ces deux images, afin de mesurer la déformation du graphe qui en résulte, soit :

$$X^{\alpha 0} = X^\beta$$

où l'on note $X^{\alpha 0} = \Phi^\alpha(V^0)$ et $X^\beta = \Phi^\beta(V)$. La satisfaction de cette condition se répercute alors dans la structure vectorielle ΔV du graphe, et c'est à ce niveau qu'apparaît la mesure de la déformation. Plus tard, au §2.2.b, on envisagera le cas où $X^{\alpha 0}$ et X^β ne coïncident pas parfaitement : il faudra alors ajouter un coût correspondant à leurs différences (cf. figure 13).

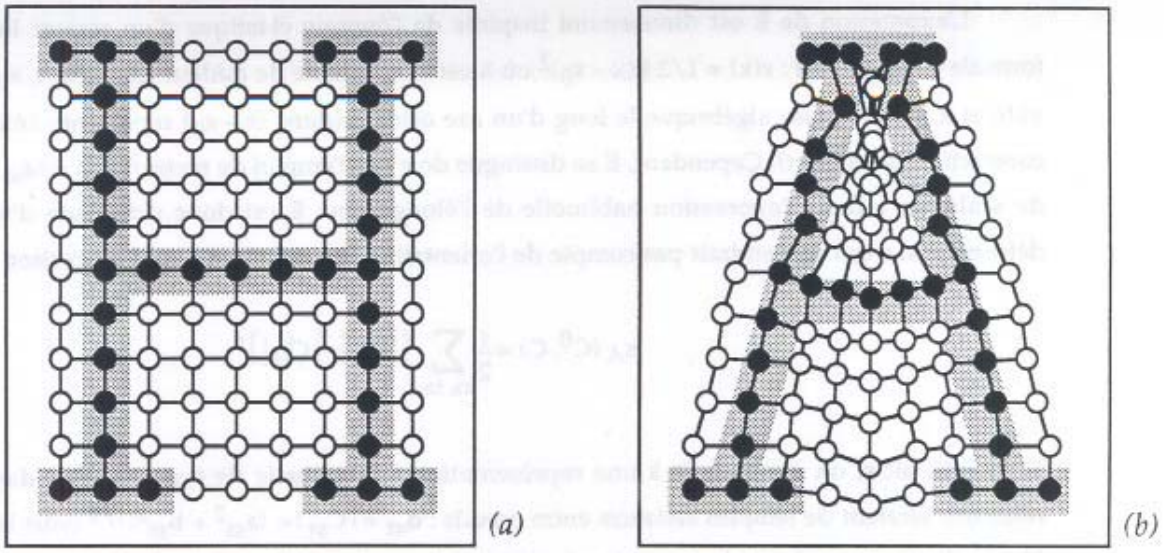


Figure 12 : Déformation intégrale d'un caractère sur un autre (pour plus de clarté, cette illustration utilise des images binaires). Dans cet exemple, Φ^α est l'image d'un "H" et Φ^β l'image d'un "A". *(a)* Représentation en graphe standard de Φ^α avec son image à l'arrière-plan (cf. figure 11b). *(b)* Le même graphe est ensuite déformé sur l'image Φ^β, de telle sorte que les étiquettes de ses nœuds soient en accord avec les zones blanches et noires de Φ^β. Donc, dans l'état où il se trouve, ce graphe est à la fois une représentation *difforme* de Φ^α et une représentation *conforme* à Φ^β. L'énergie E associée à cette déformation est donnée par la formule ci-dessous.

Nous choisirons une énergie E de type quadratique, par analogie avec les systèmes physiques élastiques. A la déformation de la structure $C^0 = \Delta V^0$ en $C = \Delta V$ sera associée la quantité :

$$E_A(C^0, C) = \frac{1}{2}\left|C - C^0\right|^2 = \frac{1}{2}\sum_{<s,t>}\left|C_{st} - C^0_{st}\right|^2$$

(E est une somme sur les arêtes de *A*). Comme on s'intéresse aux variations de E en fonction de **V**, à travers $\mathbf{C} = \Delta\mathbf{V}$, on notera plus commodément : $E_A(C^0, C) = E(\mathbf{V})$. L'expression de E(**V**) est alors :

$$E(\mathbf{V}) = \frac{1}{2}\sum_{<s,t>}\left|V_t - V_s - V^0_t + V^0_s\right|^2$$

et il s'agit de minimiser E par rapport à **V**. En l'absence de contraintes, il est clair que la déformation la moins coûteuse est la déformation nulle $\mathbf{C} - \mathbf{C}^0 = 0$. Mais sous la contrainte de coïncidence des images, la recherche du minimum de E s'écrit :

$$E_{min} = \min_{\mathbf{V};\, \chi^\beta = \chi^{\alpha 0}} E(\mathbf{V})$$

Donc tout le problème est que le processus de relaxation de E est restreint à l'espace des déformations **V** qui satisfont à $\Phi^\beta(\mathbf{V}) = \chi^{\alpha 0}$.

L'expression de E est directement inspirée de l'énergie élastique d'un ressort linéaire, dont la formule classique est : $\varepsilon(x) = 1/2\ k(x - x_0)^2$ où k est la constante de raideur du ressort, x_0 sa longueur à vide et x une abscisse algébrique le long d'un axe orienté (donc $(x - x_0)$ représente l'élongation ou la contraction du ressort). Cependant, E se distingue de ε par l'emploi de *vecteurs* $C_{st} = (a_{st}, b_{st})$ à la place de scalaires x dans l'expression habituelle de l'élongation : E est donc différente d'une énergie de déformation ε qui ne tiendrait pas compte de l'orientation des arêtes, et dont l'expression serait :

$$\varepsilon_A(C^0, C) = \frac{1}{2} \sum_{<s,t>} \left(|C_{st}| - |C_{st}^0| \right)^2$$

Si on utilisait ε, on reviendrait à une représentation relationnelle de type scalaire, dans laquelle les relateurs seraient de simples *distances* entre nœuds : $d_{st} = |C_{st}| = (a_{st}^2 + b_{st}^2)^{1/2}$ (voir la remarque du §2.1.b). Dans ce cas, la seule cause de pénalité locale serait une élongation de l'arête, quelle que soit sa rotation. Avec ε, les arêtes sont directement analogues à des ressorts de longueur à vide $|C_{st}^0| = 1$, tandis qu'avec E, on doit considérer qu'elles sont d'abord composées d'un segment rigide C_{st}^0 (de longueur 1 et d'orientation fixe), puis d'un ressort de longueur à vide 0 attaché à l'extrémité de ce segment (ce point de vue mécanique sera précisé au paragraphe 2.3).

Remarquons d'autre part qu'une expression plus générale de E aurait été :

$$E_A(C^0, C) = \frac{1}{2} \sum_{<s,t>} k_{st} \left| C_{st} - C_{st}^0 \right|^2$$

où k_{st} représente une constante de raideur propre à chaque arête <s,t>. Ce type d'énergie non-uniforme permet de différencier les régions du graphe $(G, C, X^{\alpha 0})$ en leur attribuant des degrés de rigidité plus ou moins élevés : en effet, les caractéristiques locales des images n'ont pas toutes même importance, et il peut s'avérer utile de pénaliser plus fortement la déformation de celles qui constituent les "traits saillants" spécifiques de ces images. Par exemple, hors contexte, le chiffre "5" se distingue en général de la lettre "S" par son angle droit dans le coin supérieur gauche : il est donc important de préserver ce coin, qui fait la particularité du "5", en pénalisant davantage les déformations qui tendraient à l'arrondir, et, pour cela, les arêtes situées dans cette zone devraient recevoir une constante de raideur plus grande. Ceci veut dire que pour Φ^α = "5" et Φ^β = "S", l'énergie élastique associée à un V qui réalise $\Phi^\alpha(V^0) = \Phi^\beta(V)$ serait nettement augmentée. En revanche, l'extrémité inférieure du "5" a une signification beaucoup moins importante (elle peut être inclinée de diverses manières et être plus ou moins longue), et sa déformation locale, tant qu'elle reste raisonnable, n'a pas de grandes conséquences sur la reconnaissance : donc les k_{st} à ce niveau resteraient faibles. Ainsi, dans une représentation non-homogène, chaque arête <s,t> porterait non seulement une information sur la position relative des constituants X_s et X_t, mais également une information sur la *force de leur interaction*, c'est à dire sur l'importance de leur proximité dans la figuration d'une caractéristique locale. Au total, les relateurs seraient donc des triplets de valeurs $(a_{st}, b_{st}, k_{st}) = (C_{st}, k_{st})$, mais, contrairement aux vecteurs C_{st} qui varient pour créer une déformation du caractère, les valeurs k_{st} sont des paramètres fixes au cours de l'appariement : ils sont propres à α, et on peut donc les noter k_{st}^α. Nous avons donc opté pour des constantes de raideur invariables et uniformes sur toutes les arêtes : $\forall\ \Phi^\alpha \in D, \forall$ <s,t>, $k_{st}^\alpha = 1$.

2.2.b Energie de couplage des images

En résumé, on cherche une structure vectorielle $\mathbf{V}$ différente de $\mathbf{V}^0$ pour déformer l'image $\mathbf{X}^{\alpha 0}$ de telle sorte que, d'une part, elle coïncide avec $\mathbf{X}^\beta = \Phi^\beta(\mathbf{V})$, et que d'autre part, cette déformation soit la moins grande possible, c'est à dire minimise $E(\mathbf{V})$. Cependant, au lieu de requérir *a priori* la coïncidence des images, c'est à dire d'optimiser E sous cette contrainte, il sera en général plus commode d'étendre l'espace de la recherche à tous les $\mathbf{V}$ et d'acheminer *progressivement* la déformation de Φ^α vers la coïncidence avec Φ^β. Ainsi, l'image $\mathbf{X}^\beta = \Phi^\beta(\mathbf{V})$, variable au long du parcours de la déformation, ne coïncidera pas tout de suite avec l'image constante $\mathbf{X}^{\alpha 0}$, mais seulement lorsque la représentation atteindra un certain état de déformation (figure 13). La contrainte stricte $\mathbf{X}^\beta = \mathbf{X}^{\alpha 0}$ est donc assouplie en *contrainte faible*, et on définit alors une autre fonction-coût, notée Γ, qui sera chargée de mesurer la distance entre les deux images. L'expression de Γ est :

$$\Gamma_O(\mathbf{X}^{\alpha 0}, \mathbf{X}^\beta) = \sum_{s \in O} f\left(X_s^{\alpha 0}, X_s^\beta\right)$$

(Γ est une somme sur les nœuds de O). Comme pour E, on notera plus simplement : $\Gamma_O(\mathbf{X}^{\alpha 0}, \mathbf{X}^\beta) = \Gamma^{\alpha\beta}(\mathbf{V})$, pour mettre en évidence la dépendance de Γ en fonction de $\mathbf{V}$.

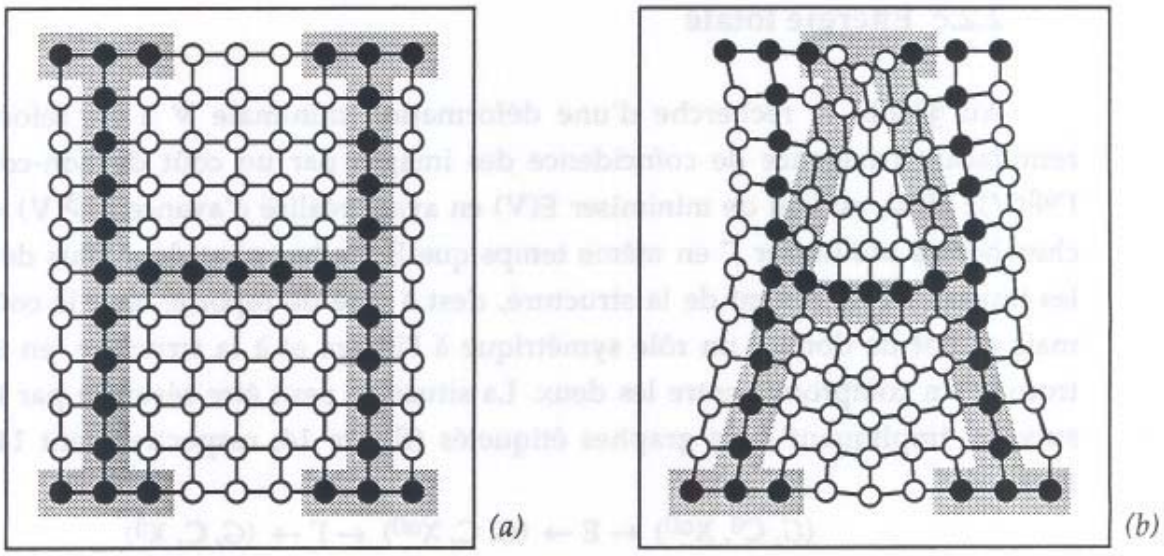


Figure 13 : Déformation inachevée d'un caractère sur un autre. Même principe que dans la figure 12, à la différence ici que le graphe représentant Φ^α n'est pas encore conforme à l'image Φ^β. Dans cette situation, la déformation est moins importante que précédemment, donc E est plus faible; en revanche, les étiquettes des nœuds ne correspondent pas toutes avec les zones de l'image où elles se trouvent, donc il s'ensuit une pénalité Γ non-nulle qui devra être rajoutée à E.

La fonction $f(X, X')$ est un coût local, qui mesure la différence entre deux étiquettes d'un même nœud. Selon le type d'étiquettes adopté, il en existe plusieurs versions, qui sont en général symétriques en X et X'; par exemple :

- $f_1(X, X') = |X - X'|^2$
- $f_2(X, X') = -XX'$
- $f_3(X, X') = 1 - \delta(X, X')$
- .....

Mais quelle que soit sa formulation, la fonction f doit être conçue pour être d'autant plus grande que X et X' diffèrent : donc le cas $X = X'$ doit nécessairement correspondre au minimum absolu de $f(X, X')$ (ce minimum n'étant pas forcément nul : voir f_2). Par conséquent, le minimum de Γ est atteint avec la coïncidence des étiquettes issues de Φ^α et Φ^β : si on définit $\Gamma^{\alpha\beta}{}_{min} = \min_{\mathbf{V}} \Gamma^{\alpha\beta}(\mathbf{V})$, on a : $\Gamma^{\alpha\beta} = \Gamma^{\alpha\beta}{}_{min} \Leftrightarrow \mathbf{X}^{\alpha 0} = \mathbf{X}^\beta$. Cependant, contrairement à $E(\mathbf{V})$, dont la valeur nulle est atteinte seulement par $\mathbf{V}^0$ (à une translation près), il existe une infinité de $\mathbf{V}$, non-équivalents par translation, qui atteignent $\Gamma^{\alpha\beta}(\mathbf{V}) = \Gamma^{\alpha\beta}{}_{min}$, tout le problème étant justement de trouver celui est le moins coûteux du point de vue de E.

Rappelons que, dans le cas pratique, les images que nous utilisons seront seuillées : il s'agit de $\Phi^{\alpha*}$ et $\Phi^{\beta*}$, et non pas de Φ^α et Φ^β. Donc, comme elles fournissent des étiquettes noir-et-blanc, c'est la fonction discriminante simple $f_3(X, X')$ qu'il faudra prendre (elle vaut 0 si $X = X'$, et 1 si $X \neq X'$), et dans ce cas, $\Gamma^{\alpha\beta}$ est simplement une distance de Hamming. Cependant, on préférera garder provisoirement la formulation générale avec f et Φ (au §2.2 et §2.3), et supposer également que f est continue et dérivable (ce qui n'est pas le cas de f_3). Nous renvoyons au paragraphe 2.4 l'étude des variables discrètes.

2.2.c Energie totale

Au §2.2.b, la recherche d'une déformation minimale $\mathbf{V}$ a été reformulée en remplaçant l'exigence de coïncidence des images par un coût de non-coïncidence $\Gamma^{\alpha\beta}(\mathbf{V})$: ainsi, au lieu de minimiser $E(\mathbf{V})$ en ayant réalisé d'avance $\Gamma^{\alpha\beta}(\mathbf{V}) = \Gamma_{min}$, on cherchera à minimiser Γ en même temps que E. Le but n'est donc plus de favoriser les images au détriment de la structure, c'est à dire de reporter tout le coût dans E, mais plutôt de donner un rôle symétrique à l'image et à la structure, en tentant de trouver un compromis entre les deux. La situation peut être résumée par le schéma suivant, impliquant trois graphes étiquetés (figure 14; respectivement 14a, 14b et 14c) :

$$(G, \mathbf{C}^0, \mathbf{X}^{\alpha 0}) \leftarrow E \rightarrow (G, \mathbf{C}, \mathbf{X}^{\alpha 0}) \leftarrow \Gamma \rightarrow (G, \mathbf{C}, \mathbf{X}^\beta)$$

C'est le deuxième graphe (au centre) qui constitue le système déformable : il porte la même image que le premier, mais n'a pas la même structure (cette déformation étant mesurée par E), tandis qu'il a même structure que le troisième, mais pas nécessairement même image (cette non-coïncidence étant mesurée par Γ). Au §2.2.a, on s'était placé dans le cas où le deuxième et le troisième étaient nécessairement identiques, c'est à dire $\mathbf{X}^{\alpha 0} = \mathbf{X}^\beta$ (la déformation de Φ^α devait se conformer à Φ^β), puis, au §2.2.b, cette contrainte a été supprimée (la déformation de Φ^α n'est plus forcément conforme à Φ^β) et remplacée par le coût Γ.

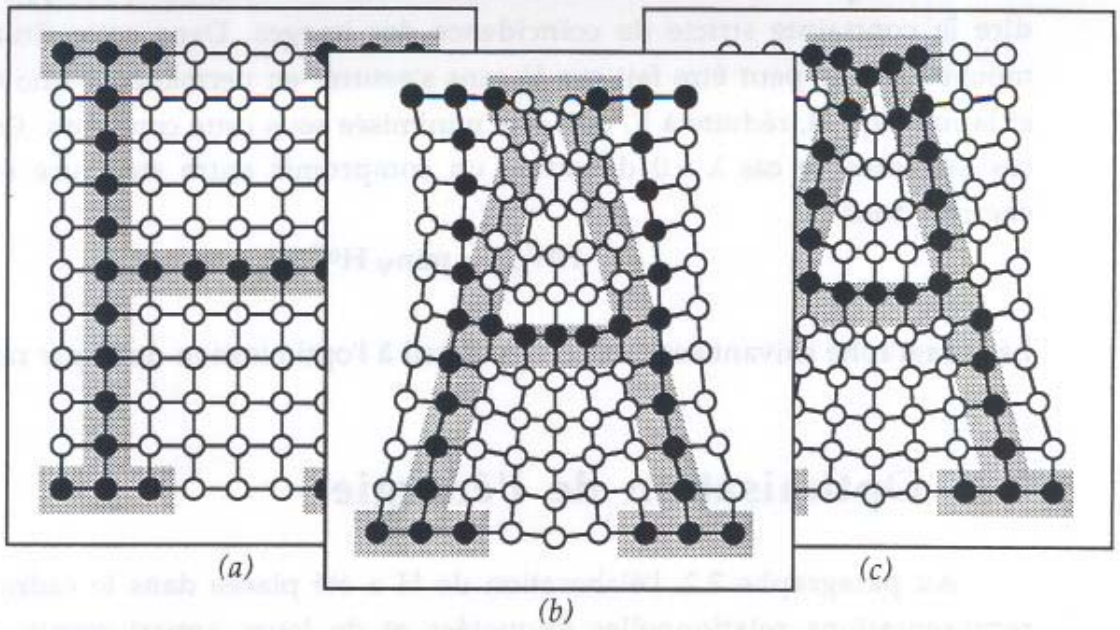


Figure 14 : illustration des deux composantes du coût total. Cette figure reproduit la situation de la figure 13, en y ajoutant un troisième graphe qui met en évidence les différences d'étiquettes avec le deuxième. *(a)* Représentation standard du "H". *(b)* Représentation difforme du "H". *(c)* Représentation conforme au "A". Le graphe (b) diffère du graphe (a) par la structure, mais possède les mêmes étiquettes de nœuds : la pénalité de déformation est E. Au contraire, le graphe (b) diffère du graphe (c) uniquement par les étiquettes de nœuds : la pénalité de non-coïncidence est Γ. La pénalité totale est alors : $H = E + \lambda \Gamma$. Remarquons que dans le cas de la figure 12, (b) et (c) étaient identiques, c'est à dire $\Gamma = 0$: cependant, E était plus grand qu'ici, et surpassait à lui tout seul la somme actuelle H (en effet, si la situation présente (b) est un état d'équilibre, elle réalise par définition $H = H_{min}$).

Finalement le deuxième graphe se trouve dans un état d'équilibre où il est écarté du premier par une déformation de structure, mais ne coïncide pas encore parfaitement avec le troisième au niveau des étiquettes des nœuds. Au total, il lui correspondra une énergie H, qui sera la somme de E et Γ pondérée par un coefficient $\lambda > 0$:

$$H_G(\mathbf{C}^0, \mathbf{C}, \mathbf{X}^{\alpha 0}, \mathbf{X}^{\beta}) = E_A(\mathbf{C}^0, \mathbf{C}) + \lambda\, \Gamma_O(\mathbf{X}^{\alpha 0}, \mathbf{X}^{\beta})$$

$$= \frac{1}{2} \sum_{<s,\,t> \in A} \left| C_{st} - C^0_{st} \right|^2 + \lambda \sum_{s \in O} f\left(X^{\alpha 0}_s, X^{\beta}_s\right)$$

En fonction de **V**, on a : $H^{\alpha\beta}(\mathbf{V}) = E(\mathbf{V}) + \lambda\, \Gamma^{\alpha\beta}(\mathbf{V})$, soit :

$$H^{\alpha\beta}(\mathbf{V}) = \frac{1}{2} \sum_{<s,\,t> \in A} \left| V_t - V_s - V^0_t + V^0_s \right|^2 + \lambda \sum_{s \in O} f\left(\Phi^{\alpha}(V^0_s), \Phi^{\beta}(V_s)\right)$$

Le coefficient λ règle le rapport de forces entre le terme de structure et le terme d'images. Si $\lambda = 0$, cela signifie qu'on ne se préoccupe pas des images, et qu'on cherche uniquement la meilleure structure vectorielle, c'est à dire celle qui minimise E dans l'absolu : on trouve alors $\mathbf{C} = \mathbf{C}^0$. Inversement, $\lambda = +\infty$ signifie que l'on est forcé de réaliser constamment $\Gamma = \Gamma_{min} = 0$ (ce qui suppose d'ailleurs que *f*

soit conçue pour avoir un minimum nul, comme c'est le cas pour f_1 ou f_3), c'est à dire la contrainte stricte de coïncidence des images. Dans cette situation, aucun mouvement ne peut être fait sur **V** sans s'assurer en permanence que $\Phi^\beta(\mathbf{V}) = X^{\alpha 0}$, et la fonction H, réduite à E, doit être minimisée sous cette condition. Entre ces deux cas extrêmes, le cas $\lambda > 0$ demande un compromis entre structure et image. On cherche donc :

$$H^{\alpha\beta}{}_{min} = \min_{\mathbf{V}} H^{\alpha\beta}(\mathbf{V})$$

Le paragraphe suivant est consacré en détail à l'optimisation de H par rapport à **V**.

2.3 Optimisation de l'énergie

Au paragraphe 2.2, l'élaboration de H a été placée dans le cadre général des représentations relationnelles étiquetées et de leurs appariements. Nous avons considéré la comparaison des caractères comme la confrontation de deux étiquetages fixés sur une même structure de graphe *G*, l'un devant varier pour être le moins distant possible de l'autre, tant du point de vue des vecteurs d'arêtes que des étiquettes de nœuds.

Il est alors possible d'interpréter H comme l'énergie d'un unique système déformable, à travers l'analogie mécanique suivante : un ensemble de mxp jetons est disposé sur un plan aux intersections de m lignes et p colonnes, et ces jetons sont reliés de proche en proche par des ressorts de telle sorte que l'ensemble forme une grille orthonormée (il s'agit en réalité de baguettes rigides elles-mêmes reliées à des ressorts, cf. §2.3.b). Les jetons sont donc les analogues des nœuds s, et les ressorts les analogues des arêtes <s,t>. Initialement, ce système mécanique est *au repos*, c'est à dire dans la position standard V_s^0 décrite plus haut, sur le plan de l'image Φ^α (figure 14a). Chaque jeton reçoit alors une couleur fixe $X_s^{\alpha 0}$, qui relève le niveau-de-gris du point de l'image où il se trouve, c'est à dire : $X_s^{\alpha 0} = \Phi^\alpha(V_s^0)$. Puis, le système est retiré du plan de Φ^α et déposé sur celui de Φ^β (figure 14b). Les jetons doivent alors se déplacer pour retrouver des zones d'image où les niveaux-de-gris sont comparables à la couleur qu'ils portent : à tout instant, leur position est repérée par un couple de coordonnées $V_s = (x_s, y_s)$, et ils se trouvent confrontés en cette position à un niveau-de-gris variable X_s^β issu de de l'image Φ^β, c'est à dire : $X_s^\beta = \Phi^\beta(V_s)$ (étiquettes de la figure 14c). Ils sont donc soumis à deux forces : une force élastique en provenance des jetons voisins, et une force d'attraction en provenance de l'image sur laquelle ils se déplacent. La première tend à rétablir la structure standard du maillage (grille orthonormée de la figure 14a), tandis que la deuxième tend à faire coïncider la couleur de chaque jeton avec celle de la zone qu'il parcourt (pour atteindre l'état de la figure 12b). On pourrait comparer $X_s^{\alpha 0}$ à une "charge" et X_s^β à un "potentiel", mais la nature précise de l'interaction entre ces deux grandeurs est donnée par la fonction *f*. Ainsi, le graphe *G* formé par les jetons et les ressorts se

déplace sous cette double influence, en s'étirant et se contractant, et finit par atteindre un état d'équilibre, qui est un compromis entre l'attraction exercée par l'image et les contraintes de structure. Ce processus dépend des deux images Φ^α et Φ^β de façon non-symétrique : la première est fixée par morceaux sur le graphe (étiquettes de nœuds), tandis que la deuxième est utilisée comme fonction du plan. C'est donc Φ^α qui se déforme sur Φ^β et non l'inverse.

Les variables du système sont les coordonnées mobiles des nœuds, $\mathbf{V} = (V_s)$: on définit une énergie totale $H(\mathbf{V})$, qui tient compte à la fois des distorsions internes de G, à travers une énergie élastique $E(\mathbf{V})$, et des différences d'étiquettes au niveau des nœuds individuels, à travers une énergie de couplage $\Gamma(\mathbf{V})$, soit : $H = E + \lambda \Gamma$. Si $\lambda = 0$, les forces d'image sont annulées, et le maillage relaxe rapidement vers un état de repos $\mathbf{V} \equiv \mathbf{V}^0$, sans tenir compte des étiquettes. Si $\lambda = +\infty$, le maillage est forcé de respecter en permanence la coïncidence des étiquettes, et la minimisation de E doit s'effectuer sous cette contrainte (figure 12b) : les nœuds-jetons sont alors placés dès le départ sur des zones de Φ^β où ils trouvent exactement $\Phi^\beta(V_s) = X_s^{\alpha 0}$, et leurs déplacements doivent rester limités à ces équipotentielles. Signalons que cette option sera celle que nous prendrons pour traiter les images seuillées binaires Φ^* : ainsi, les nœuds caractérisés par des étiquettes noires $X_s^{\alpha *0}$ seront cantonnés dans les zones noires de $\Phi^{\beta *}$, et les nœuds caractérisés par des étiquettes blanches resteront dans les zones blanches. Dans le cas réel continu, en revanche, la contrainte stricte d'image n'est pas souhaitable, car elle bloquera la minimisation de H dans de nombreuses configurations locales à cause d'équipotentielles restreintes : il faut alors autoriser des mouvements discontinus de nœuds pour tenter de traverser le paysage d'énergie.

Dans ce paragraphe 2.3, nous conserverons le cas réel continu avec $\lambda > 0$. La dérivée de l'énergie H par rapport à une position $V_s = (x_s, y_s)$ a pour analogue la *force locale* qui s'exerce sur le nœud s (à une inversion de signe près) : celle-ci se décompose en deux forces élémentaires provenant de E et Γ. Nous cherchons à présent les conditions d'équilibre de ce système mécanique, c'est à dire l'arrêt du mouvement des nœuds, correspondant à l'annulation mutuelle de ces forces. Pour cela, les nœuds du graphe seront visités séquentiellement, et leur position locale améliorée dans la direction d'une diminution de l'énergie. Chaque déplacement d'un nœud s devra donc tenir compte aussi bien de son étiquette que de ses interactions élastiques avec ses voisins dans G.

2.3.a Forces locales et pénalités locales

Nous préparons ici le calcul des grandeurs qui seront utiles à la minimisation de H par rapport à $\mathbf{V}$: il s'agit d'une part des forces locales, qui sont issues des dérivées partielles de H (variations infinitésimales de l'énergie), et d'autre part des pénalités locales, qui sont les intégrales des premières sur le déplacement d'un nœud individuel (variations finies de l'énergie). Minimiser les deuxièmes nécessite

donc d'annuler les premières en chaque nœud : on obtient alors l'immobilisation du graphe. Bien sûr, l'état stable ainsi atteint n'est pas forcément celui de plus faible énergie globale H, mais on se contentera en pratique d'une stabilisation dans un minimum local (dans les conditions discrétisées du §2.4, nous choisirons une descente directe en énergie, c'est à dire une relaxation "à température 0").

On rappelle que **V** est un ensemble de vecteurs du plan, $V_S = (x_S, y_S)$, donc que les variations de H par rapport à **V** s'expriment comme autant de gradients partiels, calculés par rapport à chacun de ces vecteurs, $(\text{grad}_S H) = (\partial H / \partial V_S)$:

$$\frac{\partial H^{\alpha\beta}}{\partial V_S} = \frac{\partial E}{\partial V_S} + \lambda \frac{\partial \Gamma^{\alpha\beta}}{\partial V_S}$$

avec :

$$\frac{\partial E}{\partial V_S} = 2 \sum_{\substack{t \in O; \\ <s,\,t> \in A}} \left(V_S - V_t - V_S^0 + V_t^0 \right)$$

et :

$$\frac{\partial \Gamma^{\alpha\beta}}{\partial V_S} = \frac{\partial f}{\partial X'} \left(\Phi^\alpha(V_S^0), \Phi^\beta(V_S) \right) . \nabla \Phi^\beta(V_S)$$

$\partial f / \partial X'$ représente la dérivée par rapport à la deuxième variable de la fonction $f(X, X')$ (grandeur scalaire). C'est dans $\nabla \Phi^\beta(V_S) = d\Phi^\beta / dV_S$ que se trouve la nature vectorielle de $\partial \Gamma / \partial V_S$: il s'agit du gradient de l'image Φ^β au point de coordonnées $V_S = (x_S, y_S)$. De son côté, le terme $\partial E / \partial V_S$ est la somme de contributions vectorielles dépendant des voisins du nœud s (dans le calcul de cette dérivée, le nœud s doit être identifié aussi bien comme "s" que comme "t" dans la somme, ce qui produit un facteur 2 supplémentaire à cause de la symétrie du graphe). Les quantités opposées à ces dérivées, $-\partial E / \partial V_S$ et $-\partial \Gamma / \partial V_S$, représentent respectivement la *force élastique* et la *force d'image* qui s'exercent sur le nœud s. Ces forces locales fournissent les variations infinitésimales de l'énergie occasionnées par un déplacement élémentaire dV_S.

Nous introduisons à présent la notion de *pénalité locale*, qui correspond aux variations finies de l'énergie occasionnées par un déplacement macroscopique du nœud s. Dans ce but, on définit les grandeurs suivantes attachées à s :

$$h_S(\mathbf{V}) = e_S(\mathbf{V}) + \lambda\, g_S(\mathbf{V})$$

avec :

$$e_S(\mathbf{V}) = \sum_{\substack{t \in O; \\ <s,t> \in A}} \left| V_t - V_S - V_t^0 + V_S^0 \right|^2$$

et :

$$g_S(\mathbf{V}) = f\left(\Phi^\alpha(V_S^0), \Phi^\beta(V_S) \right)$$

(remarquer l'absence de facteur 1/2 dans e_S). A une constante près, la pénalité élastique e_S et la pénalité d'image g_S représentent l'opposé du travail fourni par les forces locales $-\partial E/\partial V_S$ et $-\partial \Gamma/\partial V_S$ à l'occasion d'un déplacement individuel du nœud s. Ce sont donc les intégrales des forces locales :

$$h_S(\mathbf{V}) = \int \frac{\partial H}{\partial V_S}\, dV_S + \text{cste}$$

c'est à dire, plus précisément :

$$h_S(\mathbf{V};\, V_S=(x,y)) = \int_{[x',y']}^{[x,y]} \frac{\partial H}{\partial V_S}\, dV_S + h_S(\mathbf{V};\, V_S=(x',y'))$$

où (x', y') sont les anciennes coordonnées de s et (x, y) ses coordonnées actuelles (la notation "$\mathbf{V}$; $V_S=(x,y)$" signifie que V_S prend une valeur particulière (x, y) tandis que les positions des autres nœuds $(V_t)_{t\neq s}$ sont fixées). Ainsi, les pénalités qui viennent s'ajouter aux quantités globales H, E et Γ à la suite d'une variation isolée de V_S ne sont pas mesurées directement par les quantités locales h_S, e_S et g_S, mais par leurs variations :

$$H(\mathbf{V};\, V_S=(x,y)) - H(\mathbf{V};\, V_S=(x',y')) = h_S(\mathbf{V};\, V_S=(x,y)) - h_S(\mathbf{V};\, V_S=(x',y'))$$

et on a aussi : $\partial H/\partial V_S = \partial h_S/\partial V_S$. Remarquons par ailleurs que la somme des (h_S) ne redonne pas H, donc que ces quantités ne sont pas des contributions indépendantes à l'énergie totale.

Plus précisément, ce sont les (e_S) qui ne forment pas une partition de E, car les domaines d'influence des différents nœuds sont en recouvrement : ces domaines contiennent toutes les arêtes rattachées à s, et ces arêtes sont communes aux voisins t. C'est pourquoi la somme des (e_S) vaut le double de E (cf. formule du §2.2.a) :

$$E(\mathbf{V}) = \sum_{s \in O} \frac{1}{2}\, e_S(\mathbf{V})$$

(donc la cause n'est pas la double orientation <s,t>/<t,s>, mais le fait qu'une arête, même orientée, engage toujours deux nœuds). En définissant une pénalité locale, notre intention est de répartir une fonction-coût globale sur des valeurs individuelles attachées à chaque nœud. On se trouve donc devant une alternative : soit on décompose E en contributions élémentaires dont la somme redonne exactement E, et dans ce cas on doit partitionner arbitrairement l'ensemble des arêtes en fonction des nœuds; soit on préfère compter tout ce qui dépend du nœud s, afin de disposer d'une pénalité e_S qui mesure vraiment l'influence d'un déplacement élémentaire de V_S sur l'énergie globale, et dans ce cas on a forcément des recouvrements d'un nœud à l'autre entre les groupes d'arêtes qui leur sont rattachés, donc une contribution totale plus grande que E. Par contre, lorsqu'il s'agit de dériver ces quantités par rapport à V_S, le facteur 1/2 disparaît :

$$\frac{\partial E(\mathbf{V})}{\partial V_S} = \frac{1}{2}\frac{\partial e_S(\mathbf{V})}{\partial V_S} + \frac{1}{2}\sum_{\substack{t \in O;\\ <s,t> \in A}} \frac{\partial e_t(\mathbf{V})}{\partial V_S} = \frac{\partial e_S(\mathbf{V})}{\partial V_S}$$

et on a finalement :

$$\frac{\partial H(\mathbf{V})}{\partial V_s} = \frac{\partial E(\mathbf{V})}{\partial V_s} + \lambda \frac{\partial \Gamma(\mathbf{V})}{\partial V_s} = \frac{\partial e_s(\mathbf{V})}{\partial V_s} + \lambda \frac{\partial g_s(\mathbf{V})}{\partial V_s} = \frac{\partial h_s(\mathbf{V})}{\partial V_s}$$

Donc la fonction locale $h_s(\mathbf{V})$ répercute exactement les variations de $H(\mathbf{V})$ selon V_s, bien que cette dernière ne soit pas directement déductible de la première par une simple somme sur s.

2.3.b Equilibre local élastique

Disposant des forces locales et des pénalités locales attachées au nœud s, notre but est à présent de trouver un *équilibre local* pour ce nœud, c'est à dire l'optimisation de sa position V_s pendant que les autres positions $(V_t)_{t\neq s}$ sont maintenues fixes : V_s se stabilisera en un point $V_{s\,min}$ qui correspond par définition à une force $-\partial H/\partial V_s$ nulle, donc en général à une énergie locale h_s minimale. Dans un premier temps, on étudiera isolément les conditions d'équilibre des forces élastiques : donc les forces d'image seront provisoirement ignorées en posant $\lambda = 0$, puis, au §2.3.c, nous traiterons l'équilibre complet, avec $\lambda > 0$.

Examinons la situation locale autour de s. On considère le petit morceau de graphe dont les caractéristiques géométriques dépendent de la position de s. Ce sous-graphe est constitué des quatre voisins de s, que l'on notera ici : t_1, t_2, t_3, t_4 (ils ne seraient que 3 ou 2 au bord de la grille, mais nous gardons le cas général), et des huit arêtes qui relient ces voisins à s : $\langle s,t_1\rangle$, $\langle s,t_2\rangle$, $\langle s,t_3\rangle$, $\langle s,t_4\rangle$ ainsi que $\langle t_1,s\rangle$, $\langle t_2,s\rangle$, $\langle t_3,s\rangle$, $\langle t_4,s\rangle$. L'ensemble a la forme d'une étoile à quatre branches centrée en s. Les coordonnées de ces nœuds sur le plan sont : V_s, V_{t1}, V_{t2}, V_{t3}, V_{t4}. Dans la représentation standard $\mathbf{V} = \mathbf{V}^0$ ces composants forment une "croix orthonormée" (branches de longueur 1, angles droits en s), tandis qu'avec un $\mathbf{V}$ quelconque, cette croix est déformée : ses branches sont étirées ou rétrécies, et les angles droits sont écartés ou refermés (figure 15).

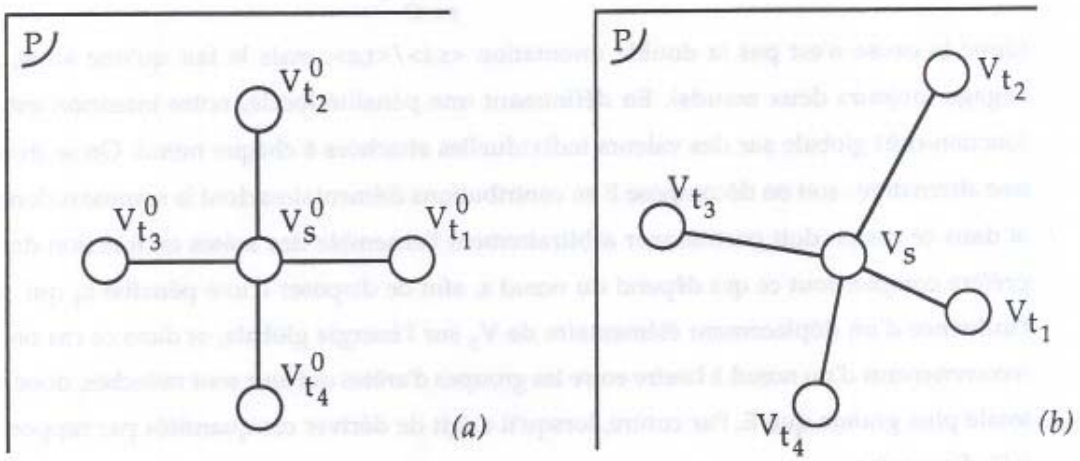


Figure 15 : Elément de graphe autour d'un nœud s. Les quatre voisins de ce nœud sont notés t_1, t_2, t_3, t_4 (donc si $s=s_{ij}$, il s'agit respectivement de : $s_{i\,j+1}$, $s_{i-1\,j}$, $s_{i\,j-1}$ et $s_{i+1\,j}$). Ils sont reliés à s par 8 arêtes: $\langle s,t_1\rangle$, $\langle t_1,s\rangle$, ... (l'orientation des arêtes n'est pas précisée sur la figure). *(a)* Au départ, les nœuds se trouvent sur les positions $\mathbf{V}^0$, et le sous-graphe à la forme d'une croix orthonormée. *(b)* Puis le déplacement des nœuds en d'autres positions $\mathbf{V}$ provoque la déformation de cette croix.

Dans ce contexte, la force élastique locale qui s'exerce sur s, $-\partial E/\partial V_s$ (et qui est donnée également par $-\partial e_s/\partial V_s$) s'écrit :

$$-\frac{\partial E}{\partial V_s} = -2\sum_{i=1}^{4}\left(V_s - V_{t_i} - V_s^0 + V_{t_i}^0\right)$$

Donc, les positions des quatre voisins $\{V_{t_i}\}$ étant fixées, ce terme est équivalent à quatre forces de rappel $\{F_1, F_2, F_3, F_4\}$ qui s'exerceraient sur s à travers quatre ressorts de longueur à vide 0 et de coefficient de raideur 2 :

$$-\frac{\partial E}{\partial V_s} = \sum_{i=1}^{4} F_i, \quad \text{avec} \quad F_i = -2\,(V_s - V_{t_i}')$$

où les positions $V_{t_i}' = V_{t_i} + V_s^0 - V_{t_i}^0$ représentent les points d'attache de ces ressorts (figure 16d).

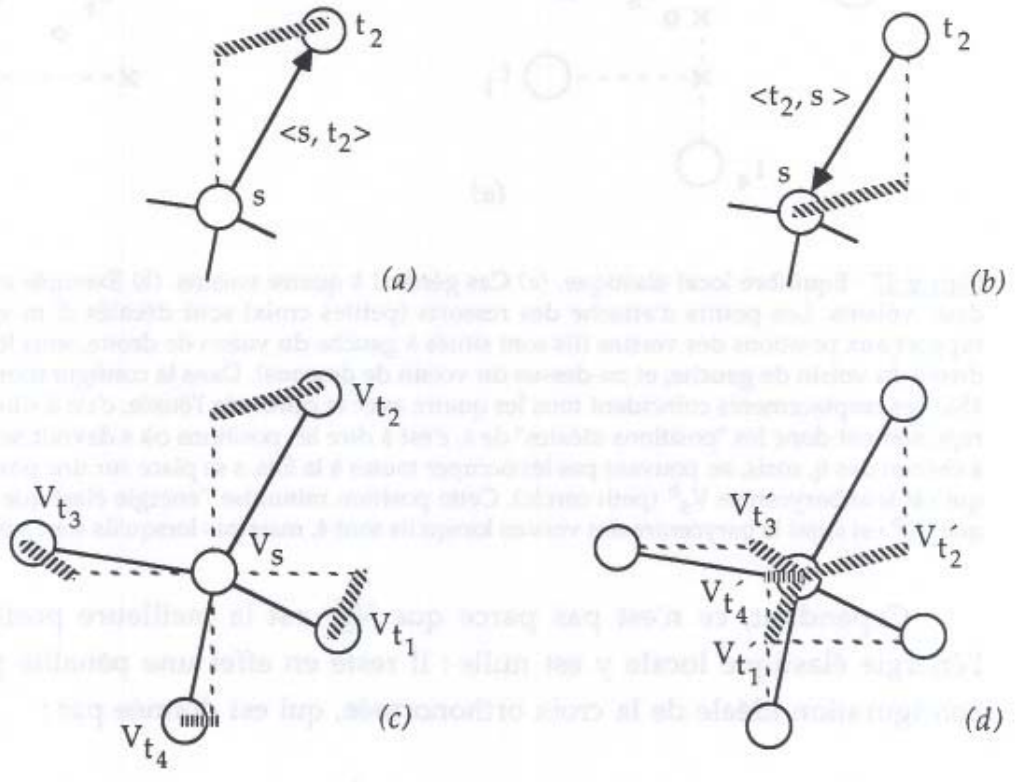


Figure 16 : Illustration des forces élastiques exercées sur le nœud s par ses voisins. *(a)* Exemple de l'arête $<s,t_2>$: cette arête n'est pas directement comparable à un ressort reliant s à t_2, mais il faut d'abord imaginer une baguette rigide de longueur 1 fixée au nœud s (qui correspond à ce qu'était $<s,t_2>$ dans l'état standard), puis un ressort de longueur à vide 0 attaché entre l'extrémité de cette baguette et le nœud t_2. *(b)* Même chose, dans le sens contraire, pour l'arête $<t_2,s>$. Par conséquent, du point de vue du nœud s (son voisin t_2 étant immobile) le ressort (a) conjugue son action à celle du ressort (b), et l'ensemble équivaut à une seule force élastique d'intensité double. *(c)* Schéma du (a) étendu aux quatre voisins. *(d)* Schéma du (b) étendu aux quatre voisins. Finalement, l'union de (c) et (d) est équivalente à la situation (d) seule, avec des ressorts de raideur 2.

Par conséquent, l'équilibre de ces forces est obtenu pour :

$$-\frac{\partial E}{\partial V_s} = 0 \iff V_s = \frac{1}{4}\sum_{i=1}^{4}\left(V_{t_i} + V_s^0 - V_{t_i}^0\right) = \frac{1}{4}\sum_{i=1}^{4} V_{t_i}'$$

Ainsi, dans le cas où on ne tient pas compte des images ($\lambda = 0$), on obtient directement l'expression analytique de la meilleure position locale V_s pour s, c'est à dire celle qui minimise l'énergie élastique seule, étant donné les positions des voisins de s. C'est l'existence de cette formule analytique directe qui a motivé le choix de la forme quadratique pour E (suggestion de D. Geman). L'interprétation de cette formule est claire : il s'agit du *barycentre* des quatre points d'attache des ressorts, $\{V_{t1}', V_{t2}', V_{t3}', V_{t4}'\}$, barycentre que l'on notera dans toute la suite : V_s^b (figure 17).

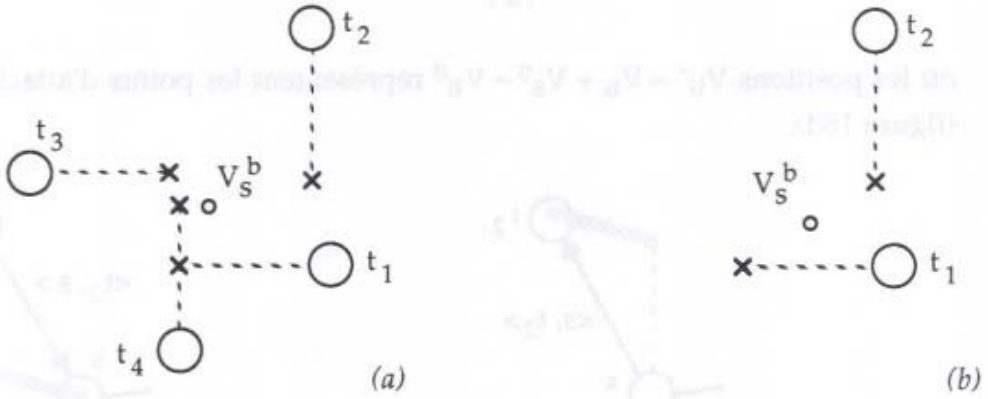


Figure 17 : Equilibre local élastique. *(a)* Cas général à quatre voisins. *(b)* Exemple où s ne possède que deux voisins. Les points d'attache des ressorts (petites croix) sont décalés d'un vecteur unitaire par rapport aux positions des voisins (ils sont situés à gauche du voisin de droite, sous le voisin de dessus, à droite du voisin de gauche, et au-dessus du voisin de dessous). Dans la configuration parfaite $\mathbf{V}^0$ (figure 15a) ces emplacements coïncident tous les quatre avec le centre de l'étoile, c'est à dire en V_s^0. Ces points représentent donc les "positions idéales" de s, c'est à dire les positions où s devrait se trouver par rapport à chacun des t_i, mais, ne pouvant pas les occuper toutes à la fois, s se place sur une position de compromis, qui est leur barycentre V_s^b (petit cercle). Cette position minimise l'énergie élastique locale (remarquons que V_s^b est aussi le barycentre des voisins lorsqu'ils sont 4, mais pas lorsqu'ils sont 3 ou 2).

Cependant, ce n'est pas parce que V_s^b est la meilleure position pour s que l'énergie élastique locale y est nulle : il reste en effet une pénalité par rapport à la configuration idéale de la croix orthonormée, qui est donnée par :

$$e_s(\mathbf{V}; V_s=V_s^b) - e_s(\mathbf{V}^0; V_s=V_s^0) = \sum_{i=1}^{4}\left|V_{t_i} - V_s^b - V_{t_i}^0 + V_s^0\right|^2 - 0$$
$$= \sum_{i=1}^{4}\left|V_{t_i}' - V_s^b\right|^2$$

Cette quantité est donc la pénalité locale *minimale* en s (dans le contexte où les autres nœuds sont fixés), que s réalise en se plaçant en V_s^b.

Dans le cas général où s possède un nombre de voisins n_S, avec $2 \le n_S \le 4$ (c'est le nombre de $t \in O$, tels que $<s,t> \in A$), la position qui minimise localement E est donnée par :

$$V_s^b = \frac{1}{n_S} \sum_{\substack{t \in O; \\ <s,t> \in A}} \left(V_t + V_s^0 - V_t^0\right)$$

Donc, partant d'une configuration quelconque **V**, le système va relaxer nœud après nœud vers un état qui satisfera à tous les équilibres locaux à la fois, c'est à dire à toutes les équations barycentriques : $\forall\, s \in O,\ V_S = V_S{}^b$. Pour l'atteindre, il suffit d'itérer le même déplacement élémentaire en chaque s : pendant que les autres nœuds sont maintenus en place, on lâche le nœud s, qui "saute" spontanément au barycentre déterminé par ses voisins. A force de répéter cette opération de nœud en nœud, le graphe finit par s'immobiliser (à la limite asymptotique), chaque nœud se trouvant dans une position satisfaisante par rapport aux autres. Cependant, si nous continuons à ignorer l'influence des images, cette perspective de minimisation globale est sans grand intérêt, puisque nous en connaissons déjà la solution : il s'agit évidemment de $V_S = V_S{}^0$ pour tout s, à une translation uniforme près (du point de vue des seules forces élastiques, la meilleure structure est la structure orthonormée, par définition de E). Il faut donc maintenant tenir compte de la force d'image, en reprenant $\lambda > 0$.

2.3.c Equilibre local complet

L'analyse du §2.3.b a permis de développer l'analogie mécanique fondée sur les ressorts, et a montré que le minimum local qu'ils désignent est un simple barycentre : nous allons à présent greffer l'action du terme Γ sur ces résultats, en calculant la déviation qu'il provoque par rapport à $V_S{}^b$. L'équilibre local complet en s est obtenu pour :

$$\frac{\partial H^{\alpha\beta}}{\partial V_S} = 0 \iff -\frac{\partial E}{\partial V_S} = \lambda \frac{\partial \Gamma^{\alpha\beta}}{\partial V_S}$$

c'est à dire :

$$-2 \sum_{\substack{t \in O; \\ <s,t> \in A}} \left(V_S - V_t - V_S^0 + V_t^0\right) = \lambda \frac{\partial f}{\partial X'}\left(X_S^{\alpha 0}, \Phi^\beta(V_S)\right) . \nabla \Phi^\beta(V_S)$$

(où $X_S{}^{\alpha 0}$ désigne l'étiquette fixe $\Phi^\alpha(V_S{}^0)$). Donc, ici, les forces élastiques en s ne doivent plus fournir une somme nulle, mais elles doivent compenser la force d'attraction due au potentiel d'image, dont l'intensité est modulée par λ. Par conséquent, à cause de l'influence des images, la position optimale du nœud s est *déviée* par rapport au barycentre $V_S{}^b$ calculé précédemment. Dans toute la suite, on prendra ce barycentre comme repère local pour s, en posant : $V_S = V_S{}^b + v_S$. Le

changement de variable de V_s en v_s ramène alors l'expression des forces élastiques (membre de gauche de l'équation d'équilibre) à la forme réduite suivante :

$$-2 \sum_{\substack{t \in O; \\ <s,t> \in A}} \left(V_s^b + v_s - V_t - V_s^0 + V_t^0 \right) = -2\, n_s v_s$$

où n_s désigne, comme précédemment, le nombre de nœuds t voisins de s ($n_s = 4$ dans le cas le plus courant). Donc l'équilibre local complet est exprimé par l'équation :

$$-2\, n_s v_s = \lambda \frac{\partial f}{\partial X'} \left(X_s^{\alpha 0}, \Phi^\beta(V_s^b + v_s) \right) . \nabla \Phi^\beta (V_s^b + v_s)$$

Pour apprécier l'influence du membre de droite, prenons par exemple : $f(X, X') = (X - X')^2$. Dans ce cas, $\partial f / \partial X' = 2(X' - X)$, et l'équation devient : $-2\, n_s v_s = 2\lambda(\Phi^\beta(V_s^b + v_s) - X_s^{\alpha 0}).\nabla\Phi^\beta(V_s^b + v_s)$. Donc, s'il n'y avait pas la force de rappel élastique $-2\, n_s v_s$, l'annulation de la force d'image seule fournirait un point $V_s^b + v_s$ en lequel les étiquettes coïncideraient : $\Phi^\beta(V_s^b + v_s) = X_s^{\alpha 0}$, ou bien en lequel le gradient $\nabla\Phi^\beta$ serait nul (ce dernier cas pouvant se produire dans une zone d'image de Φ^β avec extremum local). Avec $-2\, n_s v_s$, cette l'égalité traduit alors un compromis entre le point $V_s^b + v_s$ où la force d'image est nulle, et le point V_s^b où la force élastique est nulle.

Cette équation n'étant pas soluble analytiquement, la minimisation locale par rapport à v_s pourrait s'effectuer par la méthode du gradient, c'est à dire en déplaçant v_s par sauts discrets : $\Delta v_s = -\gamma\,(2 n_s v_s + \partial \Gamma^{\alpha\beta} / \partial v_s)$. Cette méthode nécessite la connaissance du gradient de l'image $\nabla\Phi^\beta$ (qui sera un tableau de pixels dans les simulations numériques, comme l'image elle-même), et elle suppose aussi que $f(X, X')$ soit dérivable. Cependant, en pratique, nous utiliserons des images déjà seuillées noir-et-blanc $\Phi^{\beta *}$: dans ce cas, on ne peut pas en extraire de gradient, et la conséquence est aussi que $f(X, X')$ n'est pas dérivable (elle ne prendra que deux valeurs : 0 si $X = X'$, et 1 sinon). Remarquons qu'il serait possible de recréer des images en niveaux-de-gris par une opération de lissage sur $\Phi^{\beta *}$: on disposerait alors d'un paysage Φ^β continu (dans les limites de la numérisation), et on aurait ainsi de meilleures chances de relaxer vers une déformation minimale. En effet, les nuances de niveaux-de-gris ont l'avantage de "guider" le nœud vers les zones favorables à son étiquette grâce à une pente finie, contrairement aux paliers uniformément noirs ou blancs. Toutefois, la petite taille des images dont nous disposons, et la relative simplicité du problème que nous traiterons en exemple (cf. chapitre 3) ne méritent pas ce supplément de calculs. Par conséquent, à la place des forces et des gradients, il sera préférable de raisonner dans la suite en termes de *pénalités*, que nous chercherons à minimiser séquentiellement.

Dans ce but, on doit calculer les variations de e_s et g_s occasionnées par un mouvement fini du nœud s. Tout d'abord, la déviation v_s à partir du barycentre V_s^b

apporte un supplément de pénalité élastique par rapport à la valeur minimale qu'offrait le barycentre :

$$\Delta e_s(v_s) = e_s(\mathbf{V}; V_s = V_s^b + v_s) - e_s(\mathbf{V}; V_s = V_s^b)$$

$$= \sum_{\substack{t \in O; \\ \langle s,t \rangle \in A}} \left| V_t - (V_s^b + v_s) - V_t^0 + V_s^0 \right|^2 - \left| V_t - V_s^b - V_t^0 + V_s^0 \right|^2$$

$$= n_s |v_s|^2$$

Le terme de droite dans la différence représente la pénalité au barycentre V_s^b (qui avait été calculée au §2.3.b avec quatre voisins) : c'est une constante du point de vue de v_s, et on voit qu'il est utile de la retrancher de la pénalité de $V_s^b + v_s$, pour obtenir l'expression simple : $\Delta e_s(v_s) = n_s |v_s|^2$. Ce supplément de pénalité est un potentiel élastique local en s, dont l'opposé du gradient redonne la force $-2n_s v_s$: il s'agit d'un paraboloïde dont le minimum se trouve logiquement au barycentre, en $v_s = 0$ (figure 18a). Remarquons que ce potentiel est isotrope, bien que les positions des voisins ne présentent a priori aucune symétrie particulière : Δe_s ne dépend que de la norme de v_s, pas de sa direction.

D'autre part, la déviation v_s provoque également une variation dans la pénalité d'image :

$$\Delta g_s(v_s) = g_s(\mathbf{V}; V_s = V_s^b + v_s) - g_s(\mathbf{V}; V_s = V_s^b)$$

$$= f\left(\chi_s^{\alpha 0}, \Phi^\beta(V_s^b + v_s)\right) - f\left(\chi_s^{\alpha 0}, \Phi^\beta(V_s^b)\right)$$

Mais contrairement à $\Delta e_s(v_s)$ qui est toujours positif dès que v_s s'éloigne de 0, $\Delta g_s(v_s)$ pourra également prendre des valeurs négatives, sauf si la coïncidence entre l'étiquette $X_s^{\alpha 0}$ et la zone d'image $\Phi\beta(V_s^b + v_s)$ est justement la meilleure au barycentre, c'est à dire si est f minimale en $v_s = 0$. Donc, dans la plupart des cas, le nœud s devra s'éloigner du barycentre pour se rapprocher d'une coïncidence entre son étiquette et l'image (figure 18b). Ainsi, avec $\Delta g_s(v_s) < 0$, v_s est à la fois attiré en dehors de 0 par les forces d'image, et vers 0 par les forces élastiques (figure 18) (si au contraire $\Delta g_s(v_s) \geq 0$ pout tout v_s, alors il est clair que la meilleure position reste au barycentre $v_s = 0$).

Finalement, la recherche du v_s réalisant l'équilibre revient à minimiser la variation de pénalité locale complète par rapport au barycentre, c'est à dire la quantité :

$$\Delta h_s(v_s) = \Delta e_s(v_s) + \lambda \Delta g_s(v_s)$$

qui vaut :

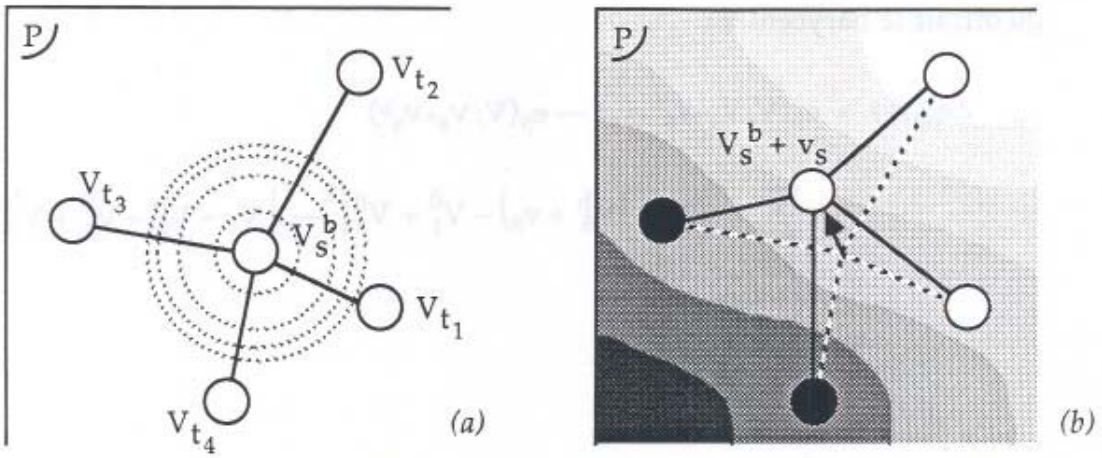


Figure 18 : Equilibre local des forces élastiques et des forces d'image. *(a)* Du point de vue des forces élastiques seules, la meilleure position pour le nœud s se trouve au barycentre $V_s = V_s^b$ (figure 17), c'est à dire en $v_s = 0$. En ce point, s réalise le minimum de la pénalité élastique locale, qui vaut $n_s | v_s |^2$ + cste (symbolisé ici par des cercles concentriques). *(b)* Avec l'influence de l'image, ce nœud est écarté de la position barycentrique et se déplace vers des zones du plan qui correspondent mieux à son étiquette. Donc ce déplacement en $V_s^b + v_s$ est motivé par une diminution de la pénalité d'image Δg_s et provoque nécessairement une augmentation de l'énergie élastique Δe_s : au total, il doit correspondre à une *diminution* de $\Delta h_s = \Delta e_s + \lambda \Delta g_s$. On rappelle que les étiquettes portées par les nœuds sont issues de l'image Φ^α (ici, elles sont blanches et noires pour simplifier), tandis que les caractéristiques du plan sont celles de l'image Φ^β. Dans cette illustration, le nœud s est blanc et subit l'attraction exercée par les zones claires.

$$\Delta h_s(v_s) = n_s | v_s |^2 + \lambda f\left(X_s^{\alpha 0}, \Phi^\beta(V_s^b + v_s)\right) - \lambda f\left(X_s^{\alpha 0}, \Phi^\beta(V_s^b)\right)$$

$$= n_s | v_s |^2 + \lambda f\left(X_s^{\alpha 0}, \Phi^\beta(V_s^b + v_s)\right) - \text{cste}$$

On notera $v_{s\,min}$ le point rechecherché :

$$v_{s\,min} = \text{argmin}\, \Delta h_s(v_s) \iff \forall v_s,\ \Delta h_s(v_s) \leq \Delta h_s(v_{s\,min})$$

(il s'agit de la forme intégrale de l'équation d'équilibre des forces écrite plus haut). Il est facile de voir que ce minimum existe toujours, dans la mesure où *f est bornée* (on évitera donc un *f* hyperbolique ou logarithmique) : plus précisément, il existe nécessairement un "éloignement limite", $| v_s |_{lim}$, à partir duquel l'augmentation du terme élastique l'emporte sur la diminution du terme d'image, et donc on a : $0 < | v_{s\,min} | < | v_s |_{lim}$.

2.3.d Equilibre global

Finalement, la relaxation vers un état d'équilibre global pour l'ensemble du graphe sera obtenue en visitant les nœuds les uns après les autres : au moment de la visite de s, les autres nœuds seront immobilisés pour permettre à v_s d'atteindre $v_{s\,min}$. Donc, si on note τ l'indice des temps, le processus de relaxation sera dirigé par les équations suivantes :

$$V_S(\tau) = V_S^b(\tau) + v_{S\,min}(\tau)$$

avec :

$$V_S^b(\tau) = \frac{1}{n_S} \sum_{\substack{t \in O; \\ <s,t> \in A}} \left(V_t(\tau-1) + V_S^0 - V_t^0\right)$$

et en l'absence de contrainte stricte sur les étiquettes, l'initialisation sera simplement choisie sur les positions standard : $\forall\, s \in O$, $V_S(0) = V_S^0$. Ainsi, à chaque instant τ, la nouvelle position du nœud s est guidée par deux influences : d'une part, les positions qu'avaient les plus proches voisins à l'instant précédent $\tau-1$, et qui désignent le barycentre V_S^b; d'autre part, l'attraction de l'image qui éloigne s de V_S^b d'un vecteur $v_{S\,min}$, cet écart étant le résultat de la minimisation d'un potentiel local $\Delta h_S(v_S)$ qui ne dépend pas explicitement des voisins (V_t) mais seulement de leur nombre, n_S. Dans une relaxation séquentielle, τ est incrémenté de 1 à chaque visite de nœud, tandis que dans une relaxation parallèle, τ n'est incrémenté de 1 qu'après visite de tous les nœuds. Pour trouver $v_{S\,min}$, on peut appliquer la méthode de Monte-Carlo, en tirant au hasard des positions $V_S^b+v_S$, et en choisissant la moins coûteuse (avec éventuellement une probabilité de Métropolis). Cependant, on verra au §2.4 qu'il est possible d'effectuer une recherche exhaustive de $v_{S\,min}$ sur un plan discrétisé, grâce à l'existence de la limite $|v_S|_{lim}$.

Jusqu'à présent, nous nous sommes placés dans la situation théorique où les grandeurs décrivaient des domaines continus : nous abordons à présent au §2.4 les conséquences de la discrétisation de ce modèle dans les simulations numériques, en précisant l'algorithme d'optimisation des positions de nœuds.

2.4 Pratique

2.4.a Discrétisation des images

Dans les simulations numériques, le plan réel continu est nécessairement discrétisé, et les images Φ sont donc codées sous formes de *tableaux de pixels*. Avec un quadrillage suffisamment fin, c'est à dire une densité de pixels élevée, on peut produire une bonne approximation de la situation réelle traitée plus haut (figure 19a). Comme précédemment, l'unité de longueur correspond aux arêtes non-déformées, et dans ce contexte, les coordonnées des nœuds sont en général des nombres rationnels, multiples d'une certaine fraction 1/c. Le nombre entier c est le coefficient de résolution du quadrillage : plus c est grand, plus la résolution est fine, et plus on se rapproche du cas réel idéal. Au contraire, lorsque c = 1, le quadrillage est du même ordre de grandeur que les arêtes du graphe, et il n'existe pas de position intermédiaire entre les coordonnées entières (figure 19b).

Il se trouve que les images numérisées que nous traiterons dans l'application

ont été renormalisées en carrés de 16x16 pixels noirs ou blancs (figure 21, chapitre 3). Par conséquent, ce nombre de pixels étant relativement petit, il sera difficile de le considérer comme un quadrillage de résolution plus fine que le maillage du graphe. Bien sûr, on pourrait envisager c=2 ou c=4 (une arête valant alors 2 ou 4 pixels), ce qui donnerait respectivement un graphe de 8x8 ou 4x4 nœuds : cependant, à ce niveau, il sera préférable d'utiliser tous les pixels pour éviter des ambiguïtés dans la définition de l'étiquetage standard $\mathbf{X}^{\alpha 0}$ porté par le graphe. Ainsi, dans toute la suite on posera c=1 : le graphe *G* contiendra 16x16 nœuds, et dans l'état non-déformé, chaque pixel sera occupé par un nœud (densité maximale du graphe sur l'image numérisée).

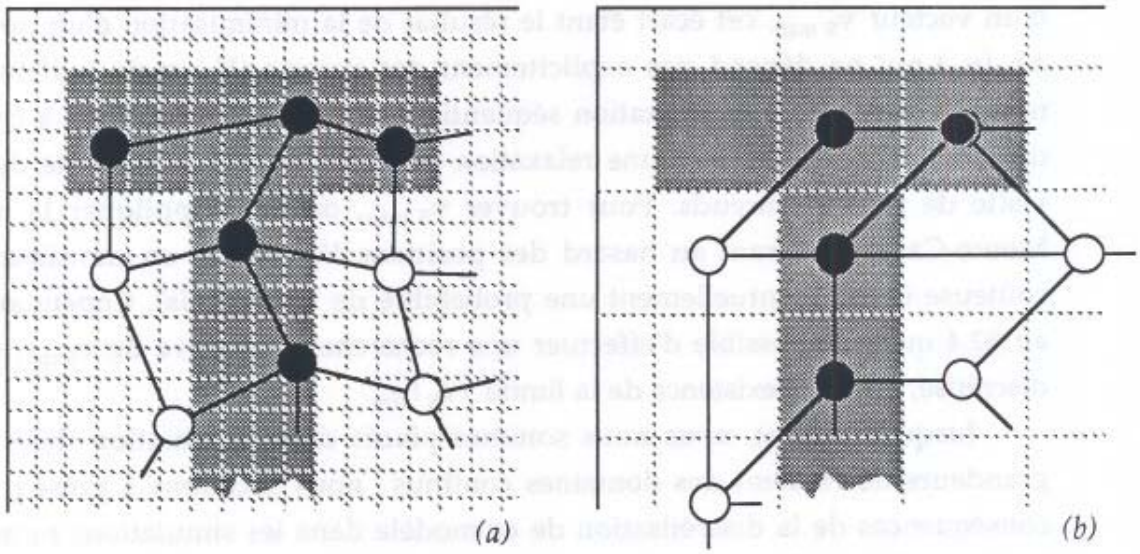


Figure 19 : Discrétisation du plan de l'image dans le cadre des simulations numériques. Cette illustration reprend l'exemple de la figure 10 (ici en gris, pour une meilleure lisibilité) *(a)* Le quadrillage est relativement fin (c=4), et réalise une bonne approximation des coordonnées réelles (comparer avec la figure 10b) : la longueur de référence, 1, étant celle d'une arête non-déformée, les coordonnées sont ici des multiples de 1/4. Donc cette discrétisation offre 16 pixels par unité de surface. *(b)* Dans ce deuxième cas, la grosseur du quadrillage est maximale (c=1, un seul pixel par unité de surface), et les nœuds ne peuvent se déplacer que sur les coordonnées entières qu'ils occupaient dans leur positions standard (deux nœuds noirs se trouvent ainsi superposés). Remarquons que le caractère "H" dont on montre ici un détail se prête parfaitement à la discrétisation en pixels, sans altération de ses frontières, ce qui ne serait pas le cas avec des traits courbes ou obliques, comme dans le caractère "A".

Dans ce cas, les coordonnées des nœuds sont désormais entières : on les notera $V_s = (k_s, l_s)$, et *R* désignera leur domaine (*R* est le *réseau* des entiers relatifs sur le plan). Par ailleurs, on rappelle que les nœuds sont repérés à l'aide une numérotation à deux indices (s_{ij}), celle-ci ayant permis de définir l'ensemble des arêtes (cf. §2.1.b) : ces indices sont internes au graphe, et ne doivent pas être confondus avec les coordonnées que peuvent recevoir les nœuds, qui sont des attributs géométriques externes. Ceci est particulièrement vrai sur *R*, et donc, pour $s = s_{ij}$, on aura en général : $V_s = (k_s, l_s) \neq (i, j)$. Cependant, comme on l'a signalé plus haut, indices et coordonnées coïncident exceptionnellement dans la définition des positions standard (cf. §2.1.c, figure 8) : les coordonnées $\mathbf{V}^0$ sont donc telles que pour

tout $s = s_{ij}$, $V_s^0 = (k_s^0, l_s^0) = (i, j)$, et elles réalisent ainsi un état non-déformé du graphe.

En conclusion, les images numérisées binaires Φ^* que nous utiliserons sont définies sur le réseau R de la façon suivante :

- $k = 1...16$ et $l = 1...16 \Rightarrow \Phi^*(k,l) = 0$ ou 1
- $(k \leq 0$ ou $k \geq 17)$ et $(l \leq 0$ ou $l \geq 17) \Rightarrow \Phi^*(k,l) = 0$

leur domaine de définition étant formellement étendu à l'ensemble du plan par un remplissage avec des pixels blancs (pour reprendre l'analogie mécanique du §2.3, les jetons du système déformable sont désormais noirs ou blancs et ne pourront se déplacer que sur des cases noires ou blanches de même largeur que la maille carrée de la grille orthonormée).

2.4.b Adaptation de l'énergie

Ces nouvelles dispositions nous conduisent à revoir l'expression de l'énergie H, et en particulier le rôle de Γ, qui doit désormais s'appliquer exclusivement à des étiquettes binaires. Jusqu'à présent, on avait $H(\mathbf{V}) = E(\mathbf{V}) + \lambda\, \Gamma(\mathbf{V})$, avec :

$$\Gamma(\mathbf{V}) = \sum_{s \in O} f\left(X_s^{\alpha*0}, \Phi^{\beta*}(V_s)\right)$$

Or, $\Phi^{\alpha*}$ et $\Phi^{\beta*}$ sont des images binaires, c'est à dire que les étiquettes des nœuds $X_s^{\alpha*0} = \Phi^{\alpha*}(V_s^0)$ et les pixels du réseau $\Phi^{\beta*}(V_s)$ valent seulement 0 ou 1. Donc f sera une simple fonction discriminante, c'est à dire : $f(1, 1) = f(0, 0) = 0$ et $f(1, 0) = f(0, 1) = 1$. Arrivés en ce point, il sera alors plus simple d'exclure Γ de l'énergie totale et d'adopter une contrainte *stricte* de coïncidence des images : ainsi, on posera $\lambda = +\infty$, ce qui obligera $\mathbf{V}$ à vérifier en permanence $\Gamma(\mathbf{V}) = 0$, c'est à dire : $\forall\, s \in O$, $\Phi^{\beta*}(V_s) = X_s^{\alpha*0}$. En effet, avec des petits graphes, Γ ne peut prendre qu'un nombre limité de valeurs entières (entre 0 et le nombre de nœuds), et ses variations ne sont guère significatives. De plus, l'optimisation des positions individuelles des nœuds sera accélérée par la restriction aux pixels de même couleur.

Adopter une contrainte stricte à la place d'une pénalité a pour effet de rendre le paysage d'énergie plus "abrupt", c'est à dire coupé par des pics infiniment hauts qui interdisent certaines régions de l'espace des **V**. Il n'est plus possible de transiger, en violant le respect des étiquettes et en payant cette violation par une pénalité. Cependant, les mouvements individuels des nœuds ne sont pas limités au voisinage immédiat : les nœuds ont la possiblité de sauter directement à la meilleure position locale $v_{s\,min}$ ayant la bonne couleur, et peuvent ainsi traverser des zones de couleur opposée (ce qui ne serait pas possible avec des déplacements de pixel en pixel). En revanche, certaines barrières d'énergie resteront infranchissables à moins d'envisager une température non-nulle ou des mouvements collectifs de nœuds (minimisation d'une l'énergie locale associé au déplacement solidaire d'un bloc de nœuds).

Avec une température nulle et des mouvements individuels, la relaxation sous une contrainte stricte de respect des étiquettes tombera inévitablement dans des minima locaux du paysage d'énergie sans pouvoir en sortir. Nous verrons au chapitre 3 que cette approximation ne nuit pas à la qualité de la reconnaissance.

Le passage de Γ d'une énergie à une contrainte stricte ne sera pas la seule adaptation au cas pratique. La deuxième modification consistera à rajouter un *nouveau terme* à l'énergie, conçu pour traiter un problème spécifique à l'application du graphe sur le réseau de pixels R. Comme on l'a fait remarquer plus haut, les pixels sont de même largeur que la maille élémentaire du graphe, ce qui veut dire que dans l'état non-déformé, l'encombrement du réseau est maximal, chaque pixel étant occupé par un nœud (entre les coordonnées (1,1) et (16,16)). Par conséquent, lorsque les nœuds commenceront à bouger, ils vont inévitablement se superposer sur les même pixels : comme il n'existe pas de positions intermédiaires entre deux nœuds initialement voisins, on assistera fréquemment à des accumulations de nœuds sur un même point (k,l), ce qui résultera en des contractions locales du graphe. Ce phénomène (d'autant moins important que c est grand) n'est pas souhaitable car il n'offre pas de bonnes conditions d'appariement : il est préférable que le graphe reste "déployé" sur l'ensemble de l'image, pour être confronté à toutes ses aspects et fournir un coût qui reflète le plus fidèlement possible les différences entre les deux images (l'une portée par le graphe, l'autre dessinée sur le plan).

Or, dans l'état actuel des formules, les superpositions sont peu pénalisées : en effet, si deux nœuds voisins s et t se trouvent sur un même point, il s'ensuit une petite pénalité élastique de 1, mais si ces nœuds ne sont pas voisins dans le graphe, cette superposition n'apporte aucune contribution à l'énergie. C'est pourquoi, étant donné le choix c=1, nous définirons la fonction-coût suivante :

$$S_R(\mathbf{V}) = \sum_{\substack{(k,l) \in R; \\ \rho(k,l) \geq 1}} (\rho(k,l) - 1)^2, \quad \text{avec} \quad \rho(k,l) = \text{card}\,\{s \in O;\, V_s = (k,l)\}$$

Ce terme est calculé directement sur le réseau R : il est fondé sur une "densité" locale de nœuds $\rho(k,l)$, c'est à dire le nombre de nœuds s ayant (k,l) pour coordonnées. La pénalité locale associée vaut alors $(\rho(k,l) - 1)^2$ si $\rho \geq 1$, et 0 sinon (les points vides $\rho = 0$ ne sont pas pénalisés pour ne pas constituer a contrario des "attracteurs" de nœuds) : donc la minimisation de cette pénalité aboutit à 0 ou 1 nœud sur le pixel (k,l). Ainsi, $S = 0$ correspond à un déploiement parfait du graphe sur le réseau.

La fonction S est alors ajoutée à l'énergie, affectée d'un coefficient positif κ, et on aura désormais :

$$H(\mathbf{V}) = E(\mathbf{V}) + \kappa S(\mathbf{V})$$

Dans l'analogie mécanique, on peut considérer que S induit des *forces de répulsion* entre jetons : ces forces les incitent à s'écarter les uns des autres seulement lorsqu'ils sont superposés sur une même case, mais elles n'agissent plus dès que ces jetons sont distants d'au moins une unité. Il serait possible d'envisager des forces de répulsion faisant intervenir des interactions à plus longue portée, cependant, le calcul serait alors d'ordre 2 (somme sur des couples de positions) au lieu de 1 comme c'est le cas ici (somme sur des positions simples).

Remarquons qu'il n'est pas possible de remplacer κS par une contrainte stricte (c'est à dire $\kappa = +\infty$, et $S(\mathbf{V}) = 0$), en même temps que le maintien de la contrainte $\Gamma = 0$, si le nombre de nœuds noirs est supérieur au nombre de pixels noirs. Par ailleurs, même dans le cas $\Gamma > 0$, l'exigence $S = 0$ bloquerait très rapidement la minimisation, à cause justement de la grande densité des nœuds sur le réseau (elle provoquerait une immobilisation dans un état de grande distorsion). Par conséquent, $S = 0$ n'est envisageable que si $\Gamma > 0$ et $c > 1$. C'est pourquoi, dans le cas présent ($\Gamma = 0$ et $c = 1$), on gardera une valeur finie pour κ.

En résumé, les deux adaptations que nous avons apportées à l'énergie sont indépendantes : la première concerne les étiquettes (X_S) tandis que la deuxième s'occupe des conflits entre positions (V_S). Dans d'autres situations, il serait possible, voire indispensable, de garder le terme Γ tout en conservant le terme S, ce qui donnerait une énergie complète : $H(\mathbf{V}) = E(\mathbf{V}) + \lambda\,\Gamma(\mathbf{V}) + \kappa\,S(\mathbf{V})$. Cependant, dans le présent problème, la formulation adoptée sera désormais : $H(\mathbf{V}) = E(\mathbf{V}) + \kappa\,S(\mathbf{V})$, et son domaine de définition sera déterminé par la contrainte permanente sur les images : $\Gamma^{\alpha\beta}(\mathbf{V}) = 0$. Nous décrivons au §2.4.c les mouvement discrets des nœuds qui aboutissent à la minimisation de H sous cette contrainte.

2.4.c Equilibre local

Etant donné un nœud s, et étant donné les positions fixes des voisins de s, le but est, comme au §2.3.c, de trouver la meilleure position locale V_S sur le réseau. Le processus de minimisation est donc le même que précédemment à trois différences près : (a) d'une part, les mouvements sont limités aux coordonnées entières $V_S = (k_S, l_S) \in R$; (b) d'autre part, il faut substituer le terme κS au terme $\lambda\,\Gamma$; (c) enfin, V_S doit toujours vérifier $g_S(\mathbf{V}) = 0$, c'est à dire $\Phi^{\beta*}(V_S) = X_S{}^{\alpha*0}$. On notera alors σ_S la pénalité locale de superposition :

$$\sigma_S(\mathbf{V}) = \rho(V_S)^2 - (\rho(V_S) - 1)^2$$
$$= 2\rho(V_S) - 1$$

et, dans la suite, tout se passera comme si σ_S "remplaçait" la pénalité locale d'images, g_S. La quantité $\sigma_S = 2\rho(V_S) - 1$ mesure la pénalité encourue par le nœud s si celui-ci se

déplace en un point $V_S = (k_S, l_S)$ sur lequel se trouvaient déjà $\rho(V_S)$ nœuds (s non-compris), et provoque alors un empilement de hauteur $\rho(V_S) + 1$: donc, avant l'arrivée de s, l'encombrement était pénalisé par $(\rho(V_S) - 1)^2$, et après son arrivée, cette pénalité augmente à $\rho(V_S)^2$. Cependant, avec la formule $2\rho(V_S) - 1$, la transition d'un pixel vide à un pixel occupé par un nœud est récompensée par -1 : donc, pour ne pas que les pixels vides attirent les nœuds, σ_S doit être exceptionnellement nulle pour $\rho = 0$, ce que l'on écrira : $\sigma_S(V) = (2\rho(V_S) - 1)^+$ (le "+" symbolise la troncation à 0 des valeurs négatives).

Comme auparavant, nous compterons V_S à partir de la position centrale barycentrique V_S^b déterminée par ses n_S voisins, en posant $v_S = V_S - V_S^b$. La variation de pénalité par rapport à V_S^b vaut alors :

$$\Delta\sigma_S(v_S) = (2\rho(V_S^b+v_S) - 1)^+ - (2\rho(V_S^b) - 1)^+$$

Remarquons que les densités ρ doivent être comptées *ayant exclu le nœud s* de son ancienne position : lorsque la position de s est remise en question à l'itération τ, ce nœud doit être retiré provisoirement du plan, en diminuant $\rho(V_S(\tau-1))$ de 1 le temps de la recherche de $v_{S\,min}$.

Finalement, la variation totale de pénalité $\Delta h_S(v_S) = \Delta e_S(v_S) + \kappa\,\Delta\sigma_S(v_S)$ s'écrit (comparer avec celle du §2.3.c) :

$$\Delta h_S(v_S) = n_S\,|\,v_S\,|^2 + \kappa(2\rho(V_S^b+v_S) - 1)^+ - \kappa(2\rho(V_S^b) - 1)^+$$

$$= n_S\,|\,v_S\,|^2 + \kappa(2\rho(V_S^b+v_S) - 1)^+ - \text{cste}$$

Remarquons que S n'est pas égal à la somme des σ_S, de même que E n'était pas égal à la somme des e_S (cf. §2.3.b) : la somme des σ_S sur s reviendrait à compter la pénalité $(2\rho(k,l) - 1)^+$ autant de fois qu'il y a de nœuds qui occupent (k,l), c'est à dire $\rho(k,l)$ fois, ce qui n'est bien sûr pas la même chose que $(\rho(k,l) - 1)^{+2}$. Les mêmes remarques que pour (e_S) s'appliquent donc à (σ_S) : il ne s'agit pas de quantités qui forment une partition de S, mais de pénalités locales qui sont en recouvrement. En revanche les *variations* de S et σ_S selon V_S sont par définition égales : c'est pourquoi il est légitime d'utiliser σ_S dans la recherche de la meilleure position locale.

La meilleure position pour s est donnée par $V_S = V_S^b + v_S$, avec :

$$v_{S\,min} = \text{argmin}_{\,g_S(V)=0}\,\Delta h_S(v_S)$$

ce qui signifie que $v_{S\,min}$ doit réaliser le minimum de Δh_S sous la contrainte $g_S(V) = 0$, c'est à dire : $\Phi\beta^*(V_S^b+v_S) = X_S^{\alpha*0}$. La discussion du §2.3.d a montré qu'il existait un éloignement limite, $|\,v_S\,|_{lim}$, au-delà duquel il n'était plus utile de rechercher $v_{S\,min}$, car l'augmentation du potentiel élastique $\Delta e_S(v_S) = n_S\,|\,v_S\,|^2$ devenait prédominante devant les fluctuations bornées du terme d'image $\Delta g_S(v_S)$. A

présent, on a remplacé $\Delta g_S(v_S)$ par $\Delta\sigma_S(v_S)$, mais le raisonnement est le même : les pénalités de superpositions σ_S sont également majorées, ne serait-ce que par la quantité $(n-1)^2$, où n est le nombre total de nœuds dans le graphe (cette majoration théorique n'étant jamais atteinte en réalité), tandis que $n_S \mid v_S \mid^2$ peut toujours croître aussi loin qu'on le désire, le réseau étant supposé infini. Par conséquent, on est sûr de trouver $v_{S\,min}$ avant un certain $\mid v_S \mid_{lim}$.

L'existence d'un cercle limite $\mid v_S \mid_{lim}$ est alors particulièrement intéressante sur le réseau des pixels, puisqu'il est possible d'imaginer une recherche exhaustive par $\mid v_S \mid$ croissants. Voici donc la procédure adoptée : on calcule d'abord le barycentre V_S^b fourni par les voisins de s en le rapportant au point de coordonnées entières le plus proche, puis on procède à une *visite centrifuge* des points autour de V_S^b, en commençant par V_S^b lui-même. Ainsi, v_S prendra successivement les valeurs suivantes : (0, 0), (0, 1), (–1, 0), (0, –1), (1, 0), (1, 1), (1, –1), ... , (2, 0), (0, 2), etc... C'est donc sur une spirale que le nœud s est déplacé, à partir du minimum élastique qui constitue le centre de celle-ci (figure 20). En chacun des points de cette spirale, on calcule la pénalité locale encourue, $\Delta h_S(v_S)$ si et seulement si la couleur de ce point est la même que celui du nœud s (en vertu de la contrainte stricte sur g_S), et on n'en retient que la plus petite valeur rencontrée jusqu'alors, $\Delta h_{S\,min}$, ainsi que la position où elle a été calculée, $v_{S\,min}$. On finit alors par rencontrer un point en lequel la pénalité élastique $\Delta e_S(v_S)$ est à elle seule plus grande que la meilleure valeur $\Delta h_{S\,min}$: il est donc inutile de poursuivre la visite, puisque les points de la spirale sont classés par distances croissantes, et la solution est $v_S = v_{S\,min}$.

Ce processus peut être formalisé de la façon suivante : on note $v^1, v^2, \ldots, v^k, \ldots$ les points de la spirale (figure 20b), c'est à dire les positions relative de s par rapport au barycentre, classées par normes croissantes, $0 = \mid v^1 \mid \leq \mid v^2 \mid \leq \ldots \leq \mid v^k \mid \leq \ldots$, et on notera $\wp^k$ l'ensemble des k premiers points de cette spirale, *dont sont exclus* ceux qui n'ont pas le même pixel que le nœud s, c'est à dire :

$$\wp^k = \{v^1, v^2, \ldots, v^k\} \cap \{v;\ \Phi^{\beta*}(V_S^b + v) = X_S^{\alpha*0}\}.$$

Ceci permet de définir le plus petit Δh_S rencontré jusqu'au point k par : $\Delta h_S{}^k{}_{min} = \min_{v \in \wp^k} \Delta h_S(v)$, ainsi que le point qui lui correspond, $v^k{}_{min}$, qui est tel que $\Delta h_S(v^k{}_{min}) = \Delta h_S{}^k{}_{min}$. Le constat qui permet d'arrêter les recherches est alors celui-ci : $\exists\, k;\ \forall\, k' \geq k,\ \Delta e_S(v^{k'}) \geq \Delta h_S{}^k{}_{min}$, où $\Delta e_S(v^{k'}) = n_S \mid v^{k'} \mid^2$ est le supplément de pénalité locale élastique encouru par l'écart $v^{k'}$. Par conséquent, on a *a fortiori* $\Delta h_S(v^{k'}) \geq \Delta h_S{}^k{}_{min}$, et c'est donc l'indice k qui fournit la solution : $v_{S\,min} = v^k{}_{min}$.

Remarque : le vrai barycentre V_S^b est la moyenne de 4 points (ou moins) du réseau *R*, donc il n'appartient pas forcément à *R*, car ses coordonnées sont rationnelles (il tombe entre les points). C'est pourquoi, avant de commencer la recherche en spirale autour de V_S^b, nous l'avons rapporté au point de *R* le plus proche, que noterons ici $V_S^b[R]$ pour le distinguer du V_S^b réel (figure 20a). Cependant, la pénalité élastique n'est alors pas exacte, puisqu'elle est comptée à partir de $V_S^b[R]$ et non V_S^b : pour être tout à fait rigoureux, il faudrait donc procéder comme précédemment, en tournant autour de $V_S^b[R]$, *mais* en corrigeant $\Delta e_S(v^{k'}) = n_S \mid v^{k'} \mid^2$ en $\Delta e_S(v^{k'}) = n_S \mid v^{k'} + (V^b{}_S - V^b{}_S[R]) \mid^2$. Dans ce cas, il resterait un petit problème, puisque $\Delta e_S(v^{k'})$ n'augmenterait plus de façon monotone avec la progression de l'indice k', et

fluctuerait légèrement. Il suffirait alors de poursuivre la visite un peu plus loin que le point d'arrêt k pour être tout à fait sûr de ne plus rencontrer de Δe_s légèrement inférieurs à la valeur frontière $\Delta e_s(v^k)$.

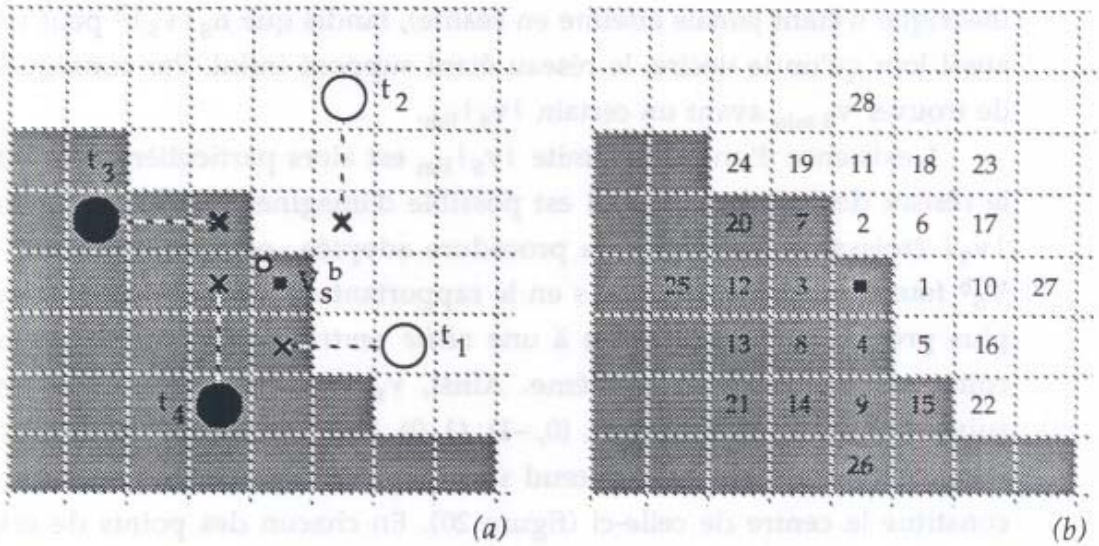


Figure 20 : Recherche du minimum local par visite centrifuge des pixels à partir du barycentre. *(a)* Les voisins du nœud s désignent leurs préférences (positions V_{ti} marquées d'une croix; cf. figure 17), ce qui place le barycentre à l'endroit du petit cercle : celui-ci doit alors être rapporté au pixel du réseau le plus proche (petit carré) (dans cette illustration, on a pris c=2, le principe étant le même pour n'importe quelle résolution du réseau). *(b)* A partir du barycentre, on parcourt les écarts v_s par normes $|v_s|$ croissantes, comme l'indique la numérotation (celle-ci n'est bien sûr pas unique, car pour un rayon r donné, il existe au moins quatre v_s tels que $|v_s| = r$). En raison de la contrainte stricte $g_s = 0$, on ne s'intéresse qu'aux pixels de même couleur que le nœud : donc si l'étiquette du nœud s est blanche, seuls les points 1, 2, 5, 6, 10, 11, ... seront pris en considération. Parmi ces points, on cherche alors celui qui minimise $\Delta h_s(v_s) = n_s |v_s|^2 + \kappa \Delta\sigma_s(v_s)$. Si le voisinage immédiat de V_s^b n'est pas encombré par d'autres nœuds (comme c'est vraisemblablement le cas sur cet exemple), il est clair que ce sont les positions les plus proches qui l'emporteront (ici, 1 ou 2). Si en revanche d'autres nœuds occupent ces positions, il faudra chercher plus loin des pixels libres, dans la limite où cet éloignement ne provoque pas une augmentation trop grande de la pénalité élastique.

2.4.d Equilibre global

Etant donné les images $\Phi^{\alpha*}$ et $\Phi^{\beta*}$, le but est d'approcher le minimum absolu :

$$H^{\alpha\beta}{}_{min} = \min_{\mathbf{V};\, \Gamma^{\alpha\beta}(\mathbf{V})=0} H(\mathbf{V})$$

qui est idéalement indépendant du chemin suivi dans le sous-espace des **V** contraints par $\Gamma^{\alpha\beta}(\mathbf{V}) = 0$. Dans toute la suite on notera ce sous-espace $\vartheta^{\alpha\beta}$, c'est à dire : $H^{\alpha\beta}{}_{min} = \min_{\mathbf{V}\in\vartheta^{\alpha\beta}} H(\mathbf{V})$. En pratique, on ne cherchera pas à atteindre cette valeur, et on se contentera de l'approximation offerte par la stabilisation de **V** dans un mininum local. Comme on l'a mentionné au §2.3.d, l'optimisation des positions de nœuds sera effectuée séquentiellement, c'est à dire : $V_s(\tau) = V_s^b(\tau) + v_{s\,min}(\tau)$, où $V_s^b(\tau)$ est calculé sur les anciennes positions des voisins, $V_t(\tau-1)$, et τ est incrémenté après chaque visite de nœud. Par conséquent, le résultat global de l'optimisation dépendra en réalité du chemin suivi dans $\vartheta^{\alpha\beta}$, c'est à dire d'un certain nombre de conditions pratiques et de paramètres, qu'il reste à décrire : il s'agit d'une part des

conditions initiales, et d'autre part du protocole de visite itérative des nœuds.

Commençons par éclaircir le dernier aspect : nous avons choisi de parcourir les nœuds dans un *ordre aléatoire*, fixé à l'avance pour s'assurer que chaque nœud s est optimisé une fois et une seule. Donc on attribue un numéro d'ordre unique aux éléments de O, s_{ij}, à l'aide d'une bijection de [1,m] × [1,p] dans [1,mp] que l'on notera ω : ainsi, le nœud visité en premier sera celui pour lequel $\omega(s_{ij}) = 1$, etc. On appellera *itération* une visite complète de tous les nœuds du graphe à travers la permutation ω. La relaxation comprendra un certain nombre d'itérations N, identique pour tous les appariements. Si N est suffisamment grand, le processus finira par tomber dans un minimum local, caractérisé par une immobilisation des nœuds (ou éventuellement dans un cycle court, dû à une frustration autour de quelques positions). Mais on peut également vouloir écourter le temps de calcul en prenant un N plus petit, auquel cas on ne cherchera même pas à atteindre le minimum local le plus proche, et on se contentera seulement de quelques pas dans sa direction. Bien sûr, l'énergie d'un minimum local ou, qui plus est, à côté d'un minimum local, sera relativement éloignée du minimum absolu $H^{\alpha\beta}{}_{min}$, donc en constituera une approximation plus ou moins grossière. Cependant, nous verrons au chapitre 3 que, dans la mesure où tous les appariements seront traités de la même manière, le critère de classification utilisant cette valeur restera cohérent et produira de bons taux de reconnaissance, sans qu'il soit besoin de pousser la précision de la minimisation.

Il reste à préciser la manière dont ce processus récurrent peut démarrer, c'est à dire le choix de $\mathbf{V}(0)$. Au départ, les nœuds noirs du graphe doivent être placés sur des points noirs du réseau, et les nœuds blancs sur des points blancs, puisque la recherche est restreinte à $\vartheta^{\alpha\beta}$. Mais il y a là une difficulté, car la façon de procéder à ce premier placement n'est pas unique, et le point de départ de $\vartheta^{\alpha\beta}$ conditionnera pour une bonne part la réussite ultérieure de l'optimisation. En tout premier lieu, on placera le graphe dans un état non-déformé sur l'image Φ^{β} avec une éventuelle translation d'un vecteur (a, b), sans tenir aucun compte des étiquettes : autrement dit, les nœuds recevront des positions virtuelles provisoires $V_s{}^{0'} = V_s{}^0 + (a, b)$, c'est à dire pour tout $s = s_{ij}$: $V_s{}^{0'} = (i+a, j+b)$ (voir la figure 23a, chapitre 3). Ainsi, la grille orthonormée est posée telle quelle sur le plan, et on a : $\Gamma^{\alpha\beta}(\mathbf{V}^{0'}) \neq 0$. On se trouve donc *en dehors* de $\vartheta^{\alpha\beta}$, et la première étape aura pour but de rapporter $\mathbf{V}^{0'}$ à un élément $\mathbf{V}(0)$ de $\vartheta^{\alpha\beta}$, l'idéal étant de trouver le point le plus proche au sens de H. On appellera *initialisation* ou *itération-0* la procédure préliminaire qui est chargée de choisir ces premières positions de nœuds. L'itération-0 est donc une projection de $\mathbf{V}^0$ sur l'ensemble $\vartheta^{\alpha\beta}$, les N itérations qui suivent ayant pour rôle de corriger cette première déformation en conduisant ensuite l'état $\mathbf{V}$ à l'intérieur de $\vartheta^{\alpha\beta}$ vers une position d'énergie plus faible. Voici deux propositions d'itération-0, dont nous discutons les avantages et les inconvénients.

L'idée la plus simple est de placer chaque nœud sur le point du réseau qui est *le plus proche* de la position provisoire de ce nœud, parmi les points qui ont même

étiquette que lui. En d'autres termes, pour $s = s_{ij}$, qui se trouve pour l'instant sur la position virtuelle $V_s{}^{0'} = (i+a, j+b)$, on posera : $V_s(0) = (i', j')$, où i' et j' vérifieront d'une part $\Phi\beta^*(i', j') = X_s{}^{\alpha*0} = \Phi\alpha^*(i, j)$, et d'autre part :

$$\forall\ (k,l) \in R,\ |i+a-i'|^2 + |j+b-j'|^2 \leq |i+a-k|^2 + |j+b-l|^2$$

Ce procédé a l'avantage de la simplicité, mais son important défaut est de ne tenir aucun compte des relations entre les nœuds, puisqu'il les place indépendamment les uns des autres. Donc il n'accorde pas d'importance à ce qui fait tout l'intérêt de l'appariement élastique, c'est à dire les *contraintes de structure*. Dans cette initialisation, les nœuds sont traités comme si les arêtes n'existaient pas, et, au moment où elles réapparaîtront pour participer aux N itérations suivantes, il sera alors difficile de réparer les torsions et les enchevêtrements du graphe causés par cette initialisation rapide.

Par conséquent, il faudrait améliorer cet algorithme en redonnant aux relations de voisinage le rôle primordial qu'elles doivent avoir. La deuxième idée est alors la suivante : la recherche de la meilleure première position $V_s(0)$ du nœud $s = s_{ij}$ se fera exactement comme les modifications ultérieures de cette position au cours des itérations suivantes. En clair : pour initialiser le nœud s_{ij} sur un point du réseau de même couleur que lui, *on suivra la procédure élémentaire récurrente décrite au* §2.4.c, c'est à dire que s_{ij} devra se référer à ses voisins dans le graphe $\{s_{i+1,j}, s_{i,j+1}, s_{i-1,j}, s_{i,j-1}\}$ pour choisir une position non pas proche de sa position provisoire, mais proche du barycentre imposé par *ceux de ses voisins qui sont déjà placés*. Evidemment, une objection majeure à cette entreprise est que les nœuds ne sont pas encore placés, puisque nous nous occupons justement de l'initialisation. La réponse est alors celle-ci : il n'est pas nécessaire de disposer de tous les voisins d'un nœud pour appliquer la recherche locale vue au §2.4.c, et, à la limite, un seul voisin peut suffire pour servir de repère à s_{ij}, et désigner le centre de la spirale. Par exemple, si s est le nœud devant recevoir une position, et si t est son unique voisin déjà positionné, alors le barycentre calculé sur ce seul point vaudra simplement : $V_s{}^b = V_t + V_s{}^0 - V_t{}^0$. Cependant, il sera nécessaire d'amorcer le processus en plaçant d'avance quelques nœuds choisis au hasard (par exemple les premiers du classement ω) indépendamment les uns des autres, en suivant la première idée (projection directe sur le point de même étiquette le plus proche). Ainsi, l'itération-0 ressemblera presque à une itération normale, c'est à dire qu'elle en appliquera les mêmes principes, avec cependant moins de matériel : la grille G sera *progressivement* appliquée sur le réseau, en commençant par quelques nœuds, puis en ralliant les autres autour de ces premiers repères. Les placements se font d'abord de façon décorrélée, puis par propagation de proche en proche. On peut voir un exemple concret d'itération-0 sur la figure 23 du chapitre suivant.

3 APPLICATION

Dans ce chapitre, nous présentons une application de la procédure d'appariement élastique décrite au chapitre 2, sur une base de données D contenant 1200 images de caractères manuscrits. L'estimation de la ressemblance entre caractères, qui s'appuie sur une déformation de l'image et qui est mesurée par le minimum de la fonction-coût H, fournira une *distance* à l'intérieur de D : ainsi, on pourra considérer que deux caractères sont d'autant plus "proches" (c'est à dire ressemblants), que leur coût minimal d'appariement élastique est faible. Munis de cette distance, nous pourrons alors définir une fonction de *classification* à l'aide du critère des plus proches voisins. Pour cette classification, on aura également besoin de se référer à des exemples connus, c'est à dire déjà classés, appelés *prototypes* (ou *exemples d'apprentissage* dans un autre contexte), et, parallèlement, disposer d'exemples réputés inconnus pour tester la méthode. Le principe sera alors le suivant : on classera les exemples de test à l'aide de l'estimateur (prototypes + distance + critère) et on comparera la décision de cette machine avec la réponse que nous connaissons par ailleurs. Il s'agit donc d'un exercice destiné à juger de la qualité de la distance à travers le classifieur, celui-ci devant être le plus fidèle possible à une réalité connue. Cependant, comme la base de données que nous possédons est par définition limitée, nous ne pourrons pas la consacrer entièrement aux prototypes, et nous devrons en réserver une partie pour les exemples tests. Les prototypes seront alors tirés aléatoirement (étant répartis de façon égale sur les classes). Finalement, les performances du classifieur se résumeront à un *taux de reconnaissance*, qui sera proportionnel au nombre de caractères tests bien classés : on calculera alors la valeur *moyenne* de ce taux sur différents ensembles de prototypes. Les taux de reconnaissance de notre méthode, qui est fondée sur une distance élastique de graphes, seront comparés avec ceux des méthodes fondées sur la distance de Hamming appliquée à des représentations non structurées (méthodes utilisant directement cette distance avec un critère de décision, ou réseaux de neurones à couches).

3.1 Problème

3.1.a Présentation des données

L'ensemble des caractères sur lequel nous avons travaillé contient 1200 exemples de chiffres manuscrits de 0 à 9 : ces chiffres ont été écrits par 12 personnes, à raison de 10 caractères par classe et par personne (production de AT&T, Laboratoires Bell). Les images sont déjà *normalisées* et *binarisées* : ce sont des

tableaux de pixels noirs et blancs, de taille 16x16 (figure 21). Les scripteurs ont pris référence sur des modèles de caractères préconçus, ce qui fait que cette base de données est relativement simple : en général il existe peu de variation entre échantillons d'une même classe (par exemple tous les "1" sont de simples bâtons, sans pied ni chapeau). Les simulations seront également effectuées sur des caractères *squelettisés* (figure 21) : ce prétraitement supplémentaire aura pour but d'éliminer les différences d'épaisseur entre caractères d'une même classe pour éviter des contractions ou extensions du graphe G inutilement coûteuses en énergie.

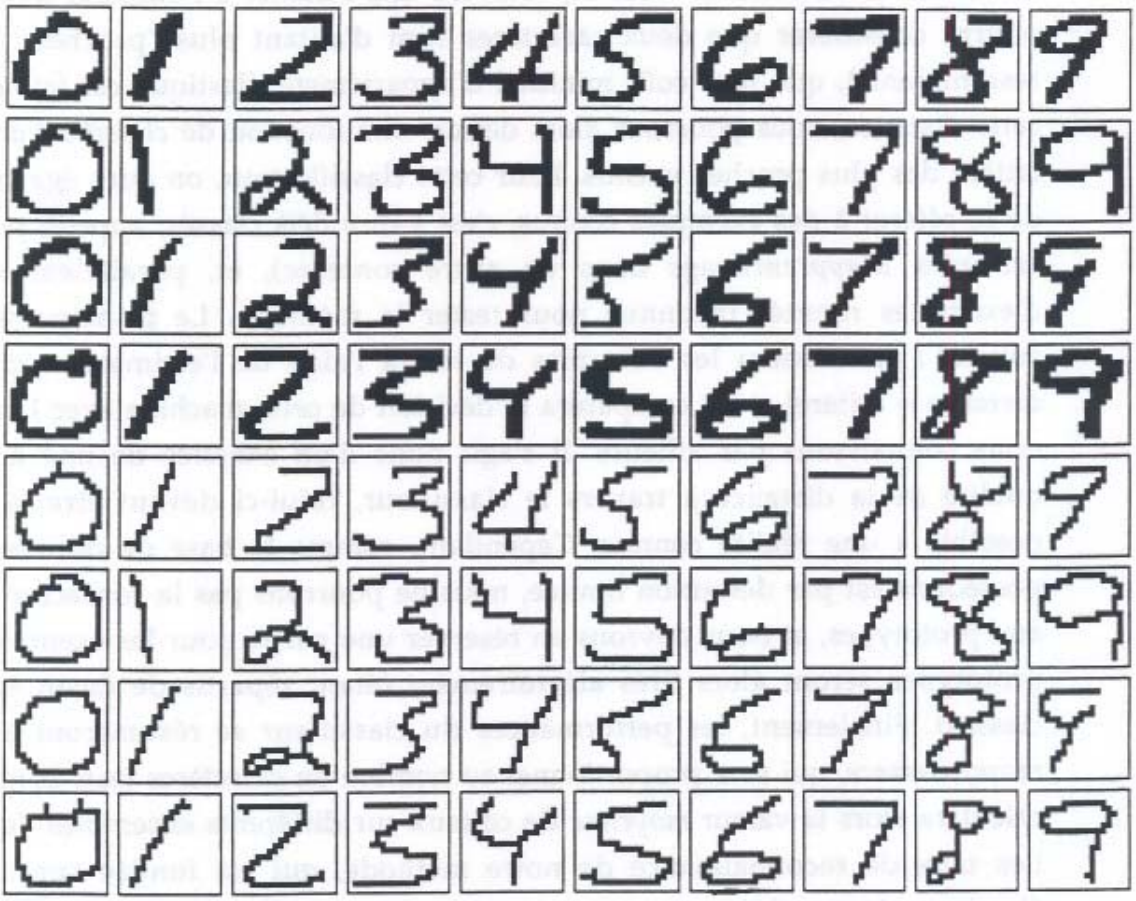


Figure 21 : Echantillon de la base des caractères. Les quatre rangées supérieures donnent un aperçu des images numérisées et renormalisées que contient la base originale. Les quatre rangées inférieures montrent les mêmes caractères après squelettisation. L'appariement élastique pourra utiliser l'une ou l'autre de ces versions.

Cette base de données, squelettisée ou non-squelettisée, sera notée :

$$D = \{\Phi^{0,1}, \ldots, \Phi^{0,120}, \ldots, \ldots, \Phi^{9,1}, \ldots, \Phi^{9,120}\}$$

(le symbole * qui signalait l'opération de seuillage sur Φ sera omis dans toute la suite). Ainsi, pour tout $c = 0...9$, et pour tout $\mu = 1...120$, $\Phi^{c,\mu}$ désignera l'image du μ-ème caractère de la classe c (on pourra également utiliser un indice global $\alpha = 1...1200$, c'est à dire : $\Phi^{c,\mu} = \Phi^{\alpha}$ avec $\alpha = 120c+\mu$). On notera D^c le sous-ensemble de D correspondant à la classe c, soit : $D = D^0 \cup D^1 \cup \ldots \cup D^9$, avec $D^c = \{\Phi^{c,1}, \ldots, \Phi^{c,120}\}$.

3.1.b Particularités des graphes et de l'énergie

Avec ces données, on peut maintenant appliquer la procédure d'appariement élastique, telle qu'elle a été décrite au §2.4 dans sa version discrétisée. On rappelle que l'énergie vaut : $H(\mathbf{V}) = E(\mathbf{V}) + \kappa S(\mathbf{V})$, et que sa minimisation doit avoir lieu sous la contrainte $\Gamma^{\alpha\beta}(\mathbf{V}) = 0$. Ainsi, H ne dépend pas explicitement des images Φ^α et Φ^β, mais seulement implicitement. Remarquons que $\Gamma^{\alpha\beta}$ n'est pas symétrique en α et β, car Φ^α est prise comme référence pour les caractéristiques fixes $\mathbf{X}^{\alpha 0}$, tandis que Φ^β sert aux caractéristiques variables $\mathbf{X}^\beta = \Phi^\beta(\mathbf{V})$: donc c'est le graphe étiqueté $(G, \mathbf{X}^{\alpha 0})$ qui se déforme sur le plan de l'image Φ^β, et non l'inverse.

Comme toujours, le graphe G est une grille plane à maille carrée : ici, il contiendra au moins 16x16 nœuds, c'est à dire en général (16+2m)x(16+2m) nœuds, où m représente une bordure supplémentaire par rapport à la taille normale des images. Ainsi, dans son état standard non-déformé il recouvrira entièrement le tableau de pixels 16x16 représentant l'image Φ^α (figure 22a). Les nœuds de O seront indexés de façon à ce que leurs coordonnées standard soient exactement égales à leurs indices : ainsi, avec $O = \{s_{1-m,1-m}, \cdots, s_{16+m,16+m}\}$, on aura $V_{s_{ij}}^0 = (i,j)$ et $X_{s_{ij}}^{\alpha 0} = \Phi^\alpha(i,j)$. Donc l'étiquette fixe du nœud s_{ij} sera simplement déterminée par la couleur du pixel (i,j). Cependant, dans toute la suite, pour donner plus de souplesse aux déformations du graphe $(G, \mathbf{X}^{\alpha 0})$, G sera en réalité restreint à un *sous-graphe* G^α, dépendant de Φ^α, dont la forme reproduira celle du caractère α (figure 22b).

Pour cela, l'ensemble des nœuds O sera restreint à un sous-ensemble O^α qui contiendra tous les nœuds portant une étiquette noire ainsi que tous les nœuds portant une étiquette blanche situés dans un voisinage immédiat autour de ces nœuds noirs, c'est à dire à une distance inférieure à m :

$$O^\alpha = \{s_{ij} \in O;\ \exists\, s_{kl} \in O;\ \Phi^\alpha(k,l) = 1 \text{ et } |i-k|^2 + |j-l|^2 \le m^2\}$$

On notera n_α le nombre de nœuds contenus dans O^α. L'ensemble des arêtes A sera alors restreint au sous-ensemble A^α déduit de O^α: $A^\alpha = \{\langle s,t\rangle \in A;\ s \in O^\alpha \text{ et } t \in O^\alpha\}$. On voit que le sous-graphe $G^\alpha = (O^\alpha, A^\alpha)$ ainsi obtenu épouse les contours de la forme noire du caractère Φ^α, en conservant une certaine couche de nœuds blancs (remarquons que si la forme noire n'était pas connexe, cette couche devrait être suffisamment épaisse pour que le graphe G^α reste connexe).

La motivation de cette restriction est double : d'une part, on allège les calculs en diminuant le nombre de nœuds, d'autre part, on supprime les morceaux de zones blanches inutiles autour du caractère. Ces zones sont un artefact du cadre rectangulaire de l'image, et l'encombrement qu'elles provoquent nuit à la souplesse des déformations, en faisant obstacle aux mouvements des nœuds. En revanche, il peut être utile de conserver une couche de nœuds blancs autour des nœuds noirs, c'est à dire d'avoir $m \ge 1$, pour faire apparaître la frontière de l'image au niveau des pixels.

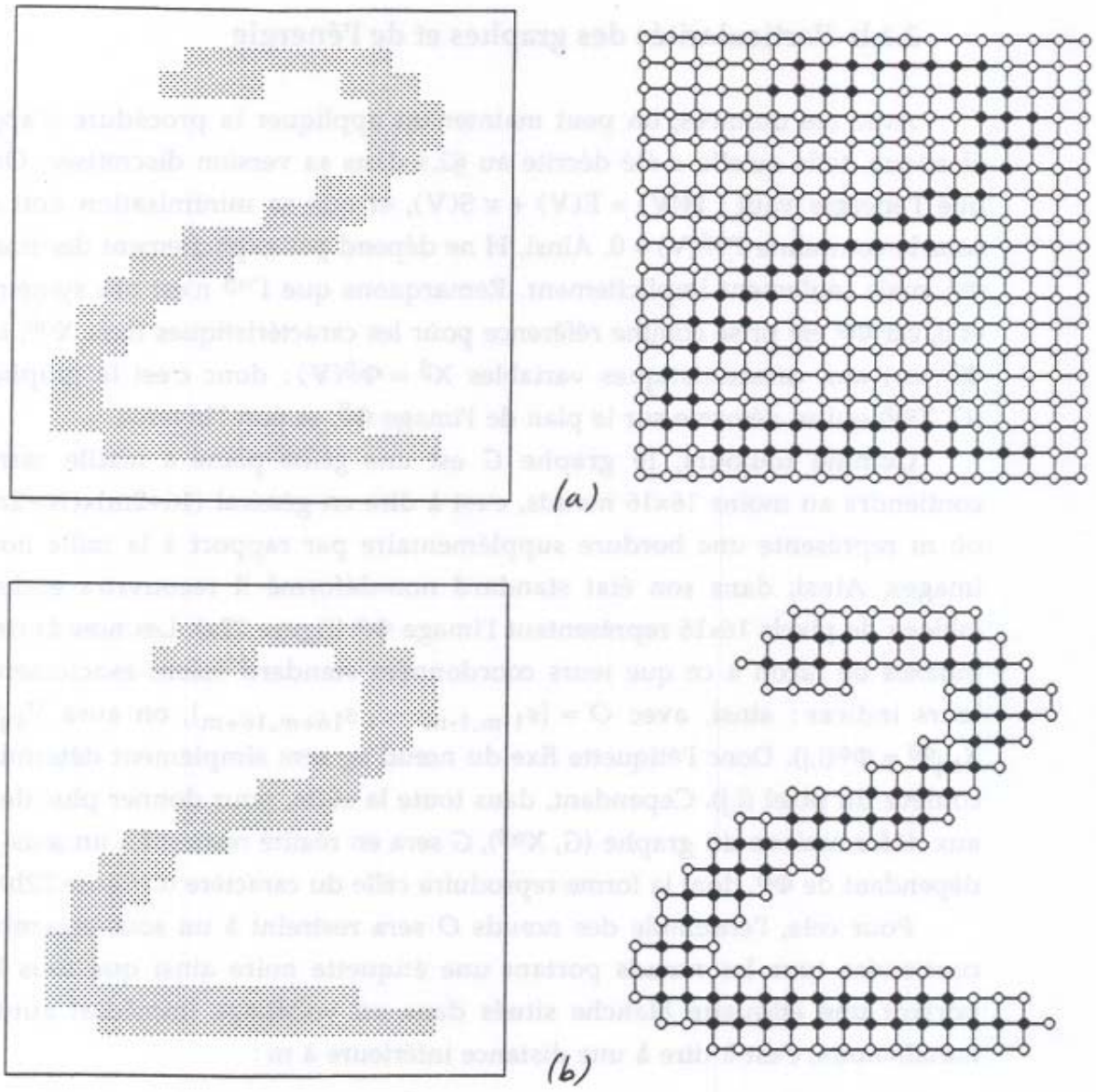


Figure 22 : Représentation relationnelle d'un caractère. *(a)* A gauche, l'image discrétisée est un tableau 16x16 de pixels noirs et blancs : chaque pixel est représenté par un petit pavé gris, et ces pavés sont soudés les uns aux autres (le cadre de l'image est plus grand que le tableau). A droite, la représentation standard en graphe étiqueté construite à partir de cette image utilise une grille plane à maille carrée à raison de 1 nœud par pixel (densité maximale du graphe sur le tableau), et avec une bordure supplémentaire de nœuds blancs de largeur m=1. Donc, ici, le graphe contient 18x18 nœuds. *(b)* Même chose, mais avec un *sous-graphe* dépendant de l'image : de la grille 18x18 sont retirés tous les nœuds blancs distants des nœuds noirs de plus de m=1, et toutes les arêtes attachées à ces nœuds. Les simulations seront effectuées sur ce graphe restreint.

En résumé, les valeurs d'énergie sont réglées par deux paramètres : la couche de nœuds blancs de largeur m, et la force relative κ du coût de superposition S par rapport au coût élastique E.

3.1.c Définition de la distance élastique

Etant donné un couple de caractères (Φ^α, Φ^β), l'intérêt de minimiser la fonction d'énergie H(V) sous la contrainte de coïncidence des images $\Phi^\beta(V) = \Phi^\alpha(V^0)$ est de

fournir une mesure du degré de dissemblance de ces caractères, qui sera considérée comme une *distance* entre Φ^α et Φ^β :

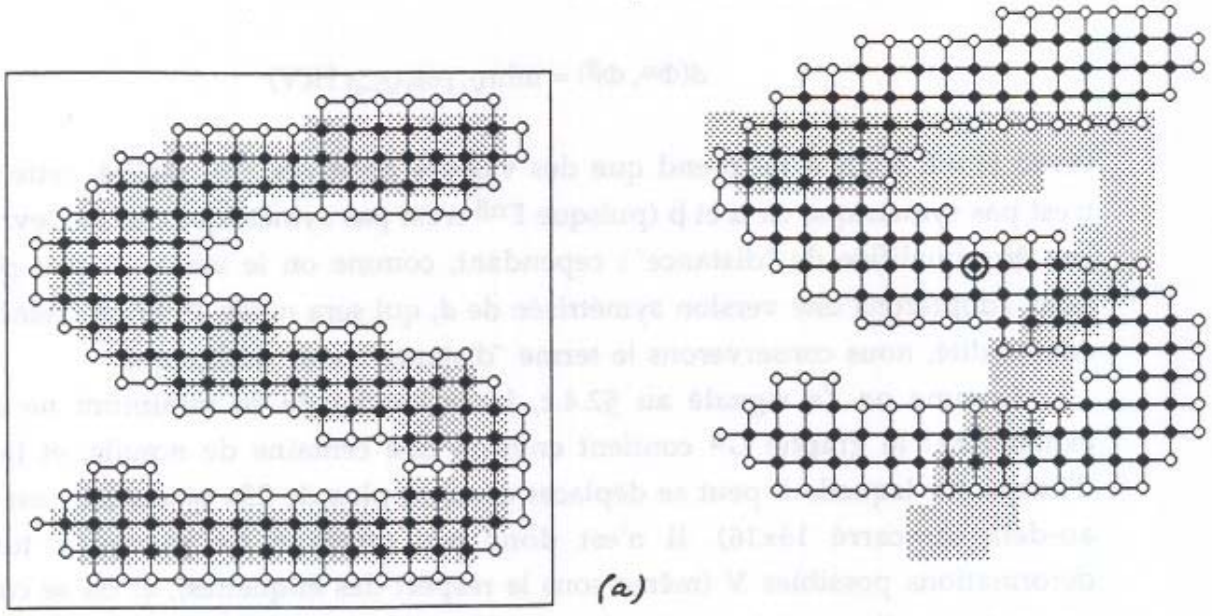
$$d(\Phi^\alpha, \Phi^\beta) = \min_{\mathbf{V};\, \Gamma^{\alpha\beta}(\mathbf{V})=0} H(\mathbf{V})$$

(avec $H = E + \kappa S$, d ne prend que des valeurs positives). En réalité, cette quantité n'est pas symétrique en α et β (puisque $\Gamma^{\alpha\beta}$ n'est pas symétrique), et ne devrait donc pas être qualifiée de "distance" : cependant, comme on le verra un peu plus loin, nous adopterons une version symétrisée de d, qui sera notée d^*, et en attendant, par commodité, nous conserverons le terme "distance" pour désigner d.

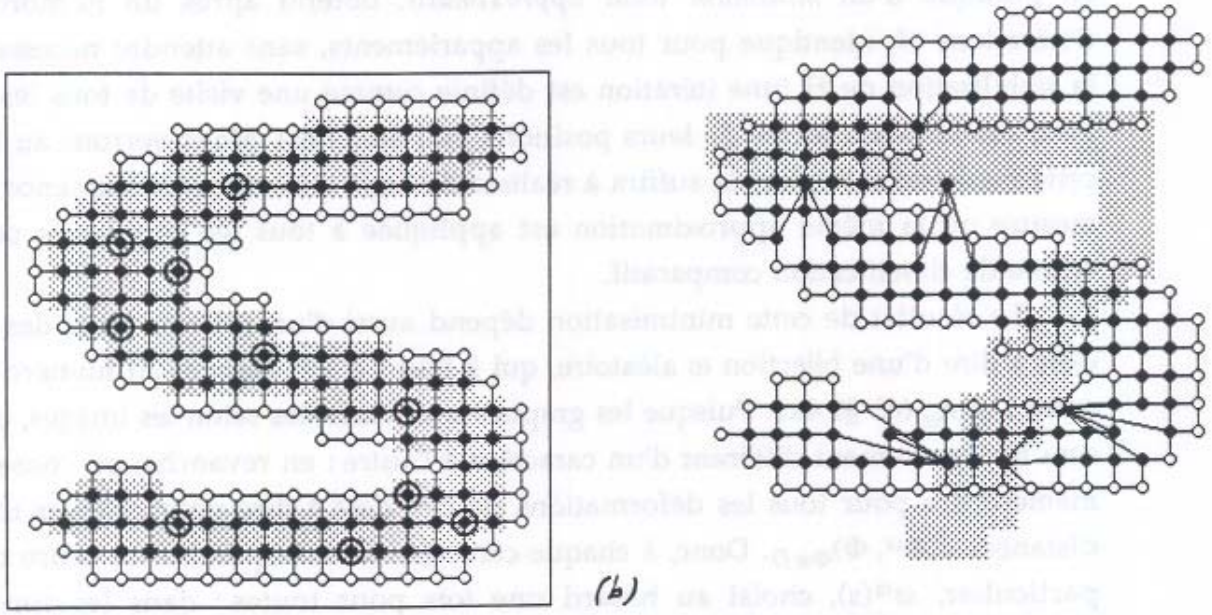
Comme on l'a signalé au §2.4.c, la recherche de ce minimum ne sera pas exhaustive : le graphe G^α contient environ une centaine de nœuds, et la surface d'image sur laquelle il peut se déplacer contient plus de 256 pixels (elle est étendue au-delà du carré 16x16). Il n'est donc pas question de parcourir toutes les déformations possibles **V** (même sous le respect des étiquettes), et on se contentera en pratique d'un *minimum local approximatif*, obtenu après un nombre donné d'itérations N, identique pour tous les appariements, sans attendre nécessairement la stabilisation de H (une itération est définie comme une visite de tous les nœuds, avec amélioration locale de leurs positions individuelles). Nous verrons au §3.2 que cette méthode suboptimale suffira à réaliser de bons taux de reconnaissance, dans la mesure où la même approximation est appliquée à tous les couples et sert à un critère de classification comparatif.

Le résultat de cette minimisation dépend aussi d'un *ordre de visite* des nœuds, c'est à dire d'une bijection ω aléatoire, qui à tout $s \in O^\alpha$ associe un numéro d'ordre entre 1 et n_α (cf. §2.4.c). Puisque les graphes sont variables selon les images, cet ordre sera nécessairement différent d'un caractère à l'autre : en revanche, on conservera le même ordre pour tous les déformations de G^α, c'est à dire pour calculer toutes les distances $d(\Phi^\alpha, \Phi)_{\Phi \in D}$. Donc, à chaque caractère α correspondra un ordre de visite particulier, $\omega^\alpha(s)$, choisi au hasard une fois pour toutes : dans les simulations numériques, le choix des 1200 bijections $\{\omega^\alpha\}$ sera le résultat d'un générateur aléatoire, c'est à dire d'une suite déterministe initiée par un "germe" x (réel compris entre 0 et 1). Bien sûr, pour être fiable, la distance $d(\Phi^\alpha, \Phi^\beta)$ devra être robuste par rapport aux changements de $\omega^\alpha(s)$, dus à des changements de x, ce que nous aurons l'occasion de vérifier au §3.2.

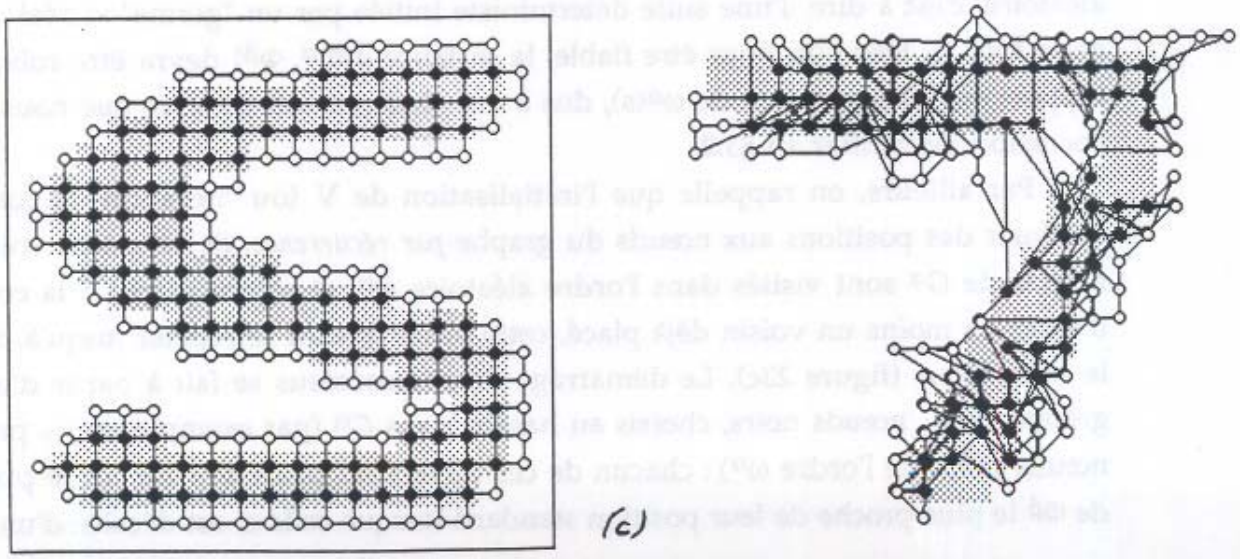
Par ailleurs, on rappelle que l'initialisation de **V** (ou "itération-0") consiste à attribuer des positions aux nœuds du graphe *par récurrence*, de voisin en voisin : les nœuds de G^α sont visités dans l'ordre aléatoire ω^α, et seront placés à la condition d'avoir au moins un voisin déjà placé, cette visite devant se répéter jusqu'à ce qu'ils le soient tous (figure 23c). Le démarrage de se processus se fait à partir d'un petit groupe de n_1 nœuds noirs, choisis au hasard dans G^α (par exemple les n_1 premiers nœuds noirs de l'ordre ω^α) : chacun de ces nœuds est placé d'office sur le pixel noir de Φ^β le plus proche de leur position standard lorsque celle-ci est décalée d'un



(a)

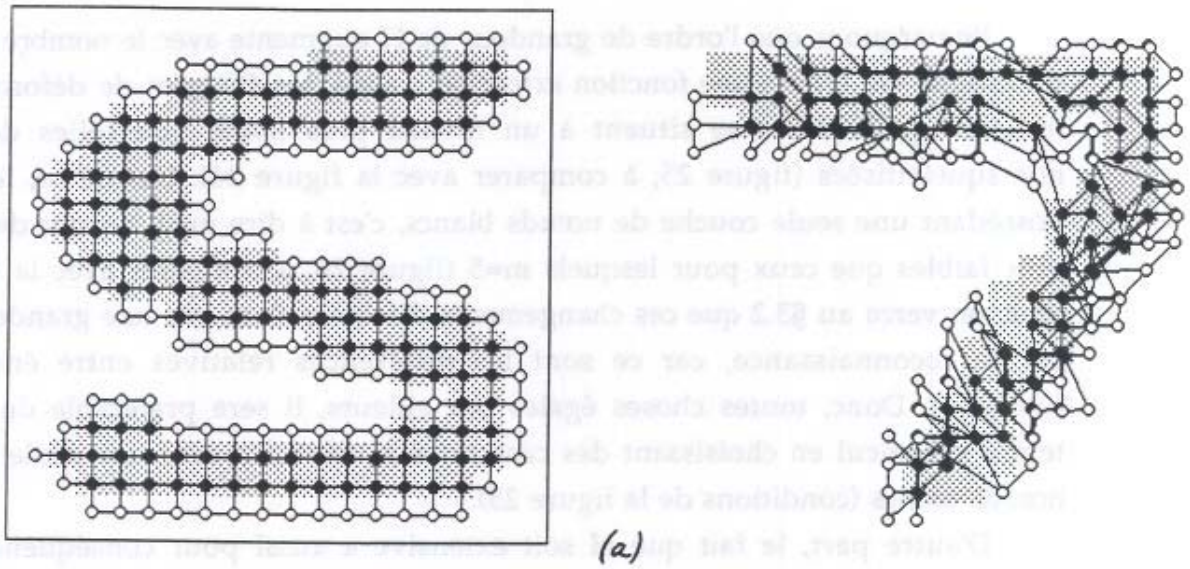


(b)



(c)

Figure 23 : les trois étapes de l'itération-0 dans la déformation d'un caractère "5" sur un caractère "7". A gauche, l'image Φ^{α} = "5" est couverte par sa représentation standard en graphe (G^{α}, $X^{\alpha 0}$), avec m=1. A droite, ce même graphe est graduellement déformé sur l'image Φ^{β} = "7". (a) Position préliminaire avant l'itération-0 : le graphe de α est dans un état non-déformé, translaté de telle sorte que les centres de masse des deux images coïncident (marqués d'un cercle). Les nœuds se trouvent alors en des positions virtuelles qui ne respectent pas leurs étiquettes, mais qui serviront de référence à l'initialisation. (b) Début de l'itération-0 : n_1 premiers nœuds noirs choisis au hasard (ici, n_1=10) sont placés directement sur les pixels noirs les plus proches (des cercles permettent de repérer ces nœuds sur le graphe de référence). (c) Fin de l'itération-0 : tous les autres nœuds ont été placés à la suite des n_1 premiers nœuds en tentant de conserver les relations de voisinages de proche en proche, mais sous le respect strict des étiquettes (nœud noir sur pixel noir, nœud blanc sur pixel blanc). Les importantes différences entre le "5" et le "7" se ressentent alors fortement dans les distortions causées par ces déplacements, et par la valeur de l'énergie qui en résulte : $d(\Phi^{\alpha}, \Phi^{\beta})$ = 878 (avec κ=2).



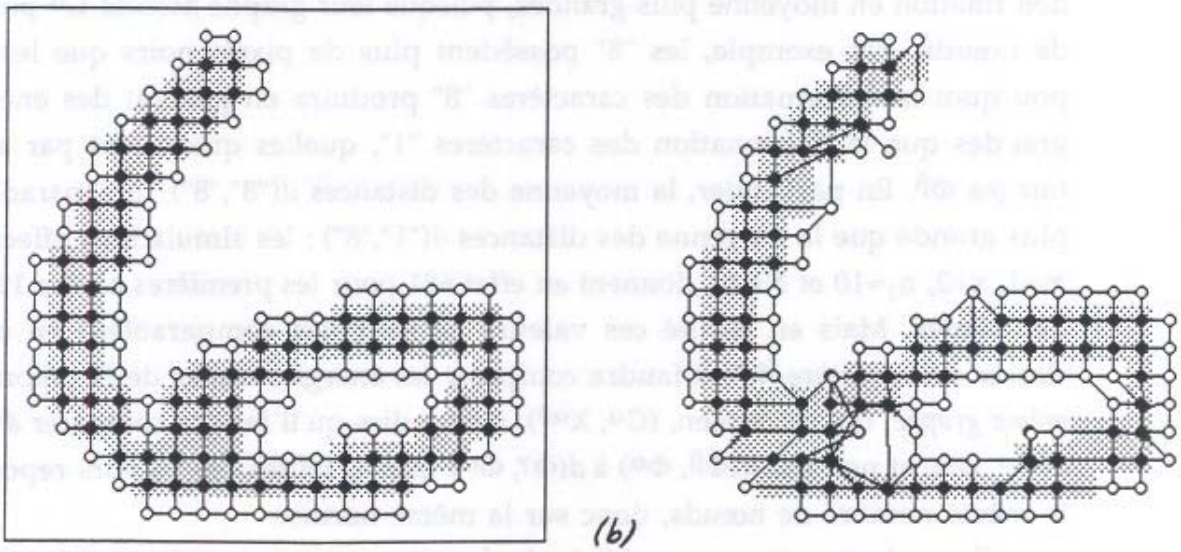


Figure 24 : Deux exemples de déformations. Dans les mêmes conditions que celles de la figure 3 (m=1, κ=2, n_1=10), l'optimisation a été poursuivie au-delà de l'itération-0 pendant N=10 itérations, ce qui a permis aux valeurs d'énergie de se stabiliser dans un minimum local. *(a)* La déformation du "5" sur le "7" (mêmes caractères que précédemment) aboutit à d = 684, tandis que la déformation d'un "6" sur un "6" *(b)* produit une énergie nettement plus faible, d = 88.

vecteur (a,b) (figure 23b). La translation initiale (a,b) sera variable en fonction des appariements : elle dépendra des images Φ^α et Φ^β et sera choisie ici de telle sorte que les "centres de masse" de ces images coïncident (figure 23a), où le centre de masse d'une image désigne le *barycentre de ses pixels noirs*. Pour Φ^α, les coordonnées entières les plus proches de ce centre seront notées (i^α, j^α), donc, dans toute la suite, on posera : $(a, b) = (i^\alpha, j^\alpha) - (i^\beta, j^\beta)$. On peut voir un exemple d'initialisation sur la figure 23. La figure 24 présente le résultat de deux appariements poursuivis sur 10 itérations, l'un avec des caractères différents, l'autre avec des caractères semblables.

En résumé, $d(\Phi^\alpha, \Phi^\beta)$ dépend des trois paramètres suivants : un germe aléatoire x déterminant l'ordre de visite des nœuds pour chaque graphe, le nombre des premiers nœuds placés a priori n_1, et le nombre d'itérations supplémentaires N.

Remarquons que l'ordre de grandeur de H augmente avec le nombre de nœuds du graphe G^α (H est une fonction *extensive*) : ainsi, les énergies de déformation des images squelettisées se situent à un niveau plus faible que celles des images non-squelettisées (figure 25, à comparer avec la figure 24). De même, les graphes possédant une seule couche de nœuds blancs, c'est à dire m=1, auront des énergies plus faibles que ceux pour lesquels m=5 (figure 26, à comparer avec la figure 25). Mais on verra au §3.2 que ces changements d'échelle n'ont pas une grande influence sur la reconnaissance, car ce sont les différences relatives entre énergies qui comptent. Donc, toutes choses égales par ailleurs, il sera préférable de limiter le temps de calcul en choisissant des caractères squelettisés avec une seule couche de nœuds blancs (conditions de la figure 25).

D'autre part, le fait que H soit extensive a aussi pour conséquence que les images Φ^α plus fournies en pixels noirs seront associées à des énergies de déformation en moyenne plus grandes, puisque leur graphe associé G^α possède plus de nœuds. Par exemple, les "8" possèdent plus de pixels noirs que les "1" : c'est pourquoi la déformation des caractères "8" produira en général des énergies plus grandes que la déformation des caractères "1", quelles que soient par ailleurs les images Φ^β. En particulier, la moyenne des distances d("8","8") sera paradoxalement plus grande que la moyenne des distances d("1","8") : les simulations effectuées avec m=1, κ=2, n_1=10 et N=10, donnent en effet 681 pour les premières contre 124 pour les deuxièmes. Mais en réalité ces valeurs ne sont pas comparables : en effet, pour classer un caractère Φ^α, il faudra comparer les énergies issues de la déformation du *même graphe*, qui est le sien, $(G^\alpha, X^{\alpha 0})$, c'est à dire qu'il faudra comparer $d(\Phi^\alpha, \Phi^\beta)$ à $d(\Phi^\alpha, \Phi^\gamma)$, et non pas $d(\Phi^\beta, \Phi^\alpha)$ à $d(\Phi^\gamma, \Phi^\alpha)$. Ainsi, toutes ces valeurs reposeront sur le même nombre de nœuds, donc sur la même norme.

Cependant, cette non-symétrie de d peut aussi mener à des ambiguïtés, comme on le montre un peu plus bas, et de plus, si l'on veut utiliser le critère des plus proches voisins, ou tout autre critère de classification fondé sur une distance, il est préférable de disposer d'une fonction symétrique. C'est pourquoi dans toute la suite on remplacera d par une version symétrisée d^*, définie de la façon suivante :

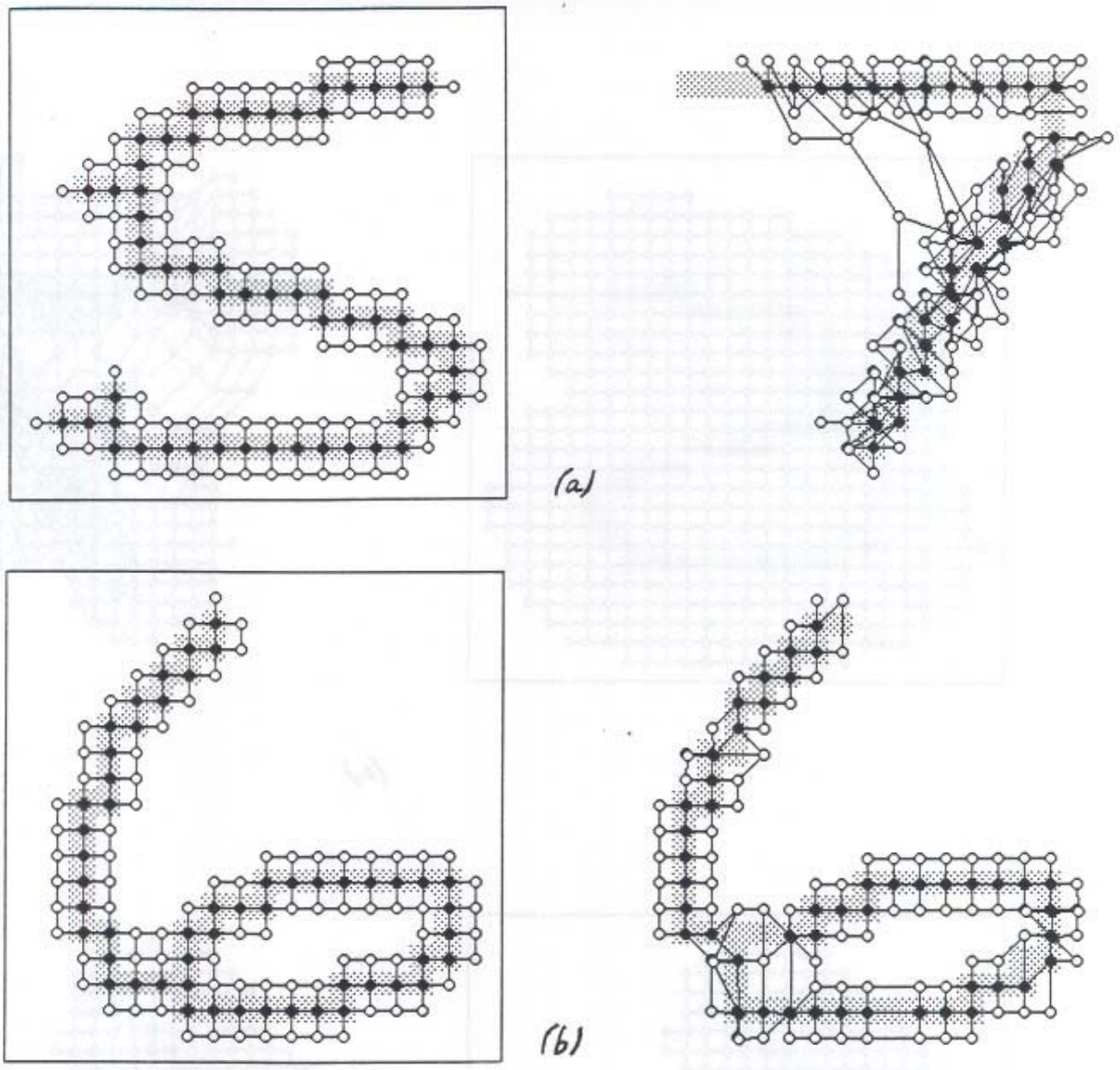
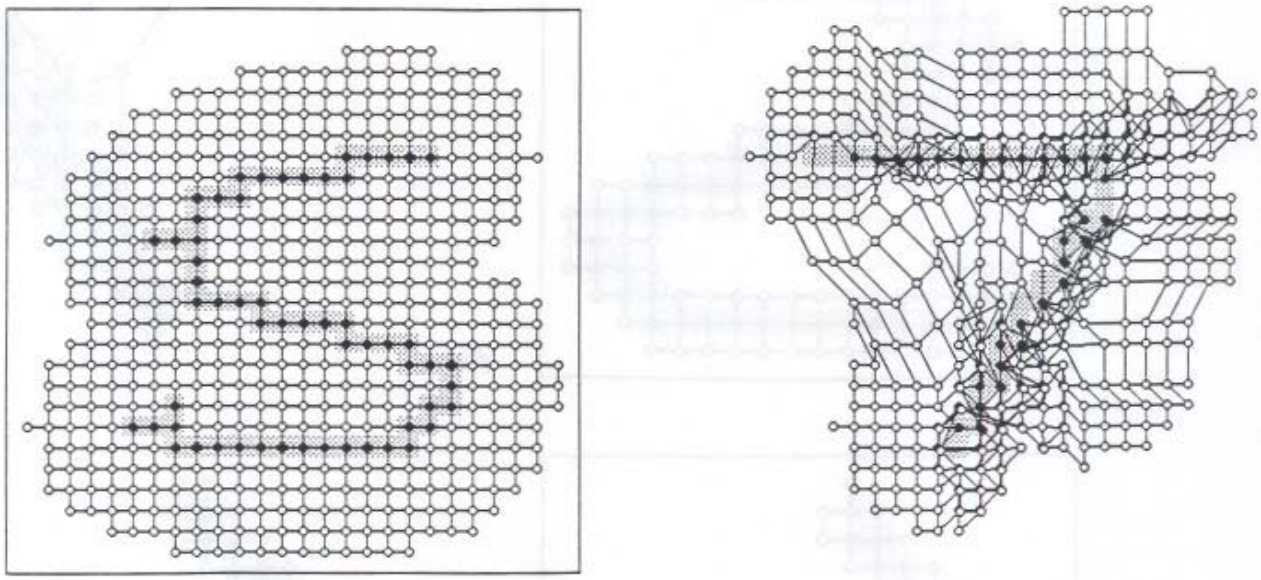


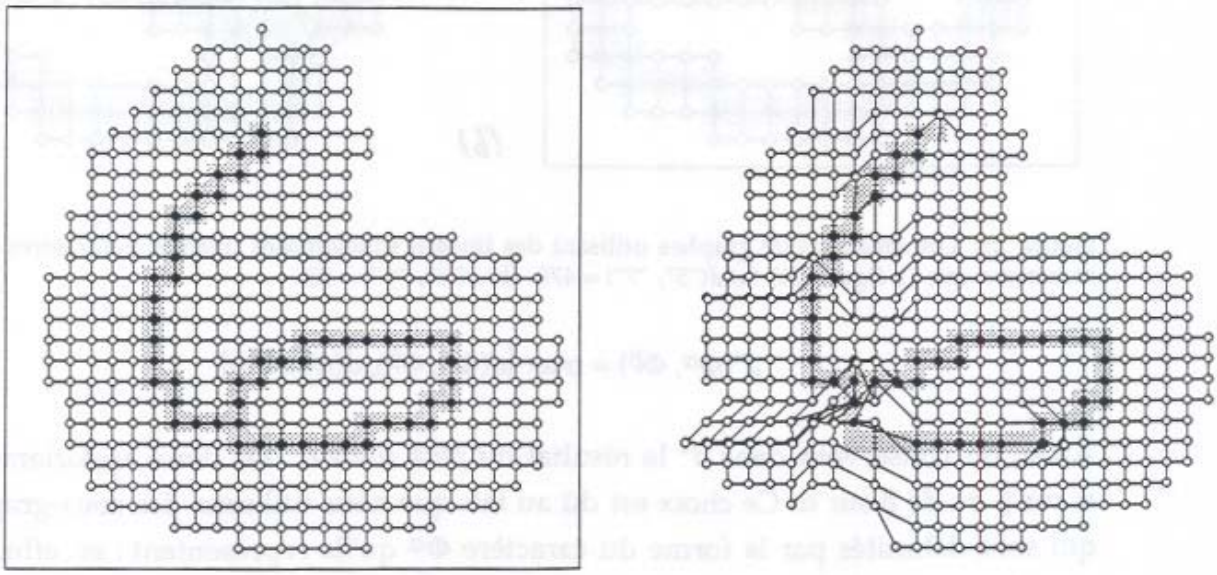
Figure 25 : Déformations de graphes utilisant des images squelettisées (mêmes paramètres et mêmes caractères que la figure 24) *(a)* d("5", "7") = 478. *(b)* d("6", "6") = 73.

$$d^*(\Phi^\alpha, \Phi^\beta) = \max \{d(\Phi^\alpha, \Phi^\beta), d(\Phi^\beta, \Phi^\alpha)\}$$

Ainsi, on conservera dans d^* le résultat du *plus coûteux* des deux appariements, de α sur β ou de β sur α. Ce choix est dû au fait que nous utilisons des sous-graphes G^α qui sont délimités par la forme du caractère Φ^α qu'ils représentent : en effet, de tels graphes peuvent être placés presque sans déformation sur des zones de Φ^β qui incluent la forme de Φ^α, mais qui possèdent par ailleurs d'autres caractéristiques la rendant très différente de Φ^α, ces caractéristiques étant hors de portée de G^α. Par exemple, un "3" peut être quasiment inclus dans un caractère "8" : c'est pourquoi, il est nécessaire de procéder à l'appariement réciproque, c'est à dire à la déformation du "8" sur le "3", pour révéler les zones du "8" que le "3" n'avait pas découvertes (figure 27).



(a)



(b)

Figure 26 : Déformations de graphes avec une large couche de nœuds blancs (ici, m=5) (images squelettisées de la figure 25) *(a)* d("5", "7") = 1444. *(b)* d("6", "6") = 198.

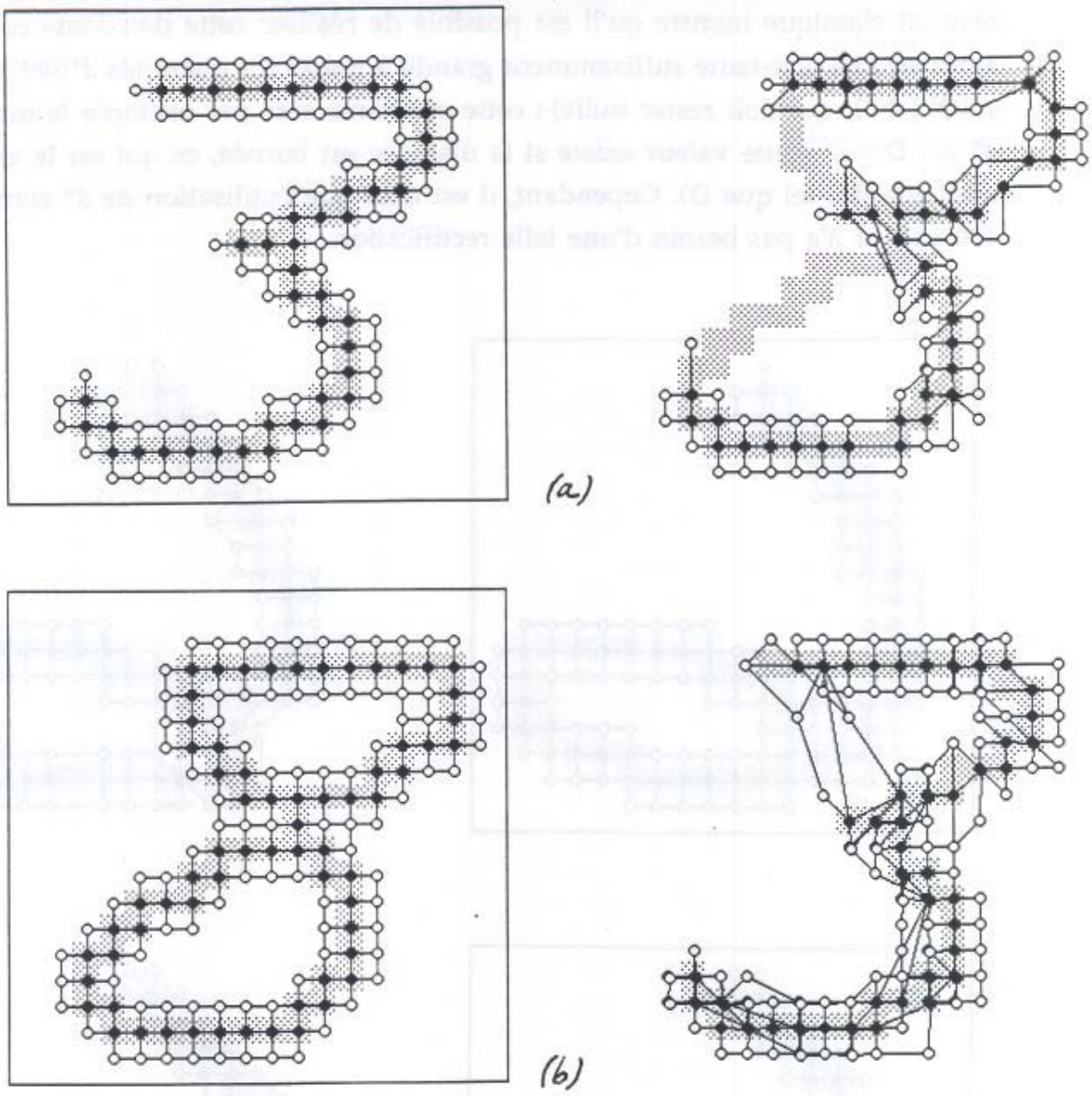


Figure 27 : Appariement mutuel d'un "3" et d'un "8" (pour plus de clarté, cet exemple utilise des caractères squelettisés). Les paramètres sont : m=1, κ=2, n_1=10 et N=10. *(a)* L'appariement du "3" sur le "8" provoque peu de déformations et fournit d("3","8") = 75, comme s'il s'agissait d'une déformation sur un "3". On remarque par exemple un étirement de la barre inférieure, trop courte d'une unité, ou la surélévation d'un nœud blanc (en bas, à gauche), obligé de se déplacer sur un pixel blanc (par contre, il n'était pas nécessaire que la barre supérieure se contracte à gauche et s'étire à droite : cette configuration suboptimale est un cas typique de minimum local). *(b)* L'appariement réciproque du "8" sur le "3" contraint fortement le graphe et résulte en une énergie beaucoup plus grande : d("8","3") = 302. L'arc convexe situé en bas à gauche du "8" s'est intégralement retourné sur l'arc concave du "3", tandis que le segment supérieur gauche s'est projeté en partie vers le haut et en partie vers le centre, laissant quelques nœuds blancs tiraillés entre ces deux pôles. En conclusion, les importantes différences entre ces caractères ne sont pas révélées par l'appariement (a) mais sont révélées par l'appariement (b), lequel donnera sa valeur à la distance finale : ainsi, d^*("3","8") = d^*("8","3") = 302.

En revanche, dans le cas où les formes de Φ^α et Φ^β sont semblables, les valeurs obtenues dans un sens ou dans l'autre sont voisines, et le fait de symétriser en prenant le maximum ne change pas le résultat (figure 28).

D'autre part, pour être qualifiée de véritable distance, d^* devrait encore satisfaire à l'inégalité triangulaire : $d^*(\Phi^\alpha, \Phi^\beta) \leq d^*(\Phi^\alpha, \Phi^\gamma) + d^*(\Phi^\gamma, \Phi^\beta)$. Or un résultat classique montre qu'il est possible de réaliser cette deuxième condition en ajoutant une constante suffisamment grande à toutes les quantités $d^*(\Phi^\alpha, \Phi^\beta)$ (sauf à $d^*(\Phi^\alpha, \Phi^\alpha)$, qui doit rester nulle) : cette constante sera par exemple le maximum de d^* sur $D \times D$ (cette valeur existe si la distance est bornée, ce qui est le cas dans un domaine fini tel que D). Cependant, il est clair que l'utilisation de d^* comme critère comparatif n'a pas besoin d'une telle rectification.

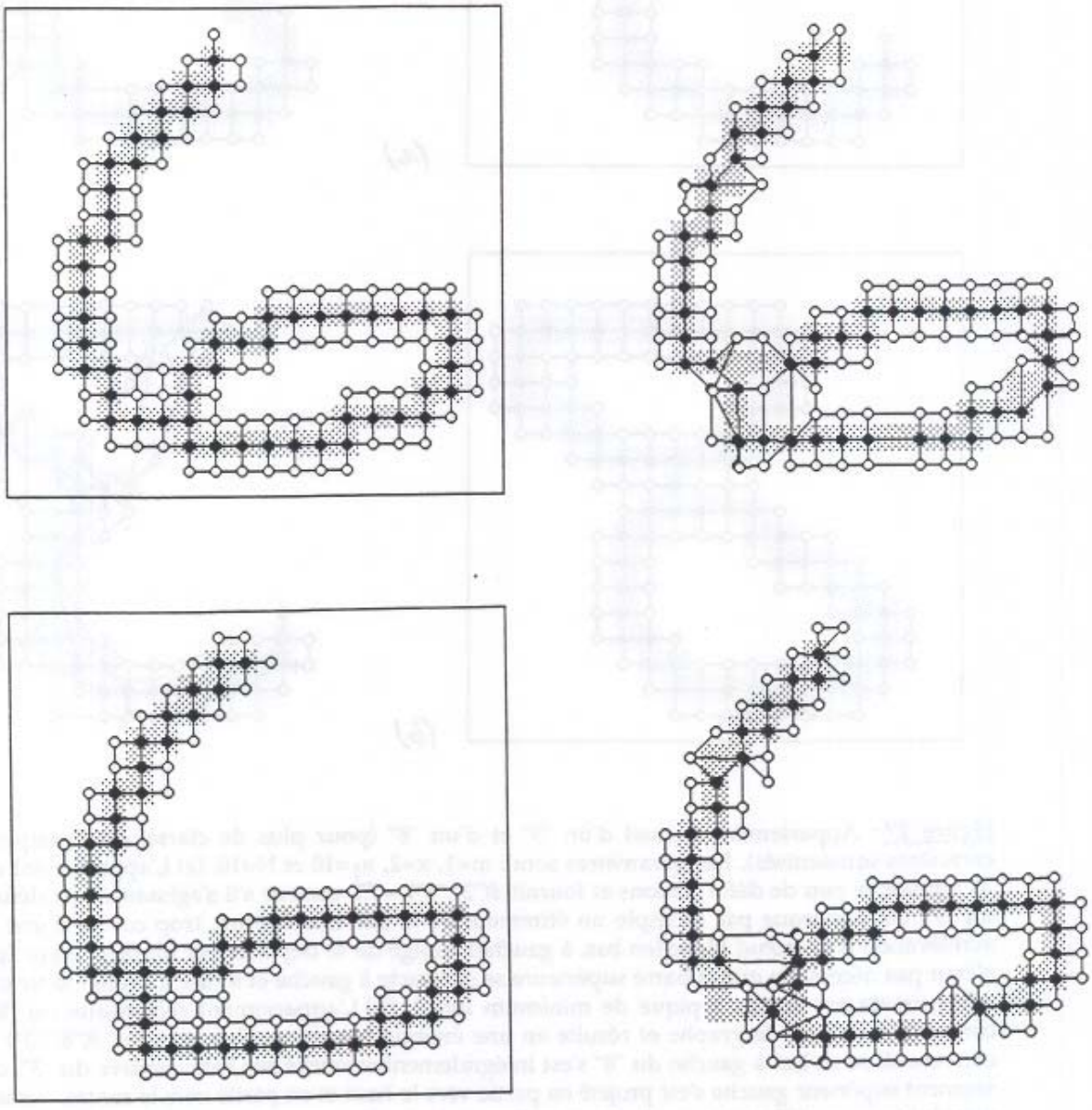


Figure 28 : Appariement mutuel de deux caractères "6" (caractères squelettisés de la figure 5b). Les paramètres sont les mêmes que pour la figure 27. Les images Φ^α et Φ^β présentent peu de différences, et l'énergie de déformation reste faible dans un sens comme dans un autre : d vaut 73 pour l'appariement *(a)* et 87 pour le *(b)*, donc, au total, d^* vaut 87.

3.2 Classification

Munis de la distance d^* dans l'ensemble des caractères D, nous sommes prêts à passer au stade du traitement statistique des données, en commençant par l'exercice suivant : un ensemble de prototypes est extrait *aléatoirement* de D, et un taux d'erreur de reconnaissance sur les exemples de test, ε, est calculé grâce au critère du plus proche voisin. En réitérant cette opération sur de nombreuses partitions de proportions identiques, on obtient finalement un *taux d'erreur moyen*, $\langle\varepsilon\rangle$, qui constituera un indicateur de performance fiable pour la méthode.

3.2.a Principe

La base des exemples est séparée en deux, équitablement sur les classes, c'est à dire de telle sorte qu'il y ait le même nombre de prototypes dans chaque classe, p, et donc le même nombre de tests par classe, $q = 120 - p$. L'ensemble de tous les prototypes sera noté P et celui des tests T, soit : $T = D \backslash P$ avec $\mathrm{card}(P) = 10p$, $\mathrm{card}(T) = 10q$. Les statistiques seront faites sur des partitions aléatoires *à p constant*. On notera également P^c et T^c les sous-ensembles de P et T à l'intérieur de chaque classe c, c'est à dire : $P^c = P \cap D^c$ et $T^c = T \cap D^c$. Le choix aléatoire des prototypes peut donc s'écrire :

$$P^c = \{\Phi^{c,\sigma(1)}, \ldots, \Phi^{c,\sigma(p)}\} \text{ et } T^c = \{\Phi^{c,\sigma(p+1)}, \ldots, \Phi^{c,\sigma(120)}\} = D^c \backslash P^c$$

où σ représente une permutation quelconque dans l'ensemble des entiers $\mu = 1...120$. La bijection σ est une variable probabiliste obéissant à une loi de tirage uniforme : ainsi, étant donné p, les exemples de D^c ont tous la même probabilité d'être tirés en μ-ème position, donc ils ont tous la même probablité $p/120$ d'être choisis comme prototypes. Cette permutation ne sera pas forcément la même d'une classe à l'autre, et on notera $\sigma = (\sigma_0, \sigma_1, \ldots, \sigma_9)$ la permutation globale de l'ensemble D, qui est composée de 10 permutations internes aux classes 0, 1, ... , 9. En résumé : la taille de l'ensemble des prototypes P est 10p, et son contenu est le résultat du mélange σ.

Etant donné l'ensemble de référence P, on définit alors une fonction de classification f_P fondée sur la distance d^* accompagnée d'un critère de décision : nous choisirons ici celui du premier plus proche voisin. Ainsi, l'attribution d'une classe à un caractère-test donné, $\Phi^\beta \in T$, se fera en référence au caractère-prototype Φ^α le plus proche selon d^* :

$$f_P(\Phi^\beta) = c \;\Leftrightarrow\; \exists\, \Phi^\alpha \in P^c;\ \forall\, \Phi^{\alpha'} \in P,\ d^*(\Phi^\beta, \Phi^\alpha) \le d^*(\Phi^\beta, \Phi^{\alpha'})$$

(remarquons que $\Phi^\alpha \in P^c$ signifie qu'il existe $\mu = 1...p$ tel que $\alpha = 120c + \sigma_c(\mu)$). Puis la réponse du classifieur $f_P(\Phi^\beta)$ est comparée à la vraie classe de Φ^β, ce qui permet de calculer un *taux d'erreur* de classification $\varepsilon(c)$ dans chaque classe c :

$$\varepsilon(c) = \frac{1}{q} \text{ card } \{\Phi^\beta \in T^c;\ f_P(\Phi^\beta) \neq c\}$$

Le taux d'erreur global ε sur tout l'ensemble de test T est alors donné par la moyenne des $\varepsilon(c)$ sur les ensembles T^c, puisque les classes contiennent le même nombre de caractères-tests, q :

$$\varepsilon = \frac{1}{10}\left(\varepsilon(0) + \varepsilon(1) + \ldots + \varepsilon(9)\right)$$

Finalement, pour obtenir une quantité fiable, on recalculera plusieurs fois ε, sur plusieurs ensembles de prototypes différents et on prendra la moyenne de ces valeurs comme évaluation finale de la méthode. On notera $\langle\varepsilon\rangle$ le taux d'erreur moyen, et on a donc : $\langle\varepsilon\rangle = E_\sigma[\varepsilon_\sigma]$, où E_σ désigne l'espérance sur la variable probabiliste σ. Rappelons que tout ceci est effectué à p constant : donc la moyenne sur σ revient à faire une moyenne sur les partitions aléatoires $D = P \cup T$ de proportion 10p/10q constante.

Le taux d'erreur moyen s'écrit en théorie :

$$\langle\varepsilon\rangle = \left(\frac{1}{120!}\right)^{10} \sum_{\sigma \in (\sigma_{120})^{10}} \varepsilon_\sigma$$

où σ_{120} représente le groupe des permutations des entiers de 1 à 120 : c'est le domaine des composantes σ_c de σ (permutations à l'intérieur des classes c), et donc σ varie dans $(\sigma_{120})^{10}$, qui est de cardinal $(120!)^{10} \approx 10^{1998}$. Toutefois, de très nombreuses combinaisons $\sigma = (\sigma_0, \sigma_1, \ldots, \sigma_9)$ produiront exactement les mêmes ensembles $\{P^0, P^1, \ldots, P^9\}$ et $\{T^0, T^1, \ldots, T^9\}$, pour p donné, puisque l'ordre des caractères à l'intérieur de P^c et T^c n'a pas d'importance : le facteur de redondance est donc (p!q!) dans chaque classe, c'est à dire au minimum $(60!)^2$ et au maximum 119!, ce qui fait que le nombre de σ qui produisent des partitions prototypes/tests vraiment distinctes est compris entre $(120!/119!)^{10} \approx 10^{21}$ et $(120!/60!60!)^{10} \approx 10^{350}$. En pratique, on se contentera d'une moyenne expérimentale calculée sur 10^2 ou 10^3 tirages aléatoires.

3.2.b Résultats

On rappelle qu'il existe plusieurs versions de distances élastiques d^*, qui sont fonction de diverses options concernant les graphes, la forme de l'énergie, et le processus d'optimisation. Pour récapituler, une distance d^* dépendra ici de six paramètres (cf. §3.1) : la couche de nœuds blancs m, le coefficient de la pénalité de superposition κ, la translation initiale (a,b), le nombre de nœuds placés en premier n_1, le germe aléatoire x déterminant les différents ordres de visite des nœuds, ainsi que le nombre d'itérations N. Comme on l'a signalé au §3.1, dans chaque calcul de $d(\Phi^\alpha, \Phi^\beta)$, (a,b) sera choisi de telle sorte que le centre de masse de Φ^α se trouve sur celui de Φ^β, c'est à dire : $(a, b) = (i^\alpha, j^\alpha) - (i^\beta, j^\beta)$. D'autre part, on posera dans toute la suite : n_1=10, ce qui représentera en général un dixième ou un vingtième du nombre de nœuds de G^α. Etant donné ces choix, il nous reste à explorer les combinaisons des

quatre autres paramètres : m, κ, N et x. Sauf mention contraire, les simulations utiliseront les caractères *squelettisés*.

Compte tenu du fait que le temps de calcul augmente avec m et N, on commencera par le choix simple m=1 et N=0 (avec un coefficient de référence κ=2) : ainsi, la valeur d'énergie retenue dans chaque appariement sera celle de l'itération-0. La courbe de la figure 29 présente les taux de reconnaissance obtenus avec ces paramètres, sur la base d'un certain $x=x_1$. En abscisse, on fait varier le nombre total de prototypes, 10p, et en ordonnée on porte le taux d'erreur moyen ⟨ε⟩ qui a été calculé sur 10^3 partitions aléatoires de proportions 10p/10q : par exemple, pour p=60, c'est à dire pour des équipartitions 600/600, on obtient ⟨ε⟩=0,3%, ce qui veut dire que sur chaque lot de 600 caractères-tests, le nombre moyen de caractères mal classés est inférieur à 2 (ce sont les caractères dont le premier plus proche voisin parmi les 600 prototypes n'est pas de leur classe). Avec seulement 100 prototypes en tout (p=10), l'erreur augmente à ⟨ε⟩=2,3%, c'est à dire en moyenne 25 caractères mal classés sur les 1100 tests restants.

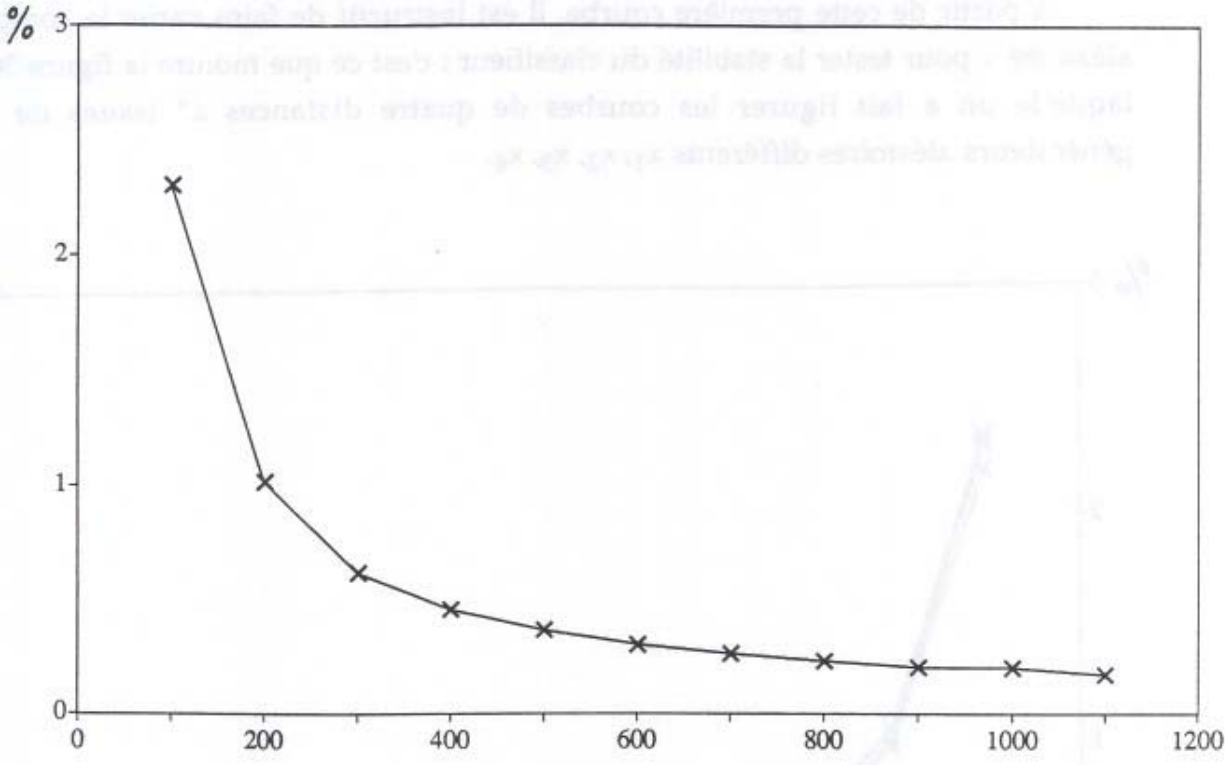


Figure 29 : Taux d'erreur moyen du classifieur utilisant la distance d^*, avec les paramètres m=1, κ=2, N=0. Chaque point représente une moyenne sur 10^3 partitions aléatoires de la base des 1200 caractères (à p constant), une partition contenant 10p prototypes et 10q=1200–10p caractères-tests.

Il est clair que l'erreur est d'autant plus faible que le nombre de prototypes est grand, comme on peut le voir sur cette courbe. Cependant, lorsque le nombre de prototypes approche la taille maximale de la base (p≥100), donc lorsque le nombre de tests est nécessairement réduit à quelques caractères par classe (q≤10), ⟨ε⟩ tend vers une valeur limite non nécessairement nulle qui correspond aux nombres de caractères mal classés "irréductibles". En effet, à ce niveau, la décision de classification d'un caractère-test Φ^β ne dépend pratiquement plus de la composition de l'ensemble P puisque celui-ci recouvre presque tout D :

cette décision dépend seulement du plus proche voisin dans l'absolu, c'est à dire *du* caractère $\Phi^{\alpha\neq\beta}$ qui est le plus proche de Φ^{β} dans toute la base (la probabilité pour que Φ^{α} soit utilisé comme test, i.e. soit indisponible pour servir de prototype à Φ^{β}, étant très faible). Par conséquent, en dehors de toute partition P/T, la base D est a priori scindée en deux parties (pour une distance d^* donnée) : l'ensemble D^+ des caractères dont le plus proche voisin est de la même classe (en nombre N^+), et l'ensemble D^- des caractères dont le plus proche voisin n'est pas de la même classe (en nombre $N^- << N^+$). Lorsque p approche 120, les premiers seront alors systématiquement bien classés, et les deuxièmes systématiquement mal classés : le taux d'erreur irréductible est donc $\langle\varepsilon\rangle_{lim} = N^-/1200$, et les calculs statistiques ne peuvent que confirmer cette valeur théorique, en y ajoutant cependant des fluctuations qui seront d'autant plus grandes que l'ensemble de test est petit, ce qui explique l'imparfaite stabilité de la courbe expérimentale pour les grandes valeurs de p (en effet, le taux d'erreur associé à une partition de 10q caractères-tests est au mieux 0 et au pire $N^-/10q$, son espérance théorique valant constamment $N^-/1200$: c'est pourquoi son écart-type augmente lorsque q diminue). Dans le cas de la figure 29, l'analyse détaillée des distances met en évidence $N^-=2$ caractères irréductibles, ce qui donne $\langle\varepsilon\rangle_{lim}=2/1200=0{,}17\%$.

A partir de cette première courbe, il est instructif de faire varier la composante aléatoire x pour tester la stabilité du classifieur : c'est ce que montre la figure 30, dans laquelle on a fait figurer les courbes de quatre distances d^* issues de quatre générateurs aléatoires différents x_1, x_2, x_3, x_4.

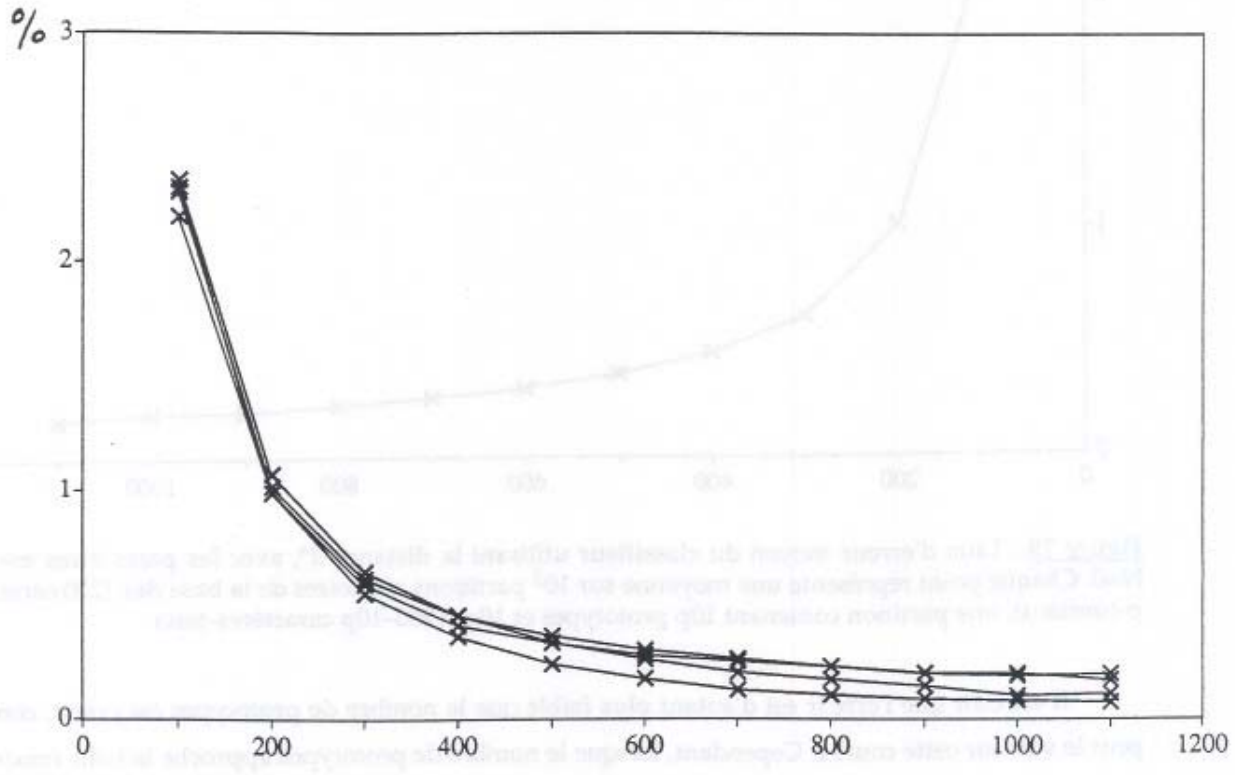


Figure 30 : Taux d'erreur calculés sur quatre variantes de d^* correspondant à quatre valeurs différentes de x (mêmes paramètres que dans la figure 29). La faible divergence des courbes prouve la robustesse de la distance par rapport aux fluctuations aléatoires du processus d'initialisation. On voit que les différences finales sont de l'ordre du dixième de pourcent : donc par exemple pour 600 prototypes et 600 tests, l'écart entre la meilleure performance et la moins bonne est d'au plus 1 caractère.

On constate alors que les performances des quatre classifieurs sont quasiment identiques : ceci prouve que l'optimisation de l'énergie H aboutissant à d^* est peu sensible à un changement de l'ordre de visite des nœuds dans les graphes G^α (on rappelle que cet ordre de visite détermine les positions initiales des nœuds, V(0), qui sont d'ailleurs la seule référence pour le calcul de d^* dans le cas présent où N=0). Signalons que l'examen plus détaillé des valeurs de H prouve en effet que, avant même l'application d'un critère de classification, les fluctuations de H selon x restent faibles.

Les figures 31 et 32 présentent quelques variations des paramètres m et κ (avec N=0 et $x=x_1$) : elles montrent que l'épaisseur de la couche de nœuds blancs n'a pratiquement pas d'influence sur les performances, mais qu'il est en revanche utile de garder un coefficient κ non-nul.

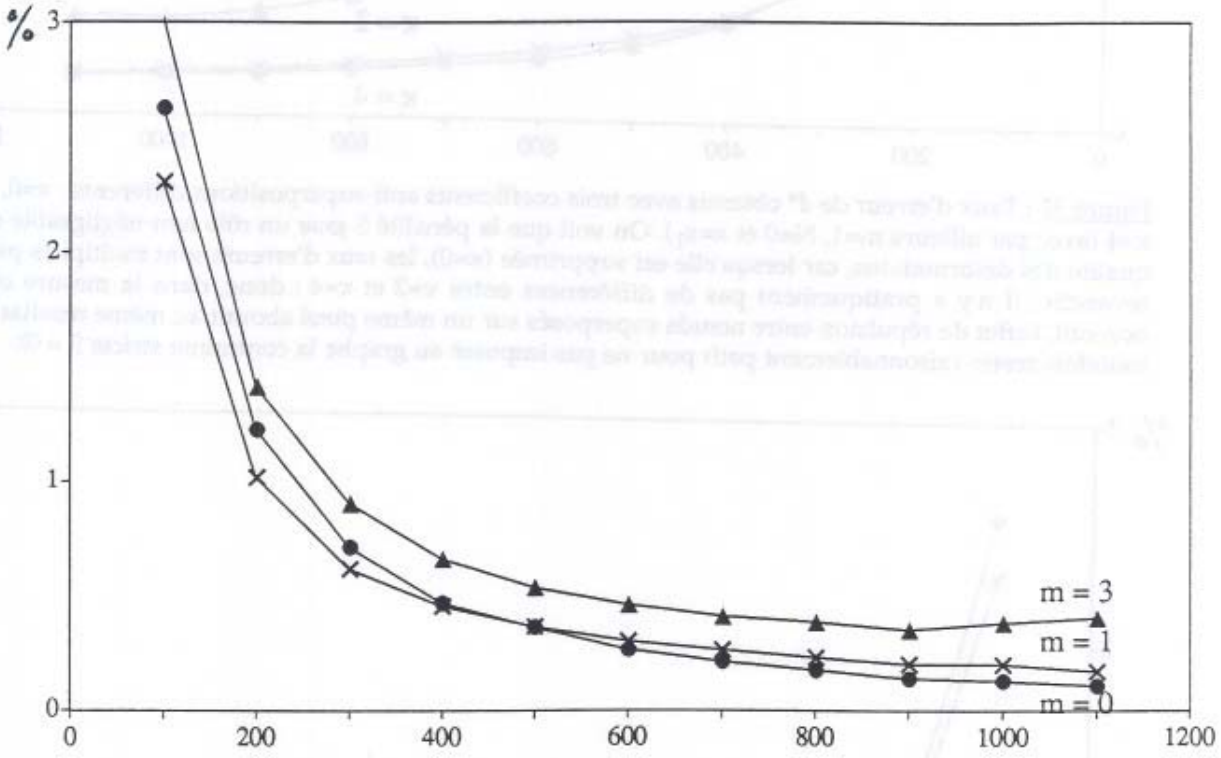


Figure 31 : Taux d'erreur de d^* obtenus avec trois épaisseurs différentes de nœuds blancs : m=0, m=1 et m=3 (avec par ailleurs κ=2, N=0 et $x=x_1$). Les courbes sont quasiment identiques avec des graphes uniquement composés de nœuds noirs (m=0) ou avec des graphes possédant une couche de nœuds blancs (m=1). Donc le fait d'avoir défini d^* comme le maximum des valeurs réciproques de d a suffi à traiter les problèmes d'inclusion des formes noires (cf. figure 27), et à ce niveau, la couche de nœuds blancs permettant de réveler la frontière du caractère n'est plus nécessaire (cette constatation n'exclut bien sûr pas l'utilisation des nœuds blancs ou d'autres types d'étiquetages pour traiter des bases de données plus complexes). Lorsque le graphe est épaissi (m=3), les performances semblent un peu moins bonnes : cependant, encore une fois, les différences jouent sur 1 ou 2 caractères irréductibles supplémentaires. La proximité de ces courbes est donc plus révélatrice sur la robustesse de l'algorithme que leur éloignement relatif.

Par ailleurs, sur la figure 33, l'emploi des caractères squelettisés est comparé à celui des caractères non-squelettisés. Bien sûr, contrairement aux variations de x qui produisent des valeurs de d^* semblables, les variations de m ou κ correspondent à des ordres de grandeurs de d^* différents, les valeurs les plus faibles étant atteintes

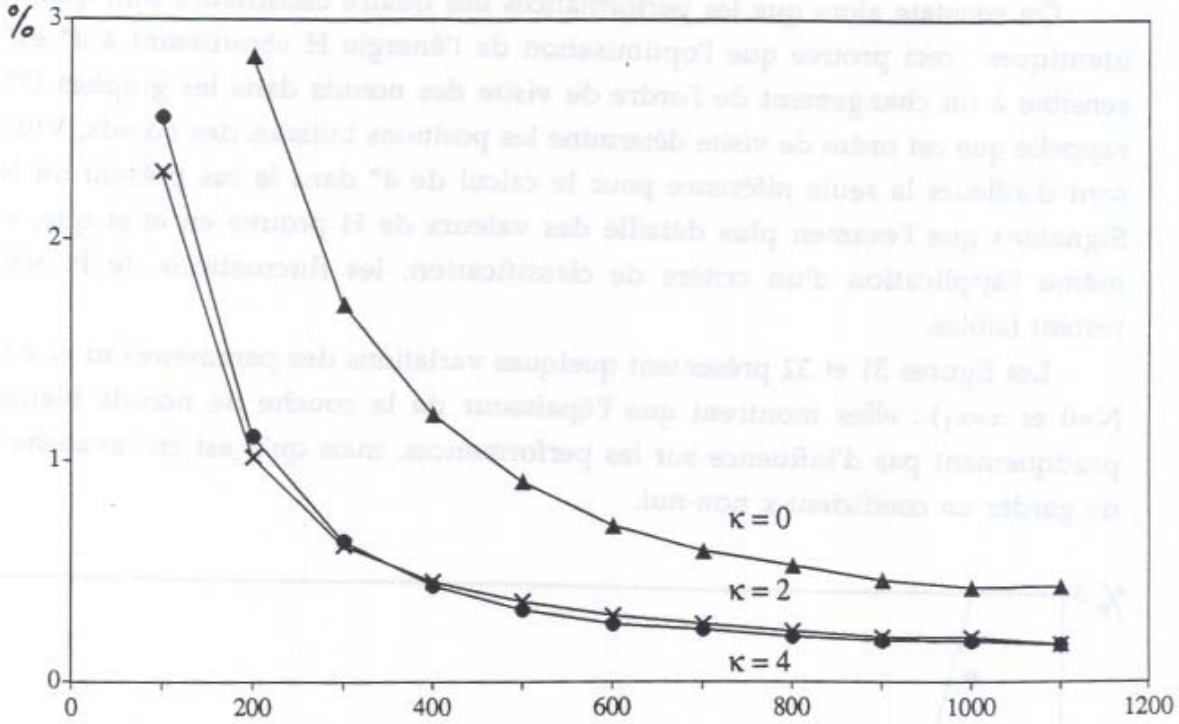


Figure 32 : Taux d'erreur de d^* obtenus avec trois coefficients anti-superpositions différents : κ=0, κ=2 et κ=4 (avec par ailleurs m=1, N=0 et $x=x_1$). On voit que la pénalité S joue un rôle non-négligeable dans la qualité des déformations, car lorsqu'elle est supprimée (κ=0), les taux d'erreurs sont multipliés par 3. En revanche, il n'y a pratiquement pas de différences entre κ=2 et κ=4 : donc, dans la mesure où κ est non-nul, l'effet de répulsion entre nœuds superposés sur un même pixel aboutit au même résultat (κ doit toutefois rester raisonnablement petit pour ne pas imposer au graphe la contrainte stricte S = 0).

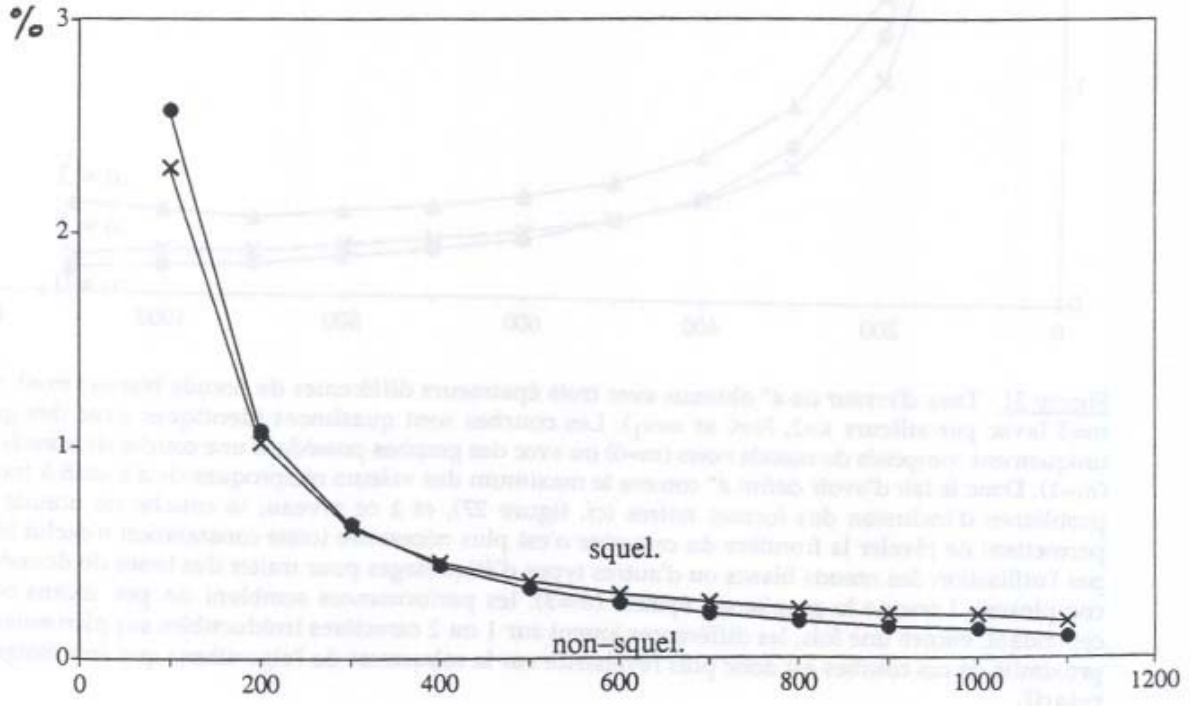


Figure 33 : Taux d'erreur sur la base des caractères squelettisés, comparé au taux d'erreur moyen sur la base des caractères non-squelettisés (m=1, κ=2, N=0, $x=x_1$). On constate à nouveau que les performances sont équivalentes : donc, dans ce cas, le seul avantage de la squelettisation est de réduire le temps de calcul en diminuant le nombre de nœuds. Cependant, en permettant de faire abstraction de l'épaisseur des caractères, la squelettisation pourra apporter une véritable amélioration dans des problèmes plus complexes.

pour m=0, κ=0, et les images squelettisées (cf. remarque du §3.1.c, figures 25 et 26). En revanche, on constate que les courbes de taux d'erreur restent toujours très proches, ce qui prouve que les *différences relatives* entre distances sont conservées et que le classifieur est robuste aux changements de paramètres.

Jusqu'à présent toutes ces simulations ont été arrêtées à l'itération-0 : ainsi, les faibles taux d'erreur rencontrés montrent que la simple étape d'initialisation, qui est pourtant encore loin du minimum idéal de l'énergie, suffit déjà à produire des énergies de déformation qui sont de bons révélateurs des différences ou des similitudes entre caractères. Il est toutefois intéressant d'étudier les avantages d'une optimisation plus poussée, c'est à dire N > 0 (figure 34). On constate alors que la poursuite des itérations permet de creuser davantage les différences relatives entre distances, et de rétablir un classement globalement meilleur. Bien sûr, dans l'ensemble, les valeurs d'énergie diminuent à peu près au même rythme (voir la figure 35), cependant, du point de vue d'un exemple de test individuel $\Phi\beta$, les caractères de sa classe se rapprochent de lui un peu plus vite que les caractères des autres classes, et il se produit alors quelques inversions dans l'ordre de ses préférences au profit de sa classe. A la limite, cette amélioration a pour conséquence d'éliminer quelques uns des derniers caractères irréductibles, en remplaçant le premier voisin de mauvaise classe par un nouveau premier voisin de bonne classe.

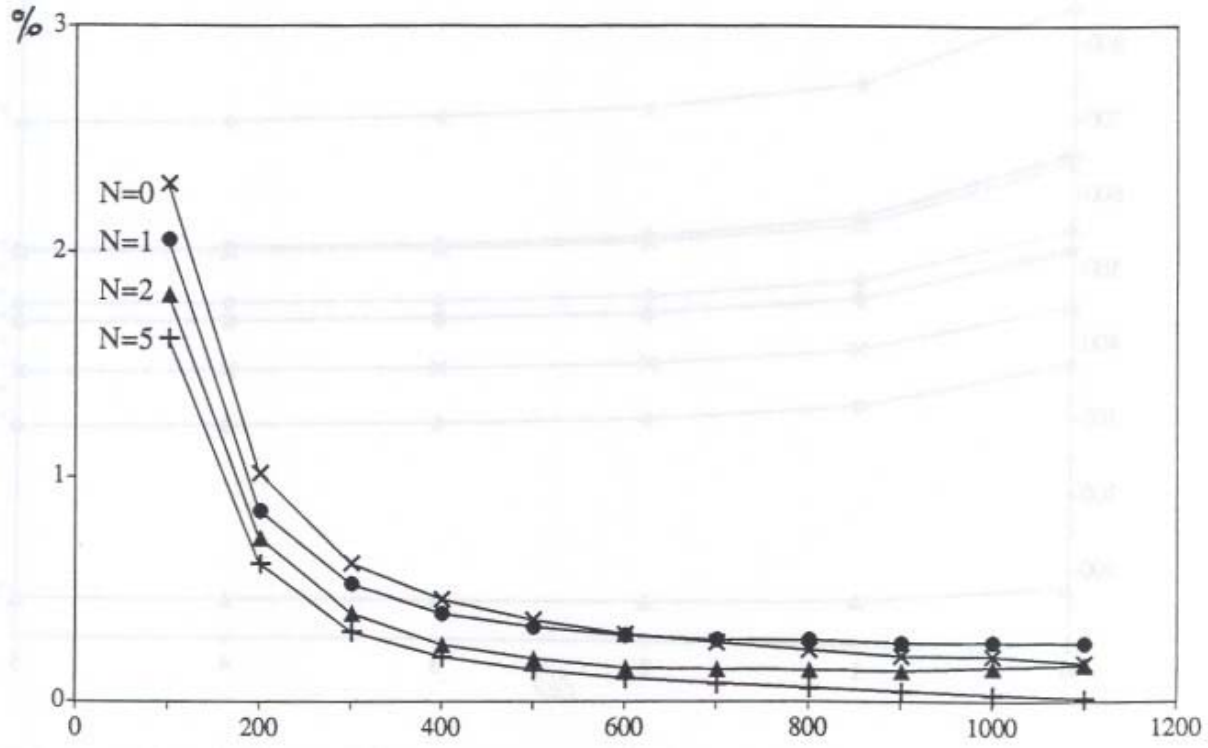


Figure 34 : Diminution du taux d'erreur avec la poursuite des itérations (m=1, κ=2, x=x_1). Une légère amélioration des performances apparaît progressivement avec N=1, N=2, puis le taux d'erreur stagne à sa meilleure valeur à partir de N≥5 : ceci est dû au fait que les valeurs d'énergie ont déjà atteint leurs minima locaux respectifs (voir la figure 37)

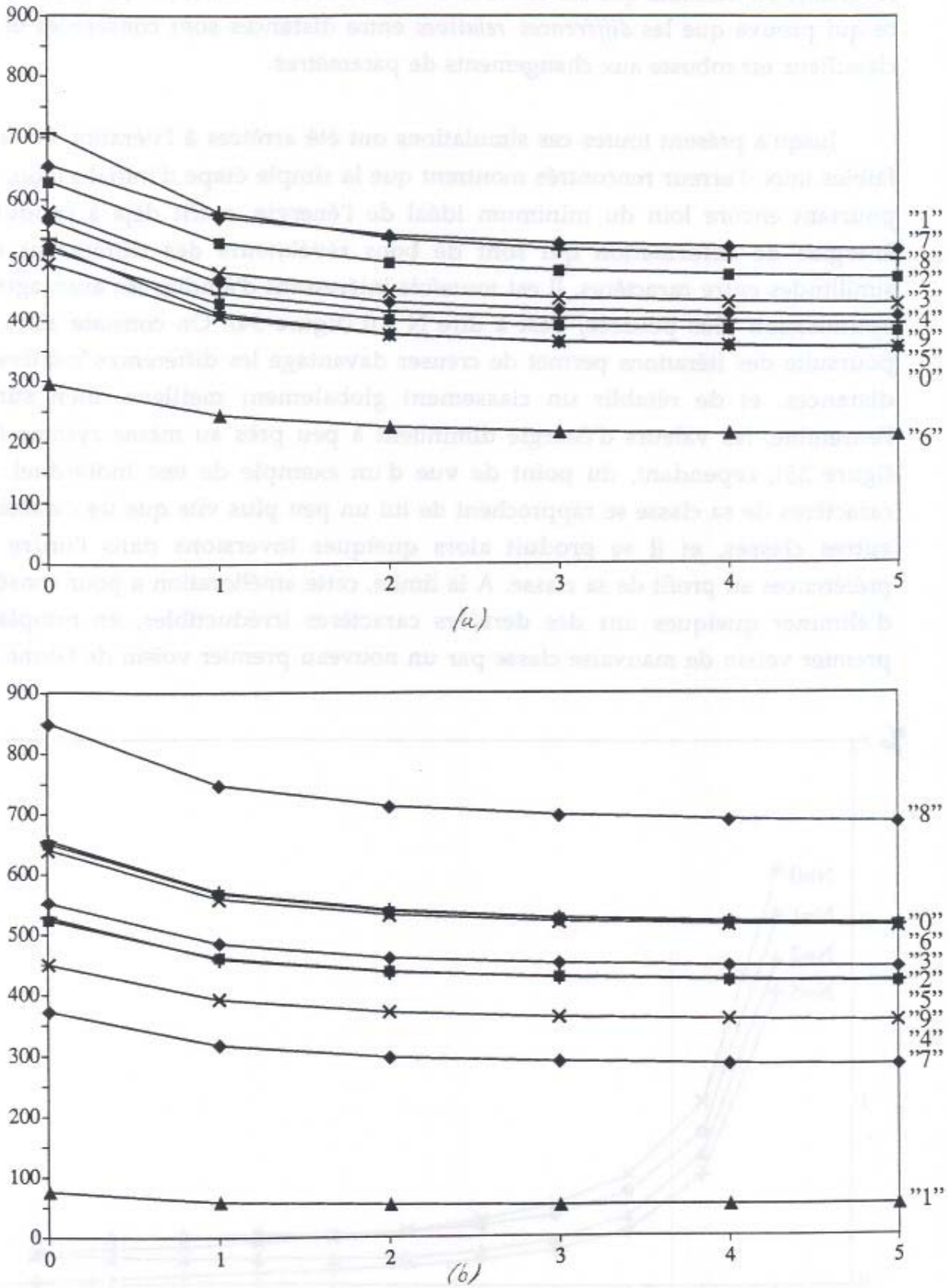


Figure 35 : Evolution des distances de classes au cours des itérations (m=1, κ=2, x=x_1). Ces valeurs moyennes ont été calculées jusqu'à N=10, mais leur stationnarité est quasi-parfaite pour N≥5, et n'a pas été représentée ici. *(a)* Du point de vue des caractères "6", ce sont bien sûr les "6" qui sont les plus proches (⟨d^*⟩≈200 avec N=5), tandis que les autres classes forment un groupe à part (les distances vont de 350 pour les "0" à 520 pour les "1"). *(b)* Du point de vue des "1", la séparation est encore plus nette, mais le groupe des autres classes est plus étalé.

L'étude des valeurs de d^* montre la bonne séparation des classes que peut opérer cette distance : sur la figure 35, on représenté l'évolution des *distances de classes* au cours des itérations, c'est à dire les distances moyennes à l'intérieur de la base D (version squelettisée) entre, d'une part, les caractères d'une classe c et, d'autre part, les caractères d'une classe c´. Il s'agit donc de la quantité :

$$\langle d^*(c, c')\rangle = \frac{1}{120^2} \sum_{\Phi^\alpha \in D^c} \sum_{\Phi^\beta \in D^{c'}} d^*(\Phi^\alpha, \Phi^\beta)$$

Par exemple, la figure 35a montre l'éloignement moyen des classes de 0 à 9 du point de vue de la classe 6, c'est à dire $\langle d^*(6, c')\rangle$ pour $c' = 0...9$: il ressort que $\langle d^*(6, 6)\rangle$ est nettement inférieur à toutes les autres valeurs. De même, la figure 35b montre que, du point de vue de la classe 1, les "1" sont de loin les plus favorisés. On voit aussi que les distances de classes diminuent globalement avec la poursuite des itérations, et qu'elles stationnent à partir de N=5 (ce qui explique la stationnarité des taux d'erreur de la figure 34), mais la propriété la plus importante de d^*, c'est à dire le fossé entre $c' = c$ d'une part et le groupe $\{c' \neq c\}$ d'autre part, persiste dans tous les cas.

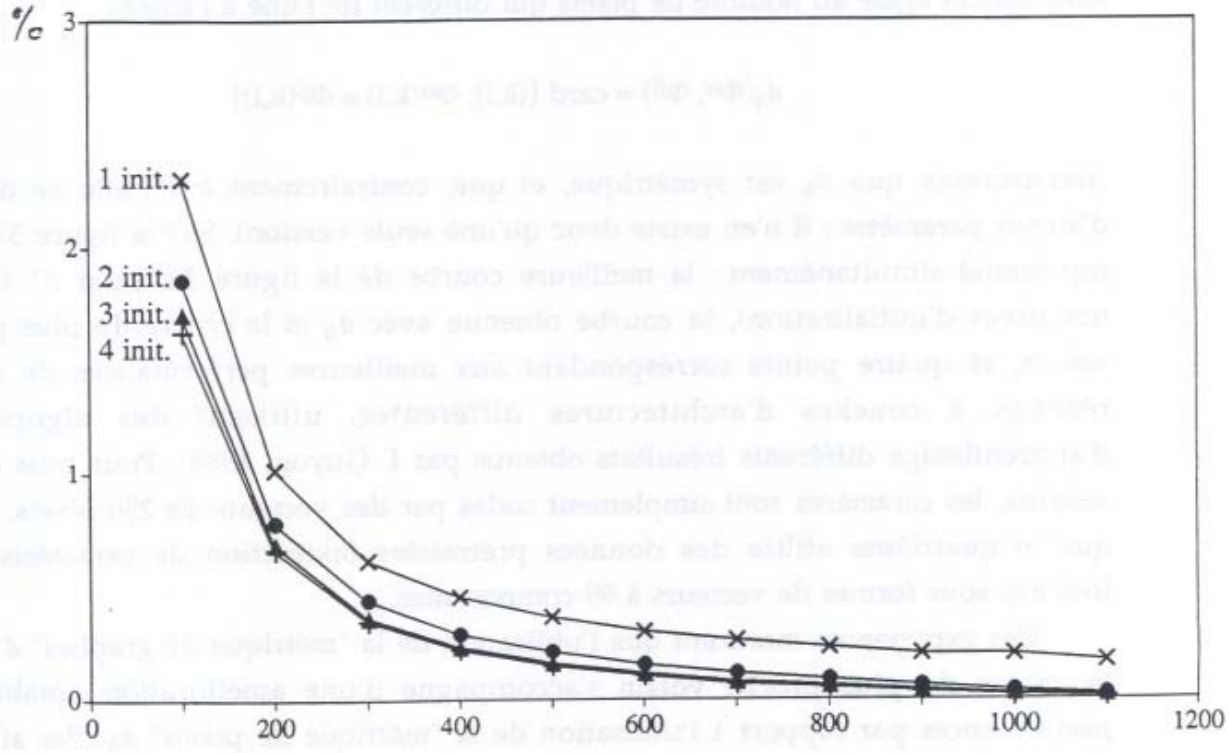


Figure 36 : Diminution du taux d'erreur avec la diversification des initialisations (m=1, κ=2, N=0). Ces quatre courbes ont été obtenues avec : x_1 seul, minimisation sur $\{x_1, x_2\}$, minimisation sur $\{x_1, x_2, x_3\}$, et minimisation sur $\{x_1, x_2, x_3, x_4\}$.

Une autre façon d'améliorer les performances consiste à rester en N=0, en effectuer par ailleurs *plusieurs tentatives* d'initialisation dans chaque appariement pour n'en retenir que la meilleure, c'est à dire celle de plus faible énergie (remarquons que cette sélection se fera dans le calcul de d, avant la symétrisation en

d^*). Ainsi, pour une image Φ^α donnée, chaque tentative sera fondée sur une visite aléatoire différente des nœuds du graphe G^α : $\omega_1, \omega_2, \omega_3, \ldots$ et la sélection se traduira par : $d_{\min}(\Phi^\alpha, \Phi\beta) = \min\{d_{\omega i}(\Phi^\alpha, \Phi\beta)\}$. Puis, on posera comme avant : $d^*_{\min}(\Phi^\alpha, \Phi\beta) = \max\{d_{\min}(\Phi^\alpha, \Phi\beta), d_{\min}(\Phi\beta, \Phi^\alpha)\}$. Il s'agit donc de la recherche légitime au niveau de chaque appariement $(\Phi^\alpha, \Phi\beta)$ du meilleur état **V** dans l'espace $\vartheta^{\alpha\beta}$ (cf. §2.4.d) : au lieu de s'en approcher en poursuivant les itérations de **V**(0) à **V**(N), on calcule plusieurs états initiaux, $\mathbf{V}^1(0), \mathbf{V}^2(0), \mathbf{V}^3(0), \ldots$ et on garde seulement le plus petit $H(\mathbf{V}^i(0))$ (on peut aussi conjuguer les deux types de recherche, et sélectionner le plus petit $H(\mathbf{V}^i(N))$, ce qui nécessiterait plus de calculs). On rappelle que chaque lot particulier d'ordres de visite $\{\omega_i{}^\alpha\}_{\alpha=1\ldots1200}$ est engendré par une racine aléatoire x_i différente : la figure 36 montre l'amélioration graduelle apportée par une minimisation sur 2, 3 ou 4 initialisations déterminées par les générateurs x_1, x_2, x_3, x_4.

Pour finir, nous présentons une comparaison des performances de la distance d^* avec celles de la distance de Hamming, ainsi que d'autres méthodes de classification utilisant des réseaux de neurones formels ("réseaux à couches"). Etant donné deux images Φ^α et $\Phi\beta$, on rappelle que la distance de Hamming est simplement égale au nombre de pixels qui diffèrent de l'une à l'autre :

$$d_h(\Phi^\alpha, \Phi\beta) = \text{card}\ \{(k,l);\ \Phi^\alpha(k,l) \neq \Phi\beta(k,l)\}$$

(remarquons que d_h est symétrique, et que, contrairement à d^*, elle ne dépend d'aucun paramètre : il n'en existe donc qu'une seule version). Sur la figure 37, on a représenté simultanément : la meilleure courbe de la figure 36, pour d^* (quatre tentatives d'initialisation), la courbe obtenue avec d_h et le critère du plus proche voisin, et quatre points correspondant aux meilleures performances de quatre réseaux à couches d'architectures différentes, utilisant des algorithmes d'apprentissage différents (résultats obtenus par I. Guyon, 1988). Pour trois de ces réseaux, les caractères sont simplement codés par des vecteurs de 256 pixels, tandis que le quatrième utilise des données prétraitées (extraction de caractéristiques locales), sous formes de vecteurs à 99 composantes.

Ces expériences montrent que l'utilisation de la "métrique de graphes" d^* pour le critère du plus proche voisin s'accompagne d'une amélioration notable des performances par rapport à l'utilisation de la "métrique de pixels" d_h. Par ailleurs, elles montrent également que les appariements de graphes produisent de meilleurs résultats que l'ensemble des réseaux à couches, que ceux-ci utilisent ou non des images prétraitées.

Bien sûr, ce problème de reconnaissance de caractères manuscrits, dans lequel les exemples sont segmentés, renormalisés, non-bruités, et offent peu de variété à l'intérieur des classes (cf. figure 22), ne présente pas de difficultés fondamentales et peut être traité avec un relatif succès par les méthodes non-paramétriques : on obtient ainsi des taux d'erreur de l'ordre de quelques pourcents avec les "plus

proches voisins sur Hamming" et les réseaux à couches. Cependant, les résultats comparés de la figure 37 montrent que, même dans ce problème relativement aisé, les derniers pourcents d'erreur de ces méthodes peuvent être éliminés en adoptant un format de représentation *relationnel* et en dérivant de ce format une métrique d'"appariement de graphes" qui soit à la base des décisions comparatives.

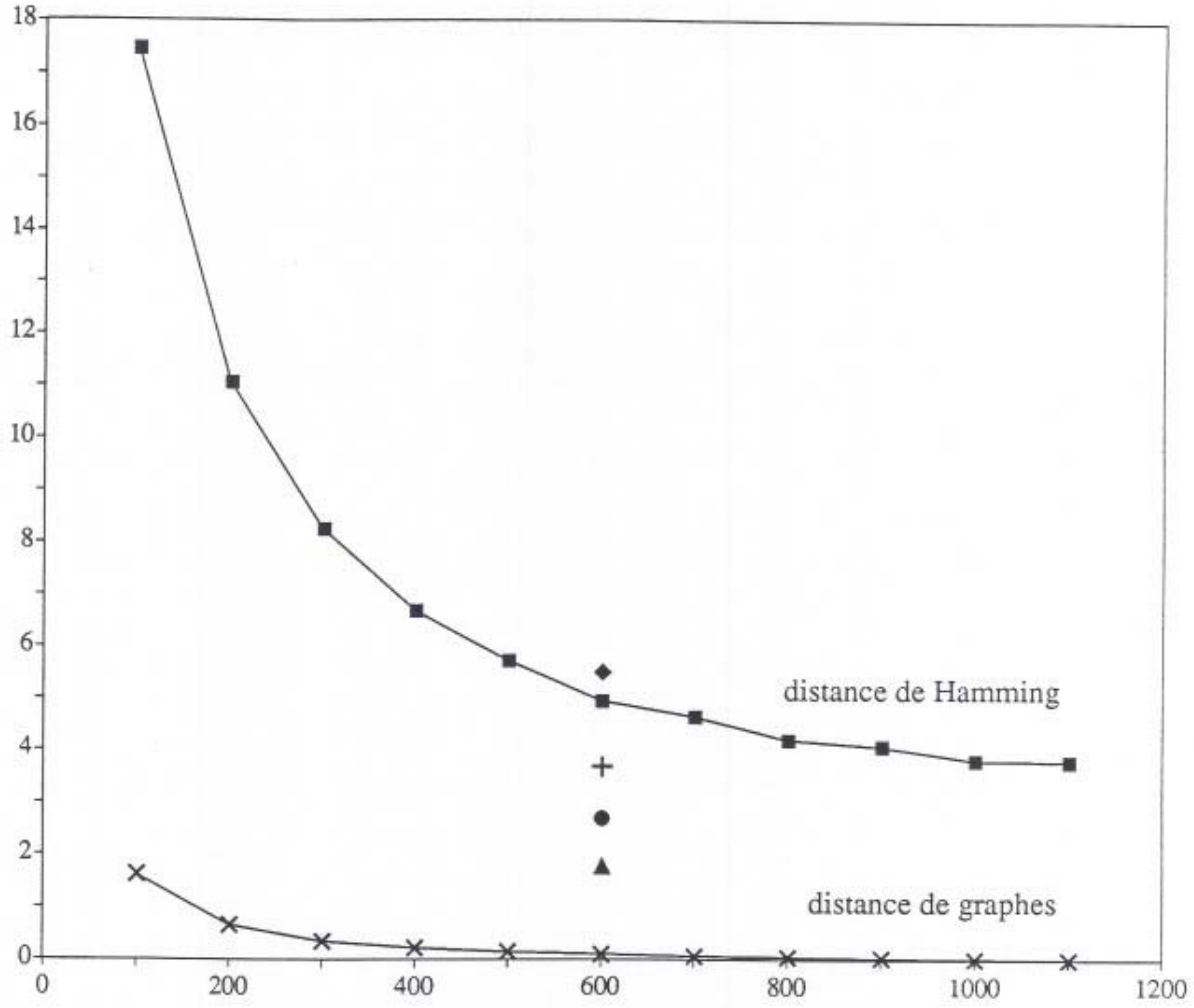


Figure 37 : Comparaison des taux d'erreur obtenus avec la méthode d'appariement élastique par rapport à diverses méthodes non-paramétriques (plus proche voisin sur distance de Hamming et réseaux à couches). La courbe correspondant à d^* reprend la meilleure courbe de la figure 36 (m=1, κ=2, N=0, et minimisation sur quatre initialisations $\{x_1, x_2, x_3, x_4\}$) : sur l'échelle de l'unité de pourcent utilisée ici (au lieu de l'échelle du dixième de pourcent), celle-ci se confond pratiquement avec l'axe horizontal. La courbe correspondant à d_h est construite exactement sur le même principe que la précédente (taux d'erreur moyen calculé sur 1000 partitions aléatoires de taille 10p/10q). En revanche, les expériences sur les réseaux à couches (Guyon, 1988) ne sont fondées que sur une seule partition, de taille unique 600/600 (les taux affichés correspondent toutefois aux meilleurs résultats obtenus en faisant varier diverses caractéristiques de ces réseaux) : de haut en bas, le premier point (losange : 5,5%) est celui d'un perceptron à une seule couche de connexions, utilisant la méthode de la "pseudo-inverse"; le deuxième point (signe plus : 3,7%) correspond à l'algorithme de rétropropagation sur un réseau à deux couches de connexions; le troisième point (cercle : 2,7%) a été obtenu sur un perceptron à une couche, avec "règle-δ" (minimisation de l'erreur quadratique utilisant des unités linéaires au lieu d'unités sigmoïdales); le quatrième point (triangle : 1,8%) est issu de la même méthode que le troisième, mais avec des données prétraitées (vecteurs de 99 "caractéristiques", au lieu des 256 pixels). Remarquons que sur ce même axe (p=600), les taux d'erreur moyens de d_h et d^* sont respectivement : 5% (carré) et 0,06% (croix).